

**CENTRO UNIVERSITÁRIO TERESA D'ÁVILA
MESTRADO PROFISSIONAL EM DESIGN, TECNOLOGIA E INOVAÇÃO**

ANDRESA CRISTIANE DOS SANTOS

**O USO DE JOGOS DIGITAIS COMO FERRAMENTA NO DESENVOLVIMENTO
DO PENSAMENTO COMPUTACIONAL APLICADO À DISCIPLINA DE LÓGICA E
LINGUAGEM DE PROGRAMAÇÃO**

**LORENA
2025**

ANDRESA CRISTIANE DOS SANTOS

O uso de Jogos Digitais como ferramenta no desenvolvimento do pensamento computacional aplicado à disciplina de Lógica e Linguagem de Programação

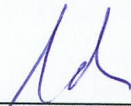
Dissertação apresentada ao Programa de Pós-Graduação - Stricto Sensu - Mestrado Profissional em Design, Tecnologia e Inovação. Centro Universitário Teresa D'Ávila, como parte dos pré-requisitos para obtenção do título de Mestre em Design, Tecnologia e Inovação.

Orientador: Prof. Dr. Adriano José Sorbille de Souza

Data: 09/12/2025

Resultado: Aprovado

BANCA EXAMINADORA



Prof. Dr. Adriano José Sorbille de Souza
Centro Universitário Teresa D'Ávila / UNIFATEA



Prof. Dr. BRUNO BUSTAMANTE FERREIRA LEONOR
FATEC Cruzeiro



Prof. Dra. ALINE FRANCISCA DE SOUZA
Centro Universitário Teresa D'Ávila / UNIFATEA

Santos, Andresa Cristiane

O uso de Jogos Digitais como ferramenta no desenvolvimento do pensamento computacional aplicado à disciplina de Lógica e Linguagem de Programação

Andresa Cristiane dos Santos, orientada por Prof. Dr. Adriano José Sorbille de Souza, Lorena, UNIFATEA, 2025. 96 páginas.

Dissertação (Mestrado Profissional) Centro Universitário Teresa D'Ávila – Programa de Pós-Graduação em Design, Tecnologia e Inovação.

1. Pensamento Computacional. 2. Programação
3. Jogos Digitais 4. Metodologia Ativa

- I. O uso de Jogos Digitais como ferramenta no desenvolvimento do pensamento computacional aplicado à disciplina de Lógica e Linguagem de Programação
- II. Orientador: Prof. Dr. Adriano José Sorbille de Souza

Agradecimentos

Agradeço a Deus por me guiar, iluminar e dar-me tranquilidade e serenidade para seguir meu caminho e dar sequência aos meus objetivos, não desanimando com as dificuldades encontradas.

Ao Professor Doutor Adriano José Sorbille de Souza pelas orientações e sugestões manifestadas ao longo do período em que decorreu a elaboração desta dissertação.

Ao todos os professores que através da suas ações, contribuíram direta ou indiretamente, para que este trabalho fosse executado.

Ao Centro Universitário Teresa D'Ávila que nos permitiu ter esta oportunidade.

Aos meus filhos pelo tempo que lhes tomei com a minha ausência.

Resumo

O setor de Tecnologia da Informação no Brasil encontra-se em expansão, com a previsão de criação de aproximadamente 800 mil novas vagas até 2025. Constatase uma forte demanda por especialistas em desenvolvimento de software, os quais frequentemente apresentam dificuldades em conteúdos relacionados às disciplinas da área de exatas. O desenvolvimento dos quatro pilares do pensamento computacional: decomposição, reconhecimento de padrões, abstração e algoritmos, através de jogos digitais, apresenta-se como uma alternativa para auxiliar este problema. Aprender a programar tem sido destacado pelos alunos como uma tarefa difícil e complexa, causando desmotivação e tendo como resultado altas taxas de reprovação na disciplina e desistência do curso. Neste contexto este trabalho tem como objetivo auxiliar o processo de ensino aprendizagem nas disciplinas de programação de computadores, através do uso de jogos digitais. Esta pesquisa apresenta os resultados de uma pesquisa exploratória e qualitativa, onde participaram deste estudo 14 alunos matriculados no curso técnico em Desenvolvimento de Sistemas integrado ao ensino médio da Escola Estadual Professor Murillo do Amaral, situada na cidade de Aparecida-SP, onde analisou-se e verificou-se os resultados da experiência da utilização dos jogos digitais e sua eficácia na aprendizagem, através de atividades e avaliações práticas e um questionário para os alunos. Como resultado apresenta-se uma metodologia educacional com ênfase no desenvolvimento das habilidades do pensamento computacional, aplicada à disciplina de Lógica e Linguagem de Programação do ensino técnico.

Palavras-chave: Pensamento Computacional. Programação. Jogos Digitais. Metodologia Ativa.

Abstract

The Information Technology sector in Brazil is expanding, with an estimated creation of approximately 800,000 new job positions by 2025. There is a strong demand for software development specialists, who often face difficulties in subjects related to the exact sciences. The development of the four pillars of computational thinking: decomposition, pattern recognition, abstraction, and algorithms, through digital games emerges as an alternative to address this challenge. Learning to program has been identified by students as a difficult and complex task, leading to demotivation and resulting in high failure rates in the subject and course dropout. In this context, this study aims to support the teaching–learning process in computer programming courses through the use of digital games. This research presents the results of an exploratory and qualitative study involving 14 students enrolled in the technical course in Systems Development integrated with secondary education at Professor Murillo do Amaral State School, located in the city of Aparecida, São Paulo, Brazil. The experience of using digital games and their effectiveness in learning were analyzed and evaluated through practical activities, assessments, and a questionnaire administered to the students. As a result, an educational methodology emphasizing the development of computational thinking skills is proposed and applied to the subject of Logic and Programming Language in technical education.

Key-words: *Computational Thinking. Pillars of Computational Thinking. Digital Games. Active Methodology*

Lista de ilustrações

| | | |
|------------|--|----|
| Figura 1- | Pilares do Pensamento Computacional..... | 16 |
| Figura 2- | Pirâmide de aprendizagem de William Glasser..... | 22 |
| Figura 3- | Exemplo código Python..... | 29 |
| Figura 4- | Tela de programação do Scratch..... | 35 |
| Figura 5- | Tela do jogo Cargo Bot..... | 37 |
| Figura 6- | Tela inicial do Code for Life..... | 39 |
| Figura 7 - | Tela de escolha do jogo da plataforma Code for Life..... | 40 |
| Figura 8 - | Tela do jogo Python Den..... | 41 |
| Figura 9- | Primeiro exemplo de <i>sprite</i> | 45 |
| Figura 10- | Segundo exemplo de <i>sprite</i> | 45 |
| Figura 11- | Variáveis no Scratch..... | 46 |
| Figura 12- | Algoritmo Scratch x Python..... | 47 |
| Figura 13- | Exemplo de estrutura de decisão..... | 48 |
| Figura 14- | Carbo Bot - Estrutura de repetição..... | 50 |
| Figura 15- | Scratch - Estrutura de repetição..... | 51 |
| Figura 16- | Python Den - Conteúdo Programático..... | 51 |
| Figura 17- | Python Den - Divisão de níveis..... | 52 |
| Figura 18- | Python Den - Tela do jogo nível 8..... | 53 |
| Figura 19- | Python Den - Tela do jogo nível 13..... | 53 |
| Figura 20- | Python Den - Tela do jogo nível 16..... | 54 |
| Figura 21- | Python Den - Tela do jogo nível 31..... | 55 |
| Figura 22- | Alunos jogando Python Den..... | 55 |
| Figura 23- | Alunos jogando Python Den..... | 56 |
| Figura 24- | Estrutura de repetição: Scratch..... | 57 |
| Figura 25- | Estrutura de repetição: Scratch x Python..... | 57 |
| Figura 26- | Exemplo Lista..... | 58 |
| Figura 27- | Seção <i>Lists – Worksheet</i> | 59 |

Lista de tabelas

| | | |
|-----------|--|----|
| Tabela 1- | Grade Curricular da disciplina de Lógica e Linguagem de Programação..... | 27 |
| Tabela 2- | Jogos excluídos..... | 31 |
| Tabela 3- | Scratch: pilares do pensamento computacional..... | 36 |
| Tabela 4- | Cargo Bot: pilares do pensamento computacional..... | 38 |
| Tabela 5- | Code for Life: pilares do pensamento computacional..... | 42 |
| Tabela 6- | Comparativo Jogos x Pilares do pensamento computacional..... | 43 |
| Tabela 7- | Comparativo Jogos x Pilares mais fortes..... | 43 |
| Tabela 8- | Comparativo de acertos Scratch x Python..... | 61 |
| Tabela 9- | Estrutura Metodológica da Aprendizagem com Jogos Digitais..... | 74 |

Lista de gráficos

| | | |
|-------------|--|----|
| Gráfico 1- | Resultado avaliação 2º bimestre..... | 62 |
| Gráfico 2- | Resultado avaliação 3º bimestre..... | 63 |
| Gráfico 3- | Resultado avaliação 4º bimestre..... | 64 |
| Gráfico 4- | Como você avalia o jogo digitalScratch?..... | 65 |
| Gráfico 5- | Como você avalia o jogo digitalCargo Bot?..... | 65 |
| Gráfico 6- | Como você avalia o jogo digitalPython Den?..... | 66 |
| Gráfico 7- | O uso do jogo digital facilitou a compreensão dos conceitos de programação?..... | 67 |
| Gráfico 8- | Quais desafios você encontrou ao aprender Lógica e Linguagem de Programação por meio dos jogos digitais?..... | 68 |
| Gráfico 9- | Qual jogo você considera mais eficaz como ferramenta para o ensino e a aprendizagem da disciplina de Lógica e Linguagem de Programação?..... | 69 |
| Gráfico 10- | Os jogos digitais tornaram as aulas mais dinâmicas e motivadoras?..... | 72 |

Sumário

| | | |
|----------|---|-----------|
| 1 | Introdução | 10 |
| 1.1 | Objetivo Geral | 12 |
| 1.2 | Objetivos Específicos | 12 |
| 1.3 | Justificativa | 12 |
| 2 | Fundamentação Teórica | 14 |
| 2.1 | Pensamento Computacional | 14 |
| 2.1.1 | Pilares do Pensamento Computacional | 15 |
| 2.1.1.1 | Decomposição | 16 |
| 2.1.1.2 | Reconhecimento de padrões | 17 |
| 2.1.1.3 | Abstração | 17 |
| 2.1.1.4 | Algoritmo | 18 |
| 2.2 | Jogos Digitais | 19 |
| 2.3 | Metodologia Ativa | 20 |
| 2.3.1 | Aprendizagem Baseada em Jogos | 23 |
| 3 | Metodologia | 24 |
| 3.1 | Classificação da pesquisa | 24 |
| 3.2 | Local e Público Alvo da Pesquisa | 26 |
| 3.3 | A disciplina de Lógica e Linguagem de Programação | 26 |
| 3.4 | A linguagem de programação Python | 28 |
| 3.5 | Processo de escolha dos jogos digitais | 29 |
| 3.5.1 | Crterios de exclusão dos jogos | 29 |
| 3.5.2 | Jogos selecionados para aplicação nas aulas | 34 |
| 3.5.2.1 | Scratch | 34 |
| 3.5.2.2 | Cargo Bot | 36 |
| 3.5.2.3 | Code for Life | 38 |
| 3.6 | Aplicação da pesquisa | 44 |
| 3.6.1 | Estrutura Básica: Conceito de variáveis | 46 |
| 3.6.2 | Estrutura de Decisão | 48 |
| 3.6.3 | Estrutura de Repetição | 49 |
| 3.6.4 | Listas | 58 |

| | |
|---|----|
| 4. Coleta de Dados | 60 |
| 4.1 Instrumentos de Coleta | 60 |
| 4.2 Análise dos Dados: Avaliações Práticas | 61 |
| 4.3 Análise dos Dados: Questionário | 64 |
| 5.0 Metodologia de Aprendizagem com Jogos Digitais | 73 |
| 5.1 Fundamentação | 73 |
| 5.2 Estrutura Metodológica | 73 |
| 5.3 Etapas da Aplicação | 76 |
| 5.4 Recursos Didáticos | 76 |
| 5.5 Resultados Esperados | 76 |
| 6. Considerações Finais | 77 |
| REFERÊNCIAS..... | 79 |
| APÊNDICE A – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO | 84 |
| APÊNDICE B – TERMO DO ASSENTIMENTO LIVRE E ESCLARECIDO | 89 |
| APÊNDICE C – QUESTIONÁRIO APLICADO AOS ALUNOS | 91 |
| APÊNDICE D – AVALIAÇÃO PRÁTICA APLICADA NO 2º BIMESTRE..... | 93 |
| APÊNDICE E – AVALIAÇÃO PRÁTICA APLICADA NO 3º BIMESTRE..... | 94 |
| APÊNDICE F – AVALIAÇÃO PRÁTICA APLICADA NO 4º BIMESTRE | 95 |

1 Introdução

O setor de Tecnologia da Informação é um dos que mais abre postos de trabalho no país. A área está em constante crescimento e expansão no Brasil, o que resulta em inúmeras oportunidades de emprego para profissionais qualificados.

A Associação Brasileira das Empresas de Tecnologia da Informação e Comunicação (Brasscom, 2023) prevê a criação de quase 800 mil novas vagas de trabalho na área de TI até 2025.

A média é de 150 mil empregos gerados por ano no Brasil pela área de tecnologia, revelou o diretor de Relações Institucionais e Governamentais da Brasscom, Sgobbi (2023). São números que refletem o quanto o mercado necessita de pessoas formadas e especializadas na área de tecnologia.

Sgobbi (2023) afirma que faltam trabalhadores para as vagas que vão ser disponibilizadas no setor de tecnologia. Nesse sentido, a Brasscom está iniciando jornadas até 2030 pensando em como as tecnologias podem alavancar o Brasil nos níveis regional e global.

Segundo o estudo da Fundação Dom Cabral (2025), divulgado em janeiro, há uma forte demanda por especialistas em desenvolvimento de *softwares*, impulsionada pela crescente digitalização de processos e automação de tarefas em diversos setores.

O desenvolvimento de *software* é uma das áreas mais populares e dinâmicas da tecnologia da informação. A crescente demanda por aplicativos e sistemas eficientes tem impulsionado a valorização de profissionais que dominam linguagens de programação e metodologias de desenvolvimento de *software*. Segundo o guia Desbravando a Tecnologia da Informação, “profissionais que dominam múltiplas linguagens de programação têm uma vantagem competitiva significativa” (SOUZA, 2022, p. 45).

Mesmo com o mercado em alta, a procura por cursos na área de desenvolvimento de *software* ainda é limitada. Estes cursos geralmente incluem disciplinas de programação, sendo a introdução à lógica de programação um componente essencial.

No entanto, essa disciplina é um obstáculo significativo para os alunos e tem sido destacado como uma tarefa difícil e complexa.

“O ensino de lógica de programação em cursos técnicos e superiores apresentam alta taxa de reprovação, acarretando atrasos, desistências e outros problemas; impactando também na formação de profissionais qualificados para o mercado de trabalho” (Pereira; Sousa, 2024).

As dificuldades mais comuns são relacionadas ao raciocínio lógico e à capacidade de abstração, fundamentais para a resolução de problemas em programação (Martins; Marin, 2024).

Nesse sentido, o desenvolvimento das habilidades de pensamento computacional, tais como abstração, reconhecimento de padrões, decomposição e algoritmos através de jogos digitais, é revelado como alternativa para auxiliar este problema.

Uma das formas de trabalhar o pensamento computacional é através de jogos, se mostra uma alternativa promissora, tendo em vista que os jogos podem potencializar a aprendizagem em diversas áreas (Pinho et al., 2016).

O pensamento computacional envolve resolver problemas, projetar sistemas e entender o comportamento humano baseando-se nos conceitos fundamentais da ciência da computação (Wing, 2016). Essa concepção evidencia a relação natural que os sistemas computacionais têm com a escola, que prepara para a vida, pois o pensamento computacional é uma habilidade fundamental para todos, não somente para cientistas da computação (Wing, 2016).

Segundo Castro Junior et al. (2023), o pensamento computacional nos permite decompor problemas complexos em partes menores e mais gerenciáveis, identificar padrões e tendências, abstrair e simplificar os elementos essenciais, e desenvolver soluções algorítmicas precisas. Ao integrar estas habilidades, indivíduos de todas as idades e profissões podem enfrentar desafios com uma nova perspectiva e eficiência notável.

1.1 Objetivo Geral

Investigar o uso de jogos digitais como ferramenta pedagógica no desenvolvimento dos quatro pilares do pensamento computacional: decomposição, reconhecimento de padrões, abstração e algoritmos, para auxiliar a aprendizagem de programação de computadores de alunos do ensino médio e técnico.

1.2 Objetivos Específicos

- Analisar os jogos que estimulem o pensamento computacional dos alunos.
- Tornar as aulas mais atrativas.
- Incentivar os alunos à prática da programação e ao desenvolvimento de *softwares*.
- Apresentar uma sequência didática, uma metodologia para estimular o pensamento computacional dos alunos através da utilização de jogos digitais.

1.3 Justificativa

O macrossetor de Tecnologia, Informação e Comunicação (TIC) no Brasil está projetando gerar entre 30 mil e 147 mil empregos formais até dezembro de 2025, com mais da metade dessas vagas diretamente ligadas à área de tecnologia, indicando uma demanda contínua por profissionais qualificados no setor (Brasscom, 2025).

Contudo, a formação de novos profissionais ainda enfrenta grandes desafios, especialmente nas disciplinas de desenvolvimento de *software*, devido à complexidade dos conhecimentos exigidos e às dificuldades que os alunos encontram em conceitos de exatas e raciocínio lógico.

No Brasil é normal que discentes cheguem em salas de aulas sem terem o conhecimento básico de conceitos computacionais, focado na Lógica de Programação, na resolução de problemas e o processo de raciocínio lógico-matemático causando assim uma taxa de reprovação e desistência nos cursos onde a disciplina é ministrada. Igualmente fazendo deste um grande desafio para os cursos de Educação em Computação (Garlet et al., 2018).

Corroborando com tal assertiva, Calderon et al.(2021), afirmam que:

O processo de aprendizagem em disciplinas de programação de computadores, em cursos da área de Computação, é uma atividade complexa e difícil. Estas disciplinas requerem que os alunos desenvolvam, ao longo da aprendizagem, diferentes habilidades como alta capacidade cognitiva de abstração dos problemas, resolução de problemas, raciocínio e pensamento. Para que os alunos desenvolvam tais habilidades, os docentes precisam adotar novas estratégias de ensino e obter uma postura mais transformadora em sala de aula, proporcionando um ambiente de aprendizagem engajador para os alunos. (CALDERON et al., 2021, p. 1152-1153).

Neste contexto, o uso de jogos digitais aplicados com ênfase nas habilidades do pensamento computacional pode ser uma alternativa para melhorar o aprendizado dos alunos nas disciplinas de programação.

2 Fundamentação Teórica

Este capítulo apresenta as áreas que forneceram os fundamentos teóricos da pesquisa e está organizado da seguinte forma. Primeiramente, tratará do pensamento computacional e seus pilares: decomposição, reconhecimento de padrões, abstração e algoritmos, seguido pelos temas: jogos digitais, metodologia ativa e por fim metodologia ativa baseada em jogos.

2.1 Pensamento Computacional

O pensamento computacional é uma abordagem que utiliza princípios e práticas da ciência da computação para resolver problemas de forma lógica e eficiente, não se resumindo apenas a saber programar ou entender o computador, mas uma forma de pensar a resolução de problemas de maneira eficiente, apesar de surgido no contexto da computação, seus fundamentos são aplicáveis a diversas disciplinas e situações cotidianas (Valente, 2017).

De acordo com Wing (2006) em seu trabalho sobre pensamento computacional definiu esse conceito sob mais de uma perspectiva. Primeiramente, afirmou que o pensamento computacional é uma habilidade fundamental para todos, uma habilidade que qualquer pessoa deve desenvolver, não sendo uma competência necessária apenas para os cientistas da computação, afinal todos precisam resolver problemas. Para leitura, escrita e aritmética, devemos adicionar o pensamento computacional à capacidade analítica de todas as crianças.

Wing (2006) afirmou que pensamento computacional consiste em processos de pensamento envolvidos na formulação de um problema e que expressam solução (ou soluções) eficazmente, de tal forma que uma máquina ou uma pessoa possa realizar.

Wing (2006) também afirmou que a aprendizagem de programação e o desenvolvimento do pensamento computacional podem ser promovidos desde a educação básica por meio de abordagens lúdicas, como os jogos digitais.

A Base Nacional Comum Curricular, em seu complemento dedicado à computação, também destaca a importância do ensino da Computação desde cedo para fomentar a exploração e vivência de experiências de maneira lúdica, se

relacionando com diversos campos da educação básica. A Base Nacional Comum Curricular ainda lista em seu documento uma série de competências e habilidades de pensamento computacional que devem ser ensinadas em cada ano da educação formal, como, por exemplo: capacidade de reconhecimento e identificação de padrões; vivenciar e identificar diferentes formas de interação mediadas por artefatos computacionais; criar e testar algoritmos de forma lúdica com artefatos do ambiente e com movimentos do corpo e resolver problemas usando de decomposição (BRASIL, 2018).

2.1.1 Pilares do Pensamento Computacional

Dentre as habilidades desenvolvidas pelo pensamento computacional estão: delimitar problemas de modo que possamos usar o computador ou ferramentas para solucioná-los; organizar, abstrair, representar e analisar dados; automatizar soluções através de algoritmos; identificar, analisar e implementar soluções, visando o caminho de maior eficiência/eficácia de etapas e recursos; capacidade de generalizar a processos de resolução de problemas para uma grande variedade de outras situações (Wing, 2006).

Outra forma de sistematizar o pensamento computacional é através de seus quatro pilares elementares: decomposição, reconhecimento de padrões, abstração e algoritmos (Brackmann, 2017).

a) Decomposição: dividir um problema complexo em partes menores e mais gerenciáveis.

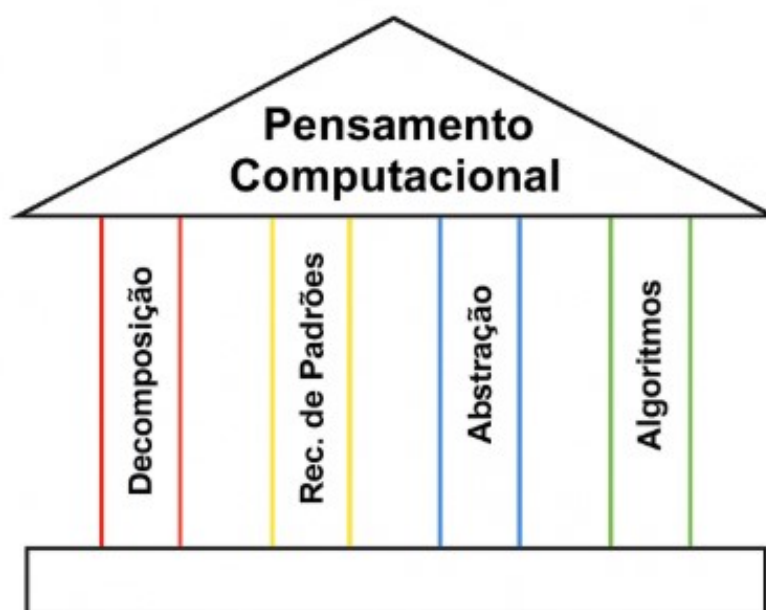
b) Reconhecimento de padrões: identificar semelhanças e diferenças entre os problemas menores para encontrar soluções comuns.

c) Abstração: focar nos detalhes importantes e ignorar os irrelevantes para simplificar a resolução do problema.

d) Algoritmos: desenvolver um conjunto de instruções passo a passo para resolver os problemas de forma eficiente.

Conforme mostra a Figura 1, os quatro pilares do pensamento computacional.

Figura 1 - Pilares do Pensamento Computacional



Fonte: Brackmann (2017)

2.1.1.1 Decomposição

O pilar da decomposição está relacionado com a capacidade de identificar um problema complexo, de difícil resolução, e quebrá-lo em partes menores tornando-o mais fácil de ser resolvido.

Como Wing (2006) observa, a decomposição é um passo crucial no processo de resolução de problemas, pois permite que nos concentremos em partes menores de um problema maior.

Para Liukas (2017, apud Brackmann), a decomposição é um processo pelo qual um problema maior é quebrado em partes menores, o que facilita o

entendimento e a resolução, além de favorecerem o aumento da atenção aos detalhes de cada uma de suas partes.

A decomposição pode ser aplicada em diversas situações do cotidiano, incluindo o ambiente de trabalho, científica e diferentes áreas técnicas.

2.1.1.2 Reconhecimento de padrões

O pilar do reconhecimento de padrões trabalha a identificação de características comuns entre os problemas e as suas possíveis soluções, uma vez que, realizando a decomposição de um problema complexo, seguidamente se encontram padrões entre os subproblemas gerados (Brackmann, 2017). Com isso, torna-se possível simplificar a solução dos problemas e replicar esta solução em cada um dos subproblemas, caso haja semelhança. Liukas (2017, apud Brackmann) define o reconhecimento de padrões como encontrar similaridades com o intuito de resolver problemas complexos de forma mais eficiente.

Esta habilidade permite a extração de informações significativas, a previsão de comportamentos futuros e a tomada de decisões informadas, fundamental para a resolução de problemas, aplicando soluções previamente definidas em experiências anteriores. Questões como “Esse problema é similar a outro que já resolvi?” ou “Como ele é diferente?” são fundamentais nessa etapa, pois ajudam a definir os dados, processos e estratégias a serem utilizados na solução do problema (Batista, 2024).

O reconhecimento de padrões permite simplificar a solução de problemas e replicar esta solução em cada problema que apresente semelhanças. Quanto mais padrões são identificados, mais dinâmica e rápida se torna a solução do problema maior (Batista, 2024).

2.1.1.3 Abstração

A abstração é a capacidade de concentrar-se nos elementos importantes da situação problema, enquanto informações irrelevantes são ignoradas, ou seja, é a ação de ignorar os detalhes desnecessários e concentrar-se no que é verdadeiramente importante. Trata-se do processo de separar detalhes que não são

necessários para se concentrar naquilo que realmente importa (Liukas, 2017, apud Brackmann). A competência essencial deste pilar é escolher o detalhe a ser ignorado para que o problema seja mais fácil de ser compreendido sem perder nenhuma informação que seja importante. Wing (2006) explica que “abstração dá-nos o poder de escala e lidar com a complexidade”.

De acordo com Wing (2006), abstração é o conceito mais importante do pensamento computacional, pois o processo de abstrair é utilizado em diversos momentos, tais como: escrita do algoritmo e suas iterações; seleção dos dados importantes; escrita de uma pergunta; alteridade de um indivíduo em relação a um robô; compreensão e organização de módulos em um sistema.

A abstração é central no pensamento computacional porque permite simplificar a complexidade, identificar padrões essenciais, generalizar soluções, facilitar a comunicação e promover a criatividade. Ao abstrair conceitos e informações, os indivíduos podem enfrentar desafios complexos de forma estruturada, eficiente e inovadora (Batista, 2024).

2.1.1.4 Algoritmo

Os algoritmos são sequências de passos e/ou regras bem definidos e ordenados que descrevem como resolver um problema ou executar uma tarefa. No contexto do pensamento computacional, os algoritmos são essenciais para a resolução sistemática de problemas, a automação de processos e a criação de soluções eficientes. Compreender e desenvolver algoritmos é fundamental para a aplicação prática do pensamento computacional em diversas áreas (Batista, 2024).

Algoritmos estão presentes em muitas situações do dia a dia, na execução lógica de ações no nosso cotidiano. Ajudam a organizar tarefas e a tomar decisões com base em uma série de etapas lógicas.

2.2 Jogos Digitais

Os jogos digitais também denominados *games*, um dos artefatos tecnológicos mais utilizados atualmente, além de conquistarem um espaço importante entre as crianças, jovens e adultos, apresenta crescente utilização no ambiente escolar.

Os jogos são plataformas complexas e atrativas. Neles, os jogadores encontram entretenimento enquanto desenvolvem habilidades para conquistar a vitória. Esse processo é fomentado pela existência do conjunto de regras dos jogos, explícitas e implícitas, característica essencial que traz ordem e direciona o fluxo de jogo, permitindo que o usuário aprenda através delas (Carvalho, 2015).

Uma vez que compreendemos o potencial do jogo enquanto aliado da aprendizagem, passamos a considerar as possibilidades de sua utilização na educação formal. Pesquisas evidenciam que jogos representam um poderoso recurso educacional.

Segundo Guarda e Pinto (2021) há evidências de que jogos, devidamente empregados nos processos de ensino e aprendizagem, proporcionam motivação, potencializam a criatividade, e contribuem para o desenvolvimento intelectual dos alunos.

Ao trazermos os jogos digitais para o cenário educacional, projetando-os com este fim, estes recebem a denominação de jogos educacionais, os quais têm como objetivo aliar o entretenimento ao aprendizado, promovendo o desenvolvimento de habilidades cognitivas e sociais nos alunos (Savi, 2008).

Deve-se ressaltar, neste ponto, que a simples inserção de um jogo no contexto educacional não é suficiente para que o jogo possa ser classificado como tal, por isso Pietro et al. (2005, p. 10) observa que estes:

Devem possuir objetivos pedagógicos e sua utilização deve estar inserida em um contexto e em uma situação de ensino baseados em uma metodologia que oriente o processo, através da interação, da motivação e da descoberta, facilitando a aprendizagem de um conteúdo (PIETRO et al., 2005, p.10).

Assim, a integração de jogos digitais na sala de aula depende do professor, de alinhar com os conteúdos programáticos, os métodos instrucionais, as necessidades dos alunos e os objetivos de aprendizado.

De certa forma, só será possível esse alinhamento, se o professor tiver propriedade dos elementos de seu conteúdo disciplinar, bem como o conhecimento do jogo proposto para atuar no contexto proposto.

2.3 Metodologia Ativa

Jean Piaget, considerado um dos mais importantes pensadores do século XX e autor da teoria construtivista, defendia que a criança assume um papel ativo no processo de aprendizagem, construindo o conhecimento por meio da experimentação, da observação e da interação com o meio (Piaget, 1973).

A pedagogia construtivista, baseada no trabalho de Piaget, defende que o conhecimento é construído pelo aluno através da interação com o ambiente e com os seus conhecimentos prévios. O professor, nesse contexto, passa a ser um mediador do processo de aprendizagem, incentivando a autonomia, a resolução de problemas e a reflexão dos alunos (Barbosa, 2015). Tal perspectiva encontra coerência nas chamadas metodologias ativas de ensino, que colocam o aluno no centro do processo educativo, promovendo sua participação ativa na construção do saber.

As metodologias ativas, como Aprendizagem Baseada em Projetos, a Aprendizagem Baseada em Problemas entre outras, compartilham com o construtivismo a valorização da autonomia, da experimentação e da reflexão crítica. Elas se fundamentam na ideia de que a aprendizagem se torna mais significativa quando o aluno se envolve diretamente na resolução de situações reais ou simuladas, mobilizando conhecimentos prévios, desenvolvendo habilidades cognitivas superiores e favorecendo a construção de novos significados (Barbosa, 2015).

Moran (2018, p. 41) define a metodologias ativas como “estratégias de ensino centradas na participação efetiva dos alunos na construção do processo de

aprendizagem, de forma flexível, interligada e híbrida”. Dessa maneira, o aluno que era um participante passivo no sistema escolar tradicional torna-se o protagonista do seu aprendizado. O professor, que na sala de aula tradicional era o detentor e transmissor do conhecimento, tem uma função mais ampla e complexa. Pois, ele é quem conduz as trilhas personalizadas de aprendizado para um determinado discente ou para grupos de alunos, como também é um orientador dos projetos da vida profissional e pessoal do aluno (Moran, 2018).

O conceito de metodologia ativa envolve práticas pedagógicas inovadoras que incentivam a participação ativa e a construção do conhecimento de forma colaborativa e prática. Segundo Freire (1996), “ensinar não é transferir conhecimento, mas criar as possibilidades para a sua produção ou construção”, um princípio que sustenta a aplicação dessas metodologias. Elas incentivam o aluno a ser ativo e crítico, conectando os conteúdos aprendidos com a prática e com problemas reais, promovendo uma aprendizagem significativa e autônoma.

Vários autores constatarem os benefícios da aplicação de metodologias ativas, Souza et al. (2020) ressaltam o desenvolvimento da autonomia do aluno, o aguçamento da visão crítica dele perante a realidade, uma melhor integração da teoria com a prática e o favorecimento para uma avaliação formativa.

Entre as metodologias ativas podemos destacar a Aprendizagem Baseada em Projetos, a Sala de Aula Invertida, a Aprendizagem Baseada em Problemas e a Aprendizagem Baseada em Jogos.

A Aprendizagem Baseada em Projetos (*Project-Based Learning*), os alunos desenvolvem projetos multidisciplinares para resolver desafios ou criar produtos, conectando teoria à prática em contextos reais (Bender, 2014).

A metodologia Sala de Aula Invertida (*Flipped Classroom*), propõe que os alunos estudem previamente os conteúdos teóricos, geralmente por meio de vídeos, textos ou materiais digitais, permitindo que o tempo em sala de aula seja dedicado a discussões, atividades práticas e à resolução de problemas, com maior interação entre alunos e professores (Bergmann, 2012).

A Aprendizagem Baseada em Problemas (*Problem-Based Learning*), os alunos trabalham em grupo para resolver problemas reais ou hipotéticos,

desenvolvendo habilidades de pesquisa, análise crítica e trabalho em equipe (Savery, 2006)

Já a Aprendizagem Baseada em Jogos (*Game-Based Learning*) é uma metodologia ativa que utiliza jogos como ferramentas pedagógicas para engajar os alunos, promover o aprendizado e desenvolver competências e habilidades de forma lúdica e interativa, potencializando a motivação e o envolvimento no processo educativo (Prensky, 2012).

Figura 2 - Pirâmide de Aprendizagem de William Glasser



Fonte: Brackmann (2017)

Segundo a interpretação dos estudos de Glasser (2001), a "Pirâmide de Aprendizagem" divide os métodos de ensino em passivos e ativos, demonstrando que a retenção do conhecimento aumenta conforme o envolvimento ativo do aluno no processo de aprendizagem.

Destaca-se para este trabalho a aplicação da metodologia baseada em jogos digitais.

2.3.1 Aprendizagem Baseada em Jogos

A Aprendizagem Baseada em Jogos, fundamentada por Prensky (2012), utiliza algum “jogo para o processo de ensino e aprendizagem em um computador ou *online*”. Portanto, entende-se que essa metodologia se beneficia de jogos digitais quando se adequam aos objetivos de ensino e aprendizagem. Essa abordagem é fundamentada na ideia de que o aprendizado é mais eficaz quando envolve experiências práticas e divertidas, que desafiam e engajam os alunos.

Nesse contexto, também podemos destacar os jogos digitais dentre as metodologias ativas, por sua capacidade de promover a motivação nos processos de ensino e aprendizagem, aumentando o interesse dos alunos em aprender (Falcão, 2015). A aprendizagem baseada em jogos, utiliza jogos digitais ou analógicos, como ferramentas para promover o aprendizado de forma interativa e envolvente. Os jogos estimulam os alunos a participarem ativamente do processo de aprendizagem, permitindo que desenvolvam habilidades como resolução de problemas, pensamento crítico, colaboração e criatividade. Além disso, o uso de jogos costuma trazer um componente motivacional significativo, já que eles frequentemente envolvem desafios, competição, recompensas e *feedback* imediato.

Para Martins et al. (2023), os jogos digitais são uma ferramenta facilitadora de aprendizado, pois possibilita os alunos aprenderem se divertindo. Os jogos digitais são uma ferramenta computacional que facilita o raciocínio lógico, devido aos *designers* gráficos que ajudam na fixação do conteúdo aprendido, tornando assim fatores importantíssimos para aprendizagem.

Ao utilizar jogos como ferramenta pedagógica, coloca-se o aluno no centro do processo de aprendizagem, estimulando sua participação ativa e engajada. Através de desafios, interações e resolução de problemas propostos nos jogos, os alunos são motivados a construir seu próprio conhecimento de forma lúdica e significativa.

3 Metodologia

Este capítulo apresenta os procedimentos metodológicos adotados para o desenvolvimento desta pesquisa. São descritos a abordagem e o tipo de pesquisa, os participantes envolvidos, o ambiente da investigação, os instrumentos utilizados e os procedimentos de aplicação.

3.1 Classificação da pesquisa

A metodologia utilizada para a realização do trabalho classifica-se como pesquisa de abordagem qualitativa e se caracteriza como exploratória. De acordo com Gil (2018), a pesquisa exploratória tem como abrangência o levantamento bibliográfico, entrevistas com pessoas e análise de exemplos que “estimulem a compreensão” de forma a propiciar maior proximidade na descoberta de um problema ou na formulação de uma hipótese.

Quanto ao procedimento para o desenvolvimento da pesquisa, utilizou-se a pesquisa bibliográfica, pesquisa-ação e o método heurística.

A pesquisa primeiramente se fundamenta em uma revisão bibliográfica, que consiste em buscar, analisar e sintetizar estudos já existentes sobre o tema. Esta etapa contribuiu para o levantamento de informações sobre os conceitos; identificar vários autores, tratando de forma diferente o mesmo conceito e facilitar a compreensão do problema e coleta de informações relevantes a pesquisa.

Para Gil (2018) a revisão bibliográfica é pesquisa em material já publicado tradicionalmente e fundamentado cientificamente em revistas científicas, livros, anais entre outros em virtude da disseminação do conhecimento científico. O autor afirma ainda que toda pesquisa científica deve se embasar na pesquisa bibliográfica pois trata de dar fundamentação teórica mesmo que para novas práticas.

Para Saur-Amaral (2010), toda investigação que inicia deve necessariamente considerar a análise do estado da arte associado ao tema integrando o aprendizado obtido com pesquisas anteriores e aumento as chances de efetivamente obter contribuições para a solução do problema da pesquisa.

O segundo método de pesquisa adotado, a pesquisa-ação é aplicável em praticamente qualquer situação problema que envolva pessoas, tarefas e

procedimentos e que necessitem de uma solução adequada, com o objetivo de não apenas compreender as relações, mas proporcionar a melhor interação e a prática das ações (COHEN et al., 2005).

O método pesquisa-ação foi aplicado seguindo as etapas segundo as proposições de Thiollent (2011) e Gil (2011); revisão bibliográfica, fase exploratória, problema da pesquisa, planejamento colaborativo, protocolo de coleta de dados, coleta dos dados, reflexão e divulgação externa. Este método permitiu identificar problemas reais, desenvolver e implementar planos de ação e analisar os resultados obtidos na prática.

O terceiro método utilizado foi a heurística. Segundo Gigerenzer (2008) o método heurístico caracteriza-se por uma abordagem intuitiva e prática para a resolução de problemas, baseada na experiência e não em regras fixas ou procedimentos formais. Visa soluções rápidas e suficientemente eficazes. Conforme Gigerenzer (2008 p. 44) “As heurísticas baseiam-se em regras simples que funcionam bem em contextos específicos, especialmente quando derivadas da experiência acumulada em ambientes semelhantes.”

O estudo de heurísticas inclui no seu escopo tanto o conhecimento oriundo diretamente de indivíduos como aquele consubstanciado em artefatos (conceitos, modelos, ferramentas, métodos, produtos/serviços). Considera informações pertinentes de base lógica ou psicológica (atitudes, opiniões, experiências, emoções, operações mentais) requeridos para a realização de soluções. Segundo Santos (2018, p. 130)

De acordo com Hiles (2001) o método heurístico é adequado para pesquisas que buscam explorar e interpretar experiências enfatizando o ponto de vista do próprio pesquisador ou da equipe de pesquisadores.

A aplicação do método heurístico possibilitou contribuições significativas para a pesquisa, especialmente nos processos de formulação de hipóteses, vivência da experiência de professor e observação dos alunos ao realizar as atividades práticas.

O *Design Thinking* foi empregado como ferramenta metodológica no âmbito do método científico heurístico, com o propósito de favorecer a resolução criativa e colaborativa de problemas, centrada nas necessidades e experiências dos alunos (Brown, 2009). Este método consiste em um processo organizado em cinco etapas: imersão, definição, ideação, prototipagem e testes (Plattner et al., 2010). A aplicação

do *Design Thinking* permitiu compreender melhor as dificuldades dos alunos, estruturar atividades pedagógicas de forma colaborativa, prototipar e testar soluções baseadas em jogos digitais, ajustando-as conforme o *feedback* obtido.

3.2 Local e Público Alvo da Pesquisa

A pesquisa foi realizada durante o horário das aulas da disciplina de Lógica e Linguagem de Programação do curso técnico em Desenvolvimento de Sistemas integrado ao ensino médio da Escola Estadual Professor Murillo do Amaral, situada na cidade de Aparecida, estado de São Paulo.

Participaram deste estudo 14 alunos entre 16 e 17 anos de idade de ambos os sexos, regularmente matriculados no 2º ano do ensino médio técnico. Todas as aulas da disciplina Lógica e Linguagem de Programação foram realizadas no laboratório de informática da escola, com a utilização de *softwares* de programação de computadores e jogos digitais como ferramenta pedagógica no auxílio de aprendizagem do conteúdo da disciplina.

3.3 A disciplina de Lógica e Linguagem de Programação

A disciplina de Lógica e Linguagem de Programação integra a base curricular do curso técnico em Desenvolvimento de Sistemas, sendo ofertada no 2º ano do ensino médio integrado. Essa distribuição ocorre porque o primeiro ano é dedicado exclusivamente às disciplinas da BNCC, enquanto a formação técnica tem início apenas no 2º ano.

Trata-se de uma das disciplinas mais relevantes do curso, cuja ênfase é a programação. Nela, os alunos têm o primeiro contato com os fundamentos da lógica de programação e com uma linguagem de programação específica, neste caso, a linguagem Python.

Entre as competências desenvolvidas, destacam-se: a capacidade de elaborar sistemas computacionais utilizando ambiente de desenvolvimento e a

habilidade de realizar a migração de sistemas, com a implementação de rotinas e estruturas de dados mais eficientes (SÃO PAULO, 2024).

Conforme o plano de curso da secretária do estado de São Paulo aprovado pelo Centro Paula Souza (2024), a disciplina têm como objetivos abordar: conceito de lógica; lógica aplicada à programação; tipos de algoritmos; conceitos de variáveis e constantes; tipos primitivos de dados; operadores (aritméticos, relacionais e lógicos); estruturas de seleção; estruturas de controle e repetição; vetores e matrizes.

A organização curricular do curso técnico em Desenvolvimento de Sistemas busca proporcionar aos alunos uma formação abrangente, contextualizada e alinhada às demandas do mercado de trabalho.

A Tabela 1 abaixo ilustra a grade curricular da disciplina conforme plano de curso da Secretaria de Educação - SP.

Tabela 1 - Grade Curricular da disciplina de Lógica e Linguagem de Programação

| Unidades Curriculares | |
|------------------------------|--------------------------------|
| I | Estrutura Básica |
| II | Estruturas de Decisão Simples |
| III | Estruturas de Decisão Composta |
| IV | Estruturas de Seleção |
| V | Estruturas de Repetição |
| VI | Vetores e Matrizes |

Fonte: Plano de curso técnico em Desenvolvimento de Sistemas (2024)

3.4 A linguagem de programação Python

Várias linguagens de programação poderiam ser utilizadas na disciplina de Lógica e Linguagem de Programação; entretanto, Python foi selecionada para o desenvolvimento das atividades propostas da base curricular do curso e desta pesquisa por apresentar sintaxe clara e de fácil compreensão. O Ambiente de Desenvolvimento Integrado utilizado para a elaboração dos códigos foi o Visual Studio Code, da Microsoft, ambos gratuitos.

Conforme Martins (2023), Python é uma linguagem de programação de alto nível, lançada por Guido Van Rossum em 1991 e mantida atualmente de forma colaborativa. Devido à sua versatilidade, é amplamente utilizada em ambientes corporativos.

Sua popularidade também se relaciona à legibilidade do código e à proximidade da sintaxe com a linguagem natural, o que facilita a aprendizagem e a implementação de soluções (Borges, 2021).

Python possui ampla área de atuação, sendo utilizada em ciência de dados, automação, desenvolvimento web, inteligência artificial e diversas outras áreas. A linguagem também oferece suporte a aplicações *desktop*, *mobile*, processamento de imagens, geoprocessamento, robótica, programação para hardware, desenvolvimento de jogos e pesquisa científica, incluindo o trabalho com números grandes e complexos (Silva, I. R. S.; Silva, R. O., 2019).

Além disso, destaca-se por permitir o desenvolvimento de aplicações com poucas linhas de código quando comparada a outras linguagens, o que reforça sua ampla adoção mundial. A Figura 3 apresenta um exemplo básico de código em Python.

Figura 3 - Exemplo códigoPython

```
print("***** CALCULA IMC *****")

nome = input('Entre com seu nome: ')
idade = int(input('Entre com sua idade: '))
peso = float(input('Entre com seu peso: '))
altura = float(input('Entre com sua altura: '))

imc = peso/(altura * altura)

print('Nome: ', nome)
print('Peso: ', peso)
print('Altura: ', altura)
print('Idade: ', idade)
print('IMC : %.2f ' %imc)
```

Fonte: Próprio Autor (2025)

3.5 Processo de escolha dos jogos digitais

A presente pesquisa foi conduzida a partir de uma revisão sistemática da literatura sobre o tema, seguida da investigação e análise de jogos digitais que demonstraram potencial para fomentar o desenvolvimento de habilidades do pensamento computacional, quando empregados como recursos no processo de ensino e aprendizagem da programação de computadores.

3.5.1 Critérios de exclusão dos jogos

Foram filtrados diferentes tipos de jogos digitais disponíveis no mercado, com o intuito de relacioná-los ao desenvolvimento das habilidades correspondentes aos quatro pilares do pensamento computacional.

A pesquisa foi conduzida através da ferramenta de busca Google, explorando *sites* especializados em educação, jogos educacionais e tecnologia, a fim de identificar os jogos digitais que promovessem o pensamento computacional, estimulasse o raciocínio lógico e ou contribuíssem para o desenvolvimento da lógica de programação

Foram selecionados quinze jogos digitais, a seleção dos jogos utilizados na pesquisa seguiu critérios específicos, pautados em aspectos técnicos e pedagógicos, com vistas à sua aplicabilidade em sala de aula. Para garantir a adequação ao ambiente escolar e aos objetivos do estudo, foram estabelecidos os seguintes critérios de exclusão:

a) Licenciamento comercial: foram desconsiderados jogos que exigissem qualquer forma de pagamento ou licenciamento, sendo selecionados apenas aqueles com acesso totalmente gratuito.

b) Plataforma de execução *mobile*: jogos desenvolvidos exclusivamente para dispositivos móveis foram excluídos, uma vez que o uso do celular não é permitido durante as aulas. Assim, priorizaram-se jogos compatíveis com computadores, preferencialmente executáveis *online* e sem necessidade de instalação.

c) Inadequação etária: foram eliminados jogos cuja classificação indicativa não contemplasse a faixa etária dos alunos do ensino médio e técnico, entre 16 à 17 anos.

d) Exigência de linguagem de programação específica: jogos que demandassem o uso de linguagens de programação diferentes de Python foram descartados, considerando que essa é a linguagem adotada na disciplina de Lógica e Linguagem de Programação, conforme a grade curricular do curso.

e) Nível de dificuldade incompatível: foram excluídos jogos cujo nível de complexidade estivesse desalinhado ao grau de conhecimento dos alunos. No entanto, foram considerados admissíveis jogos que exigissem domínio intermediário ou avançado da linguagem Python, desde que esse domínio estivesse em conformidade com os conteúdos abordados no curso.

Após a análise dos quinze jogos, doze foram excluídos por não atenderem integralmente aos critérios estabelecidos, conforme ilustrado na Tabela 2.

Tabela 2 - Jogos excluídos

| Jogo | Descrição | Linguagem de programação | Critério de exclusão |
|-------------------------|---|---|---|
| 7 Billion Humans | Este jogo programa-se grupos de trabalhadores de escritório para resolver desafios lógicos. Cada trabalhador executa o mesmo conjunto de instruções simultaneamente, mas reage de forma diferente com base em sua posição e nos dados ao redor. O objetivo é automatizar tarefas como ordenar números, mover blocos e tomar decisões condicionais. | Não utiliza linguagem de programação | Exige compra de licença |
| Colobot | Jogo educativo de estratégia combina elementos de programação e exploração espacial, o jogador assume o papel de um astronauta encarregado de encontrar um novo planeta habitável para a humanidade, após a Terra ter sido devastada. A missão envolve explorar diversos planetas, construir infraestrutura, coletar recursos e enfrentar formas de vida <u>alienígenas</u> hostis. Para auxiliá-lo, o jogador conta com robôs que podem ser programados para executar tarefas específicas. | Linguagem <i>CBOT</i> , que possui uma sintaxe semelhante à de linguagens como C++ e Java | Direcionado para aprendizagem da linguagem de programação C++ e Java. |
| CodeMonkey | Plataforma online, que ensina às crianças linguagens de programação. Crianças e adolescentes aprendem codificação baseada em blocos e texto por meio de um ambiente envolvente semelhante a um jogo. Oferece uma versão gratuita com funcionalidades limitadas. | CoffeeScript e Python | Inadequação etária. Direcionada ao aprendizado infantil. Crianças de 5 a 14 anos. |

| | | | |
|--|--|--|--|
| | | | |
|--|--|--|--|

A página seguinte apresenta a continuidade da Tabela 2.

| Jogo | Descrição | Linguagem de programação | Critério de exclusão |
|-------------------|--|---|---|
| CodeCombat | Plataforma online que ensina programação de forma interativa e lúdica. Com uma ambientação medieval e elementos de jogos de aventura, a plataforma combina desafios de codificação com uma narrativa envolvente, permitindo que os usuários aprendam linguagens de programação ao progredirem pelos níveis e resolverem problemas. | Linguagens de programação disponíveis, como Python, JavaScript ou C++ | Exige compra de licença, somente um jogo é disponibilizado gratuitamente, limitado até o nível 5. |
| Codin Game | Plataforma online voltada a programadores e alunos de programação. Oferece variados jogos para o desenvolvimento de algoritmos, estruturas de dados, matemática e lógica (<i>Puzzles</i>), desafios de jogos que obtenham o código-fonte mais curto possível (<i>Code Golf</i>) ou otimizadas segundo critérios como pontuação, uso de recursos e tempo de execução (<i>Optimization</i>). | 27 linguagens incluindo Python, C#, C++, Java, Java Script, PHP entre outras. | Nível de dificuldade incompatível. Jogos avançados para iniciantes. |
| Code.org | Plataforma educacional online com o objetivo de ensinar programação e ciência da computação. Oferece uma ampla variedade de cursos e jogos interativos voltados principalmente para crianças e adolescentes | Não utiliza linguagem de programação. Programação em blocos. | Direcionado ao público infantil. |

| | | | |
|----------------|---|---------------------|---|
| Checkio | Plataforma online que combina aprendizado de programação com elementos de jogos, oferecendo desafios interativos para aprimorar suas habilidades em linguagens de programação. Os usuários resolvem problemas de codificação, desbloqueiam novos desafios | Python e TypeScript | Página muito poluída e confusa, com muito texto em inglês, que causaria uma dificuldade para os alunos. |
|----------------|---|---------------------|---|

A página seguinte apresenta a continuidade da Tabela 2.

| Jogo | Descrição | Linguagem de programação | Critério de exclusão |
|----------------------|---|--------------------------------------|---|
| Elevator Saga | Com uma interface simples e intuitiva, aprende-se os fundamentos da programação enquanto se diverte resolvendo problemas cada vez mais complexos. O objetivo é controlar um elevador em um prédio para transportar as pessoas de maneira eficiente. As fases vão ficando cada vez mais difíceis, e para completá-las, é preciso programar as funções de acordo com os comandos solicitados. | JavaScript | Direcionado a pessoas que já apresentam certo domínio em programação e desejam aprimorar o aprendizado na linguagem Javascript. |
| LightBot | Um jogo de quebra cabeça que, de forma lúdica, ensina conceitos de linguagem de programação de maneira visual e divertida. Tem aparência atrativa e oferece um tutorial objetivo e fácil de entender. São oferecidos 40 níveis onde aprende-se desde os conceitos básicos, incluindo procedimentos, loops recursivos e condicionais. | Não utiliza linguagem de programação | Direcionado ao público infantil |
| Blockly Games | Coleção de jogos desenvolvidos pelo <i>Google</i> , projetada para ensinar conceitos de programação de forma interativa e lúdica. Utilizando uma interface de blocos visuais que se encaixam, os jogos permitem que os usuários criem algoritmos sem a necessidade de escrever código textual, tornando-se uma excelente introdução ao pensamento | Não utiliza linguagem de programação | Direcionado ao público infantil |

| | | | |
|--|--------------------------------|--|--|
| | computacional para iniciantes. | | |
|--|--------------------------------|--|--|

A página seguinte apresenta a continuidade da Tabela 2.

| Jogo | Descrição | Linguagem de programação | Critério de exclusão |
|----------------------------------|--|--------------------------------------|--|
| Catastrophic Construction | Jogo de lógica e física no qual o jogador deve empilhar formas geométricas dentro de uma área delimitada, garantindo que a estrutura permaneça estável. O jogo desafia habilidades de planejamento, raciocínio espacial e compreensão básica das leis da física | Não utiliza linguagem de programação | Direcionado para o ensino da Física e crianças. |
| RoboCode | Plataforma de programação <i>open source</i> , voltado principalmente para alunos do ensino médio e universitários. Jogador programa tanques de guerra virtuais chamados de <i>robots</i> para que lutem uns contra os outros em uma arena. Os robôs não são controlados com teclado ou <i>mouse</i> , são controlados por código. | Suporte a linguagens Java e C# | Não suporta Python como linguagem de programação |

Fonte: Próprio autor (2025)

3.5.2 Jogos selecionados para aplicação nas aulas

A partir de pesquisas realizadas e da análise de diferentes jogos digitais, foram selecionados três que atenderam aos critérios estabelecidos: Scratch, CargoBot e Code for Life.

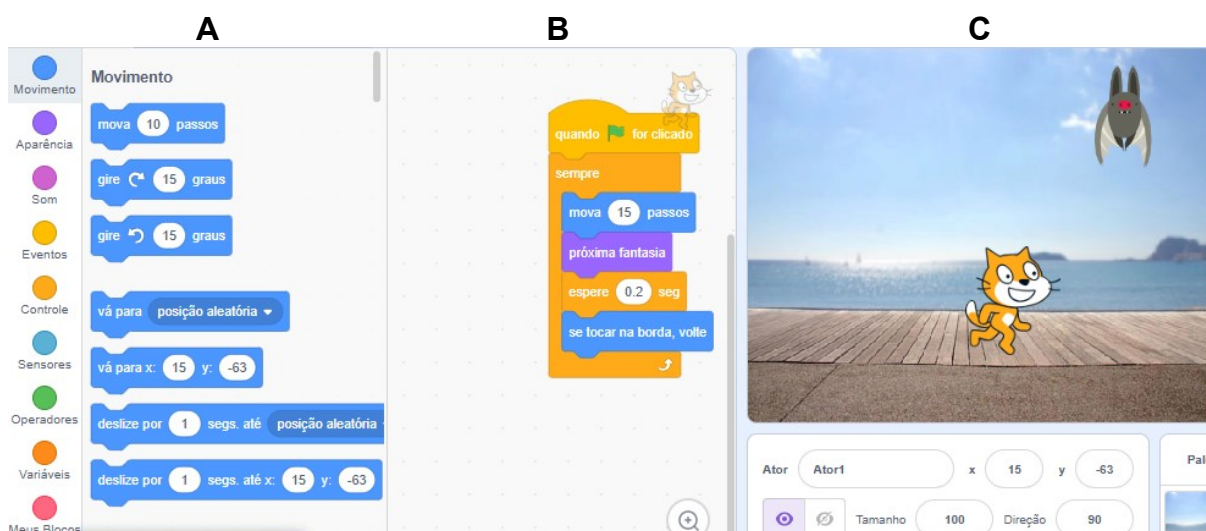
3.5.2.1 Scratch

O Scratch é uma plataforma web interativa e visual criada pelo MIT (Massachusetts Institute of Technology) para ensinar programação e lógica de forma

acessível. Usa blocos parecidos com os blocos de encaixe Lego, coloridos e contendo uma função para cada bloco, assim, apresenta forma e cor diferente para cada tipo de operação, permitindo que os usuários arrastem e soltem blocos de código para criar programas, sem a necessidade de digitar código tradicional.

Conforme mostra a Figura 4, na tela do jogo Scratch, os blocos disponíveis para uso estão localizados à esquerda (A), os blocos encaixados em sequência lógica, conhecidos como *sprites*, estão posicionados ao centro (B), e à direita está a visualização da execução do *sprite* (C).

Figura 4 - Tela de programação do Scratch



Fonte: Próprio autor (2025)

A programação em blocos é uma introdução à lógica da programação, permitindo que os iniciantes compreendam os conceitos fundamentais sem se preocupar com a sintaxe específica de uma linguagem de programação (Batista, 2024).

Além de ser voltada especialmente para alunos entre 8 e 16 anos, a ferramenta pode ser utilizada para criar um jogo, contar uma história, resolver problemas e permite o compartilhamento das criações.

O Scratch é projetado, desenvolvido e moderado pela Fundação Scratch, uma organização sem fins lucrativos. Está disponível gratuitamente tanto *online* quanto para *download*, e oferece suporte a mais de 70 idiomas (Scratch, 2025).

Promove o pensamento computacional e habilidades de resolução de problemas; ensino e aprendizagem criativos; autoexpressão e colaboração; e equidade em computação (Scratch, 2025).

De acordo com a Tabela 3, o jogo Scratch contribui para desenvolvimento do pensamento computacional ao contemplar seus quatro pilares.

Tabela 3 - Scratch: pilares do pensamento computacional

| Decomposição | Reconhecimento de Padrões | Abstração | Algoritmo |
|---|---|---|---|
| Para criar um projeto, o jogador precisa decompor a ideia principal em blocos menores como: criar o cenário, os personagens, definir interações, e implementar ações específicas. | Os jogadores identificam padrões em comportamentos repetitivos, ou interações repetidas entre objetos. Pode-se usar blocos de repetição (<i>loops</i>). | O Scratch permite que o jogador se concentre na lógica da programação, facilitando a abstração dos conceitos, sem se preocupar com a sintaxe da linguagem de programação. | Para que seus projetos funcionem, os jogadores precisam criar sequências de comandos em blocos de código de maneira lógica, criando assim algoritmos. |

Fonte: Próprio autor (2025)

3.5.2.2 Cargo Bot

Cargo Bot é um jogo de quebra-cabeça baseado em programação que ensina conceitos fundamentais de algoritmos e lógica de programação de uma forma lúdica.

O objetivo do jogo é empilhar caixas de forma específica (organiza-las no padrão solicitado), os jogadores assumem o controle de um braço robótico que deve mover caixas para destinos específicos, seguindo uma sequência de comandos.

O desafio principal do Cargo Bot é criar sequências de comandos (como mover para a direita, esquerda, pegar ou soltar uma caixa) para programar o braço robótico de maneira eficiente e alcançar os objetivos do quebra-cabeça (Pavlus, 2012).

Conforme os jogadores avançam nos níveis, os quebra-cabeças tornam-se mais complexos, exigindo uma compreensão mais profunda de conceitos como estruturas de condição, repetição e funções.

Cargo Bot é um jogo intuitiva e visual para o ensino de programação, tornando-o acessível para iniciantes enquanto oferece desafios. O jogo proporciona

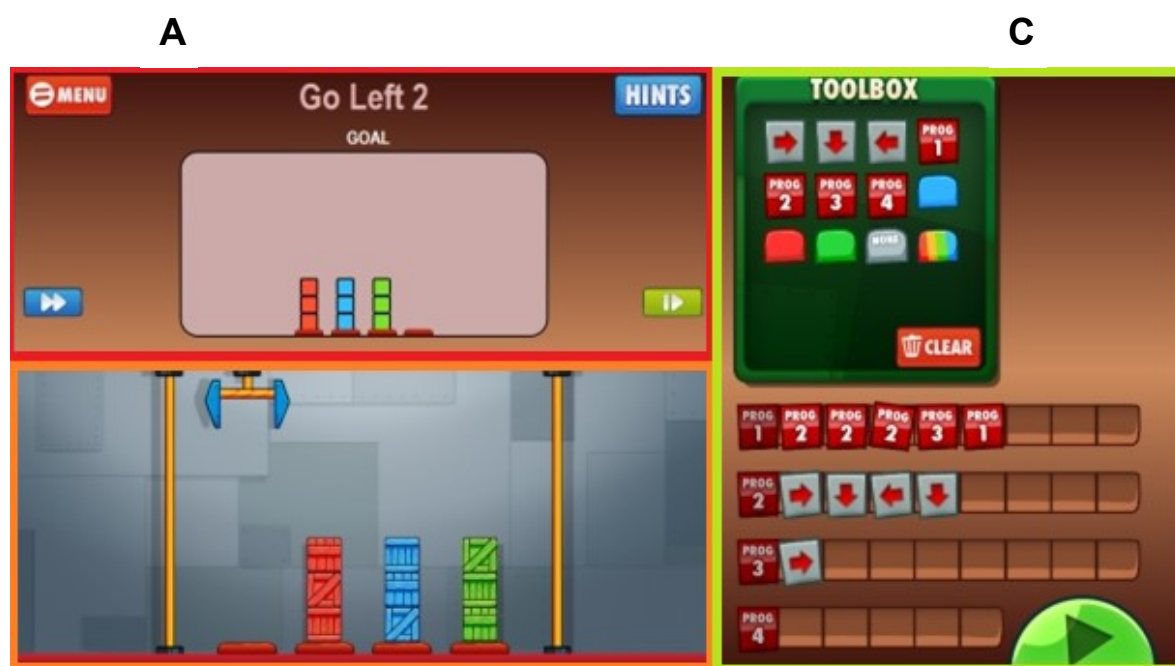
uma introdução prática e interativa ao pensamento computacional e à resolução de problemas por meio da programação.

Segundo Pavlus (2012), o jogo foi projetado para fazer com que a programação pareça não apenas factível, mas divertida: algo que você gostaria de fazer apenas para se divertir, não como um meio para atingir algum outro fim.

O Cargo Bot está disponível para download gratuito na App Store e também pode ser baixado na Microsoft Store para dispositivos Windows ou pode-se jogar direto no navegador.

Conforme mostra a Figura 5, a interface do jogo Cargo Bot é dividida em três áreas principais. Na parte superior da tela (A), encontra-se o objetivo do desafio, que mostra como as caixas devem ser organizadas. Na parte inferior (B), está o ambiente de execução, onde o braço robótico realiza as ações programadas. Na parte lateral direita da tela (C), localizam-se os blocos de comandos disponíveis, que o jogador pode arrastar e organizar em sequência lógica para programar os movimentos do robô. Essa organização permite ao jogador visualizar, testar e ajustar a lógica de sua programação de forma intuitiva e interativa.

Figura 5 - Tela do jogo Cargo Bot



B

Fonte: Próprio Autor (2025)

O jogo oferece cinco níveis de dificuldade: *Easy*, *Medium*, *Hard*, *Crazy* e *Impossible*. Ao escolher um nível, o jogador tem ainda seis opções diferentes de desafios, tornando cada etapa mais dinâmica e desafiadora.

Conforme a Tabela 4, o jogo Cargo Bot desenvolve as habilidades do pensamento computacional trabalhando os 4 pilares.

Tabela 4 - Cargo Bot: pilares do pilares do pensamento computacional

| Decomposição | Reconhecimento de Padrões | Abstração | Algoritmo |
|---|---|---|--|
| Os jogares tem que dividir a tarefa maior, em tarefas menores, como mover uma caixa para a esquerda, para a direita ou empilhá-la. Ao planejar cada movimento do guindaste, o jogador está, na prática, analisando cada parte do problema isoladamente para encontrar a solução mais eficiente. | Os níveis do jogo exigem que o jogador identifique padrões repetitivos nas tarefas, como movimentos de ida e volta do braço robótico. Ao identificar, o jogador pode reutilizar as sequencias de comandos em níveis diferentes e otimizar as soluções como loops. | O jogador precisa se concentrar nos aspectos mais relevantes, como a posição das caixas, os movimentos do guindaste e a sequência de comandos. Abstrair detalhes desnecessários e focar no cumprimento da tarefa. | O cumprimento das tarefas envolve organizar comandos de forma lógica e sequencial. Essa sequencias são os algoritmos simples que guiam o jogo. |

Fonte: Próprio autor (2025)

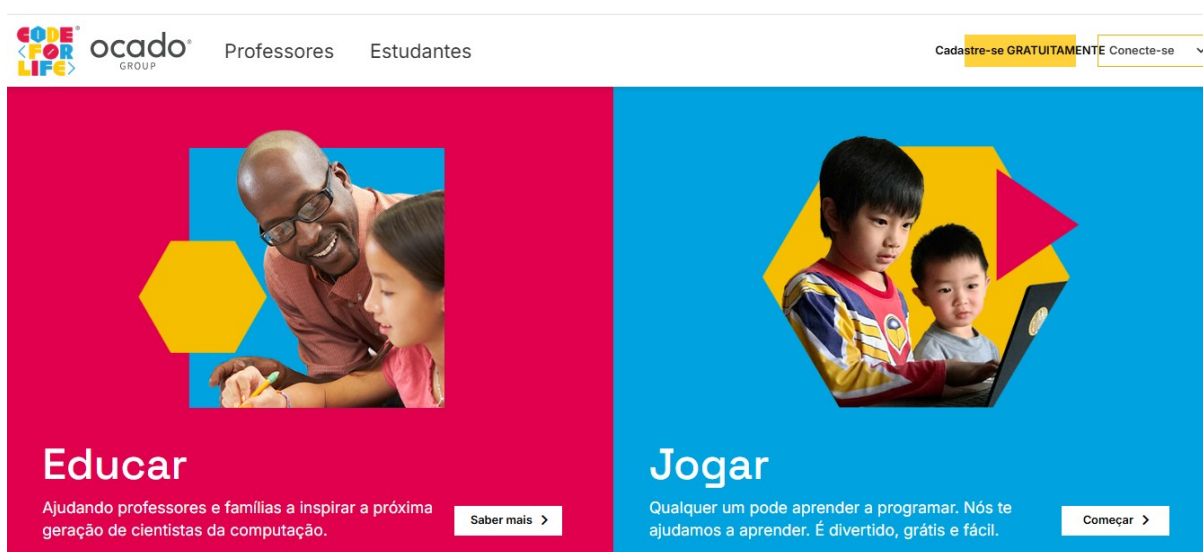
3.5.2.3 Code for Life

A plataforma Code for Life constitui-se como um recurso digital voltado para o ensino de programação e do pensamento computacional, desenvolvida pelo Cambridge Institute for Research, no Reino Unido. Trata-se de uma ferramenta gratuita e código aberto, acessível por meio da *internet*, que disponibiliza planos de ensino e aulas, guias do usuário e engajamento por meio de jogos educacionais, destinada ao público infante juvenil, mas que também pode ser explorado em diferentes níveis de ensino, servindo como ferramenta de apoio tanto para professores quanto para alunos. (CODE FOR LIFE, 2025).


O objetivo é ensinar os princípios básicos a alunos de programação, para ajudá-los a prosperar em um mundo cada vez mais digital (CODE FOR LIFE, 2025).

A Figura 6 ilustra a página inicial do Code for Life, onde o usuário precisa primeiramente se cadastrar como professor, aluno ou independente para utilizar os recursos da plataforma.

Figura 6 - Tela inicial do Code for Life



Dando a todos a capacidade de moldar o futuro da tecnologia



Apenas 16% dos graduados em ciência da computação universitária (2018/19) no Reino Unido eram mulheres, e queremos mudar isso.



A gamificação ajuda as crianças a aprender enquanto se divertem!



Isso mesmo, grátis para sempre: nosso presente para você! E também somos de código aberto.



O Code for Life tem mais de 660.000 usuários registrados no mundo todo.

[Sobre nós >](#)

Fonte: Plataforma Code for Life (2025)

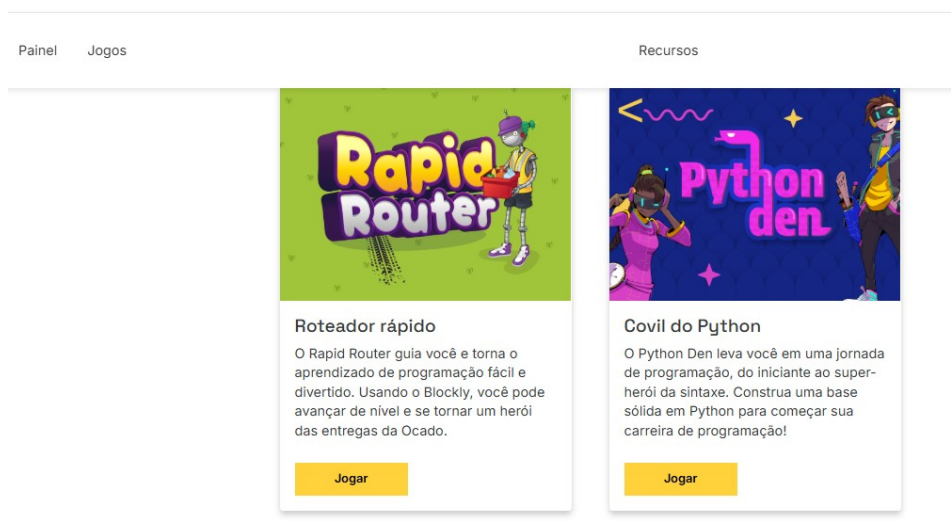
Entre os recursos disponibilizados pela plataforma, destaca-se o jogo *Rapid Router*, voltado para crianças, estruturado em uma interface de programação em blocos inspirada no *Blockly*. Essa estratégia pedagógica possibilita a introdução gradual de conceitos fundamentais de lógica e algoritmos sem a necessidade de

domínio prévio de linguagens de programação textuais. Posteriormente, o ambiente permite a transição para a linguagem Python, promovendo uma progressão no nível de complexidade das atividades e favorecendo o desenvolvimento de habilidades mais avançadas em programação.

Outro recurso é o jogo Python Den, destinado a alunos que já adquiriram conhecimentos iniciais em lógica computacional, ampliando suas competências por meio de desafios que exigem abstração, resolução de problemas e raciocínio algorítmico.

O jogo possibilita a transição dos blocos visuais para código em Python, introduzindo gradualmente a linguagem de programação. A medida que o aluno avança no jogo, escreve código Python puro, sem blocos em diferentes níveis de dificuldade, que envolvem variáveis, funções, repetições, condições e manipulação de dados.

Figura 7 - Tela de escolha do jogo da plataforma Code for Life



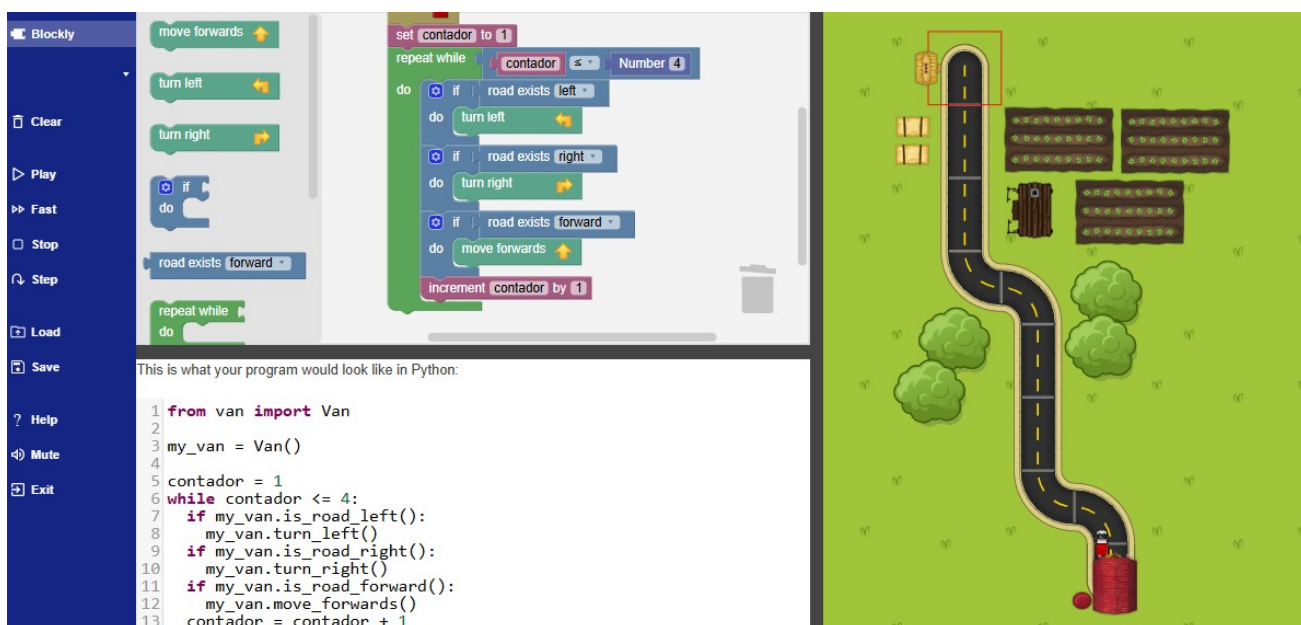
Fonte: Plataforma Code for Life (2025)

Entre as duas opções de jogos disponíveis na plataforma Code for Life, a escolha do jogo Python Den para a aplicação desta pesquisa justifica-se pelo fato de que os alunos já iriam exercitar as habilidades como lógica sequencial,

decomposição de problemas, abstração e pensamento algorítmico com a utilização dos jogos Scratch e CargoBot. Dessa forma, fez-se pertinente a utilização de um recurso que proporcionasse também a prática direta com a linguagem Python.

Na dinâmica do jogo, o participante controla um veículo em labirintos, devendo conduzi-lo até o objetivo estabelecido. Inicialmente, são utilizados blocos de programação que, automaticamente, são convertidos pelo próprio jogo em linguagem de programação Python, correspondendo a cada sequencia definido pelo jogador e visualizado em tempo real. À medida que o nível de complexidade aumenta, a opção de blocos deixa de estar disponível, sendo necessário o jogador utilizar apenas comandos escritos em Python. O código precisa ser estruturado de forma correta para que o objetivo seja alcançado, conforme ilustrado na Figura 8.

Figura 8 - Tela do jogo Python Den



Fonte: Próprio autor (2025)

Ao jogar Python Den o aluno desenvolve as habilidades do pensamento computacional trabalhando os 4 pilares, conforme ilustra a Tabela 5.

Tabela 5 -Code for Life: pilares do pensamento computacional

| Decomposição | Reconhecimento de Padrões | Abstração | Algoritmo |
|--|--|--|--|
| O jogo propõe problemas mais complexos que exigem dividir uma tarefa em passos menores. Para mover o veículo de um ponto inicial até o destino, o jogador precisa separar a tarefa em movimentos básicos (andar, virar, repetir, interagir com objetos). | Em fases sucessivas, os desafios apresentam situações semelhantes, mas com pequenas variações. Por exemplo é necessário virar à esquerda, avançar e virar à direita várias vezes. O jogo incentiva a reconhecer esses padrões e usar blocos de repetição (loops) para escrever o código de forma mais eficiente. | O jogador ignora detalhes não essenciais do ambiente e prioriza apenas as informações necessárias para resolver o desafio. Também ocorre quando o jogador introduz a criação de funções no Python, permitindo que o jogador agrupe um conjunto de instruções em um bloco reutilizável. O jogador passa a focar no que a função faz (o conceito) em vez de como cada comando interno é escrito. | Cada fase exige a escrita de códigos em Python, que nada mais são do que algoritmos. O jogador deve organizar as instruções em uma sequência lógica e correta, garantindo que o programa alcance o objetivo final. Com o avanço das fases, os algoritmos evoluem de comandos simples para estruturas mais complexas, como laços, condicionais e funções. |

Fonte: Próprio autor (2025)

Dessa forma, os jogos Scratch, Cargo Bot e Code for Life configuram-se como uma ferramenta pedagógica, ao integrar os fundamentos do pensamento computacional com elementos lúdicos e interativos, promovendo um ambiente propício à aprendizagem significativa da programação e da resolução de problemas complexos.

Conforme ilustra a Tabela 6, quadro comparativo dos pilares do pensamento computacional trabalhados em cada um dos três jogos.

Tabela 6 - Comparativo jogos x pilares do pensamento computacional

| Jogo | Decomposição | Reconhecimento de Padrões | Abstração | Algoritmos |
|-------------------|--|---|--|--|
| Scratch | Divide o projeto em partes, cenários, personagens e ações. | Identifica interações repetidas, usando <i>loops</i> | Foca na lógica sem se preocupar com a sintaxe. | Usa blocos em sequência lógica |
| Cargo Bot | Separa movimentos simples como mover e empilhar caixas. | Reconhece padrões de movimento e uso de <i>loops</i> . | Concentra-se nos elementos essenciais da tarefa. | Cria sequências simples e lógicas de comandos. |
| Python Den | Exige dividir a tarefa de mover o veículo em etapas menores (andar, virar, interagir). | Incentiva identificar repetições de movimentos e aplicar <i>loops</i> para otimizar o código. | Ensina a ignorar detalhes irrelevantes, focando na lógica e uso da linguagem Python. | Cada fase exige construir algoritmos em Python, que evoluem de simples sequências para estruturas complexas. |

Fonte: Próprio autor (2025)

A Tabela 7 ilustra quais os pilares mais fortes do pensamento computacional trabalhados em cada um dos três jogos.

Tabela 7 - Comparativo jogos x pilares mais fortes

| Jogo | Pilares mais Fortes | Observação |
|---------------|---------------------------------------|---|
| Code for Life | Decomposição e Algoritmo | Trabalha diretamente com Python, exigindo organizar instruções lógicas e dividir problemas em passos menores. |
| Scratch | Reconhecimento de Padrões e Abstração | Focado na criatividade e visualização; permite identificar padrões e criar blocos personalizados para generalizar soluções. |
| Cargo Bot | Decomposição e Abstração | Reforça a lógica sequencial; exige planejar passo a passo e criar funções que simplificam instruções repetidas. |

Fonte: Próprio autor (2025)

3.6 Aplicação da pesquisa

Todas as aulas da disciplina de Lógica e Linguagem de Programação foram realizadas no laboratório de informática da Escola Estadual Professor Murillo do Amaral, seguindo a grade curricular da disciplina, quatro aulas por semana no horário das 13h às 16:20h, sem intervalo. Cada aluno utilizou um *notebook* individualmente para a realização das atividades.

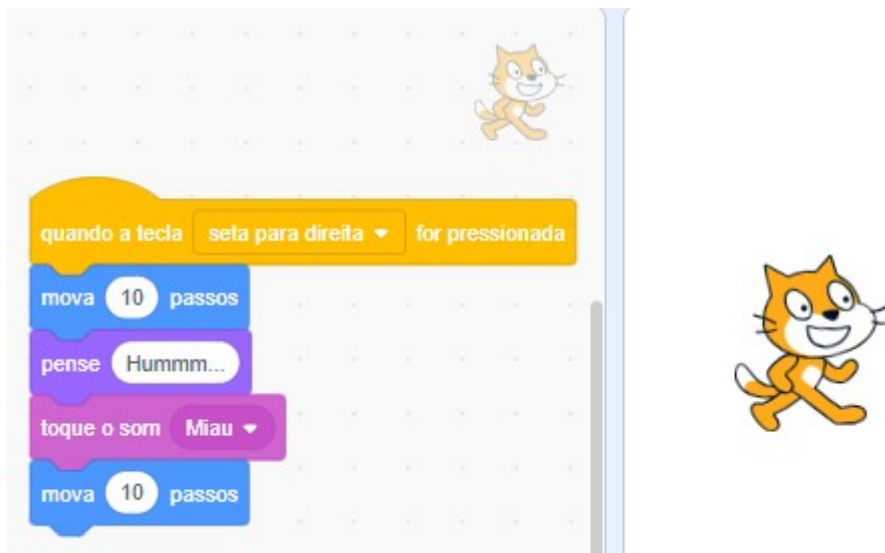
As aulas tiveram início no dia 6 de fevereiro de 2025, com conteúdo direcionado, inicialmente, para o desenvolvimento do raciocínio lógico e de Matemática básica, sem o uso de ferramentas computacionais.

A partir do dia 27 de fevereiro de 2025, foi introduzido o uso da plataforma Scratch. Os alunos já haviam tido contato prévio com a ferramenta no 9º ano, principalmente na criação de jogos, o que resultou um pouco de desinteresse. Diante disso, foi esclarecido que, nesta nova abordagem, o foco não seria a criação de jogos, mas sim o desenvolvimento de algoritmos com estrutura e lógica semelhantes aos utilizados na programação computacional. Também foi explicado o objetivo pedagógico de se trabalhar com o Scratch antes da introdução da linguagem Python: utilizar uma ferramenta visual e interativa para facilitar a compreensão de conceitos fundamentais da lógica de programação.

Realizou-se uma breve revisão dos principais elementos da plataforma, apresentando aos alunos exemplos básicos de como criar e programar um *sprite* – elemento gráfico interativo que pode ser programado para se mover, interagir com outros *sprites* e responder às ações do usuário.

A Figura 9 mostra um exemplo simples de *sprite*, utilizado em aula, que demonstra como os conceitos de lógica foram aplicados de forma prática e visual.

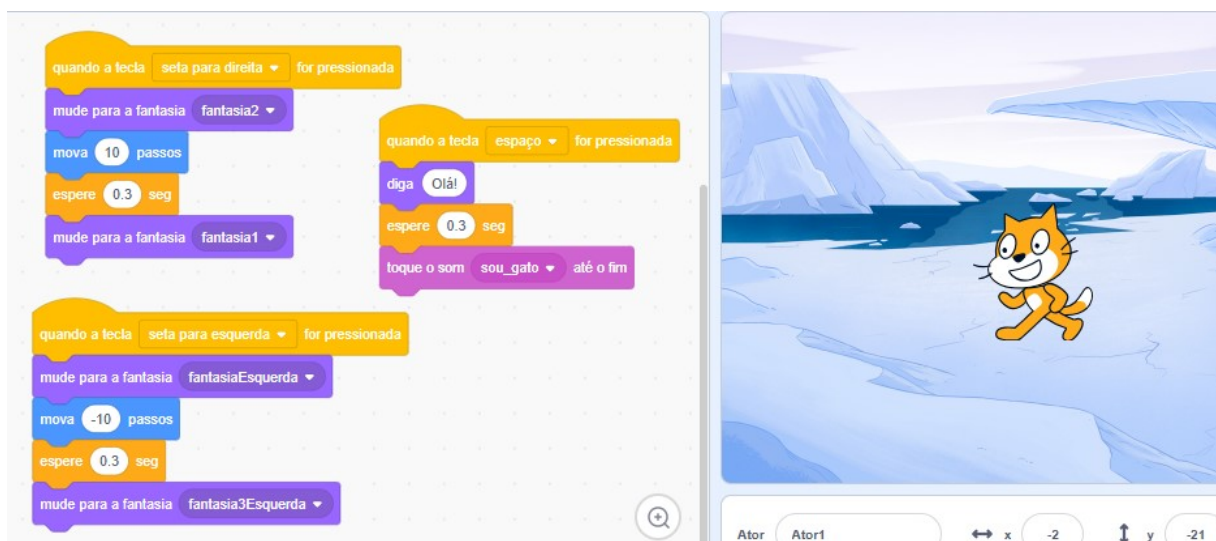
Figura 9 - Primeiro exemplo de *sprite*



Fonte: Próprio Autor (2025)

A Figura 10 ilustra um exemplo de *sprite* com mudança de cenário e mais ações do personagem.

Figura 10 - Segundo exemplo de *sprite*



Fonte: Próprio Autor (2025)

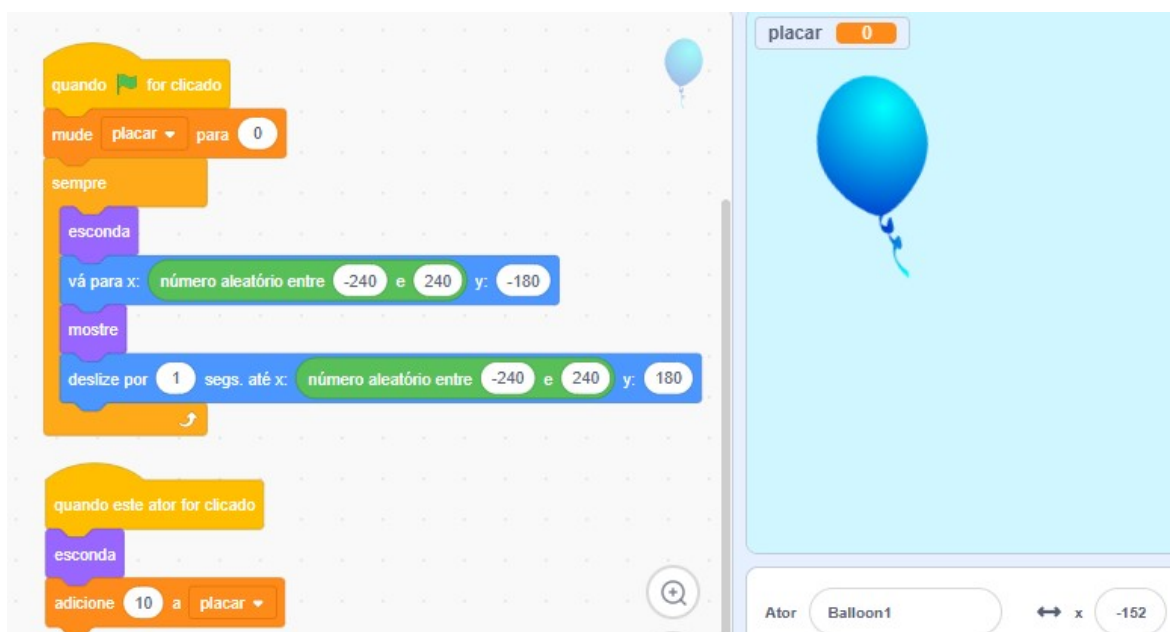
3.6.1 Estrutura Básica: Conceito de variáveis

Conforme Sebesta (2010, p. 104), uma variável é um objeto nomeado capaz de armazenar um valor que pode ser modificado durante a execução do programa. Essas variáveis, são armazenadas temporariamente na memória volátil do computador, memória *RAM (Random Access Memory)*. Isso significa que os dados associados às variáveis permanecem acessíveis enquanto o programa estiver em execução.

Seguindo a grade curricular da disciplina, a partir da aula realizada em 06 de março de 2025, utilizou-se o ambiente de programação Scratch como recurso didático para introduzir o conceito de variáveis, um elemento fundamental e amplamente utilizado na prática da programação.

A plataforma Scratch oferece funcionalidades que permitem a criação e manipulação de variáveis de forma visual e interativa, facilitando a compreensão desse conceito por parte dos alunos. A Figura 11 mostra a interface da ferramenta e exemplifica como as variáveis podem ser implementadas e utilizadas no desenvolvimento de algoritmos simples.

Figura 11 - Variáveis no Scratch

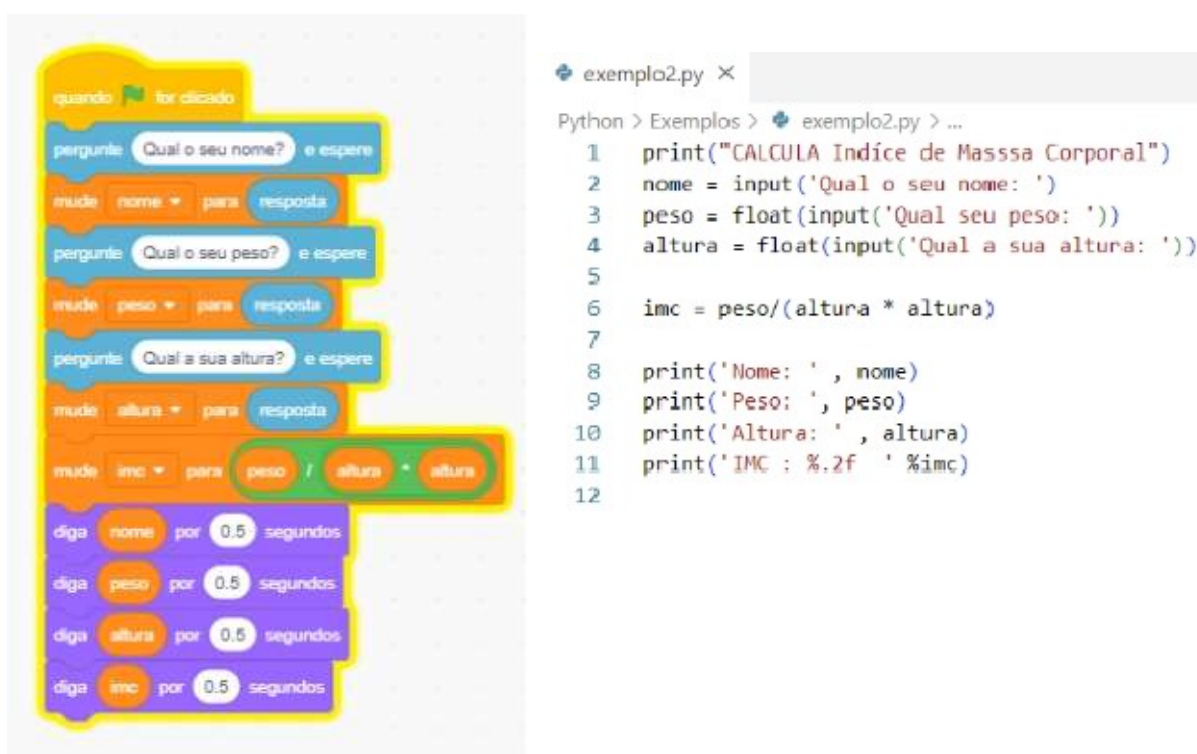


Fonte: Próprio Autor (2025)

A partir da introdução ao conceito de variáveis e da realização de exemplos práticos, as atividades desenvolvidas nas aulas da disciplina de Lógica e Linguagem de Programação passaram a incluir a construção de *sprites* na plataforma Scratch, bem como a criação de programas utilizando a linguagem Python. Primeiramente, os algoritmos eram elaborados no ambiente gráfico do Scratch, com o objetivo de facilitar a compreensão dos fundamentos da programação e estimular o desenvolvimento do pensamento computacional de forma visual e intuitiva.

Em seguida, os mesmos algoritmos foram transpostos para o ambiente textual de programação no Visual Studio Code, utilizando a linguagem Python, conforme mostra a Figura 12. Essa abordagem permitiu aos alunos estabelecerem conexões entre a lógica dos blocos visuais e a sintaxe textual.

Figura 12 - Algoritmo Scratch x Python



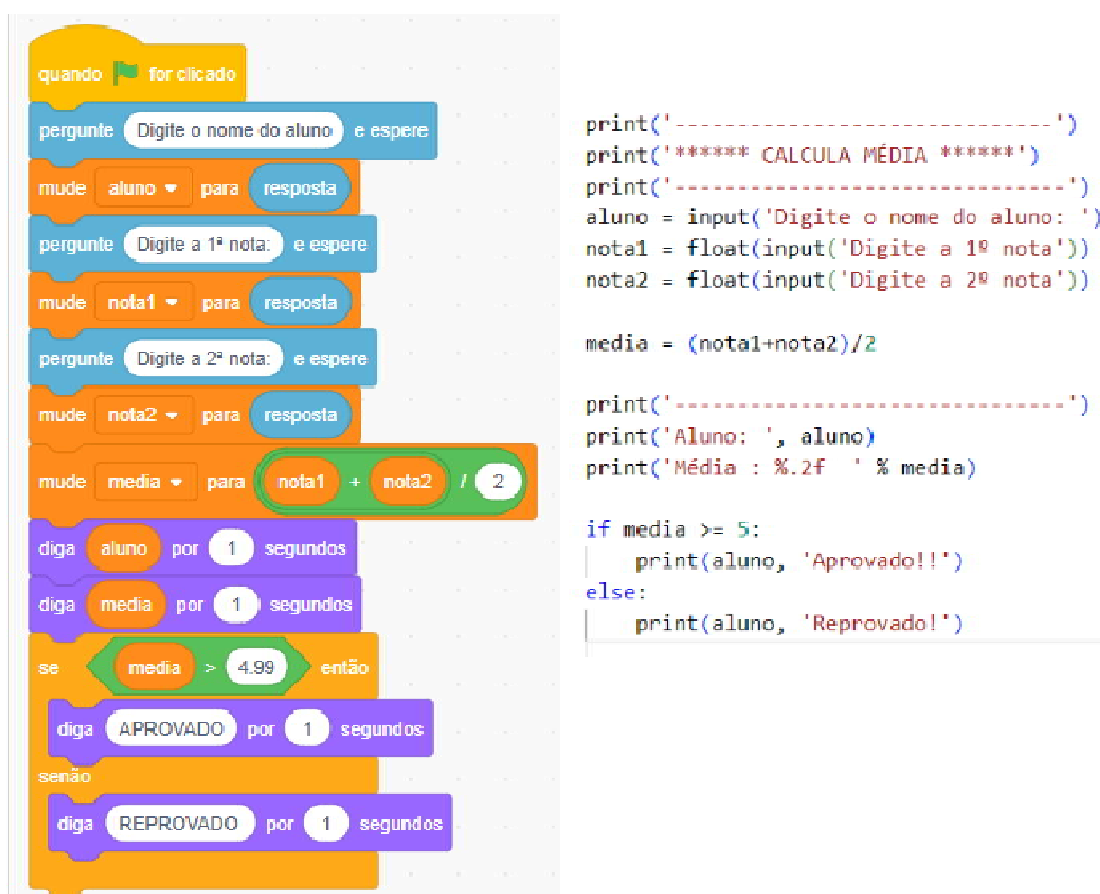
Fonte: Próprio Autor (2025)

3.6.2 Estrutura de Decisão

A estrutura de decisão é um dos pilares fundamentais da lógica de programação e permite que um programa tome diferentes caminhos ou execute diferentes blocos de código com base em condições lógicas. Em outras palavras, ela possibilita que o programa "decida" o que fazer dependendo do resultado de uma verificação (condição).

A partir da aula realizada no dia 15 de maio de 2025, foi introduzido o conceito de estrutura de decisão. No ambiente Scratch, esse conceito é representado pelo bloco "se – senão". Após a fundamentação teórica de forma lúdica através do Scratch, o mesmo conceito foi apresentado por meio do comando correspondente na linguagem de programação Python, comando "if-else", conforme mostra a Figura 13.

Figura 13 - Exemplo de estrutura de decisão



The figure displays two examples of decision structures. On the left is a Scratch script, and on the right is the equivalent Python code.

Scratch Script:

- quando a bandeira verde for clicada
- pergunte "Digite o nome do aluno" e espere
- mude "aluno" para "resposta"
- pergunte "Digite a 1ª nota:" e espere
- mude "nota1" para "resposta"
- pergunte "Digite a 2ª nota:" e espere
- mude "nota2" para "resposta"
- mude "media" para $(\text{nota1} + \text{nota2}) / 2$
- diga "aluno" por 1 segundos
- diga "media" por 1 segundos
- se "media" > 4.99 então
 - diga "APROVADO" por 1 segundos
- senão
 - diga "REPROVADO" por 1 segundos

Python Code:

```
print('-----')
print('***** CALCULA MÉDIA *****')
print('-----')
aluno = input('Digite o nome do aluno: ')
nota1 = float(input('Digite a 1ª nota'))
nota2 = float(input('Digite a 2ª nota'))

media = (nota1+nota2)/2

print('-----')
print('Aluno: ', aluno)
print('Média : %.2f ' % media)

if media >= 5:
    print(aluno, 'Aprovado!!')
else:
    print(aluno, 'Reprovado!')
```

Fonte: Próprio Autor (2025)

Foram apresentados aos alunos alguns exemplos, semelhantes ao exposto anteriormente, utilizando o Scratch e a linguagem de programação Python.

3.6.3 Estrutura de Repetição

A estrutura de repetição também denominada laço ou *loop*, constitui um recurso fundamental da programação que possibilita a execução de um mesmo bloco de instruções de forma contínua, até que uma condição previamente estabelecida seja satisfeita. Seu uso é indispensável quando há necessidade de automatizar tarefas repetitivas, reduzir redundâncias no código e otimizar o tempo de processamento. Entre os principais tipos de estruturas de repetição destacam-se os laços condicionais (como *while* e *do-while*) e os laços contados (como *for*).

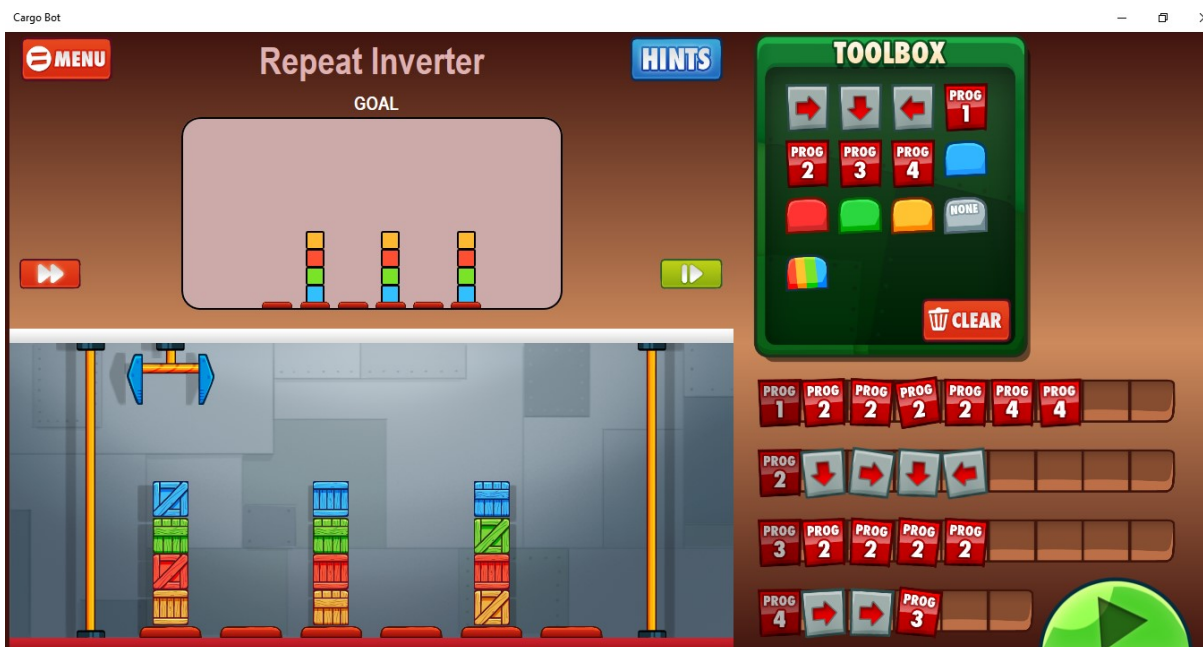
No processo de ensino e aprendizagem de programação, a compreensão deste conceito desempenha papel essencial, pois favorece o desenvolvimento do pensamento algorítmico e da lógica computacional.

Na aula realizada no dia 07 de agosto de 2025, foi introduzido o conceito de estrutura de repetição. Apresentou-se aos alunos o jogo Cargo Bot, explicou-se a dinâmica do jogo e foi definido o objetivo da aula deste dia, jogar as seis fases do nível *Easy* e avançar até o nível *Medium* para a próxima aula.

A escolha do Cargo Bot para a aprendizagem de estrutura de repetição justifica-se pelo fato de que, para atingir as metas propostas, o jogador precisa elaborar sequências lógicas compostas por múltiplas instruções repetidas. Dessa forma, o jogo favorece a compreensão prática da lógica de repetição no contexto da programação.

As instruções ordenadas ao robô são organizadas em etapas numeradas e podem ser repetidas quantas vezes forem necessárias, desde que respeitado o limite estabelecido na linha do "PROG1", conforme ilustrado na Figura 14.

Figura 14 - Carbo Bot: estrutura de repetição



Fonte: Próprio Autor (2025)

No mesmo dia, os alunos também tiveram contato com o conceito de estrutura de repetição por meio da ferramenta Scratch. Após a utilização do jogo Cargo Bot, utilizou-se a ferramenta Scratch, através de blocos de repetição, assim tornou-se o conceito mais fácil, favorecendo a compreensão e assimilação do conteúdo.

A Figura 15 ilustra o uso do bloco de programação repita aplicado a ferramenta Scratch.

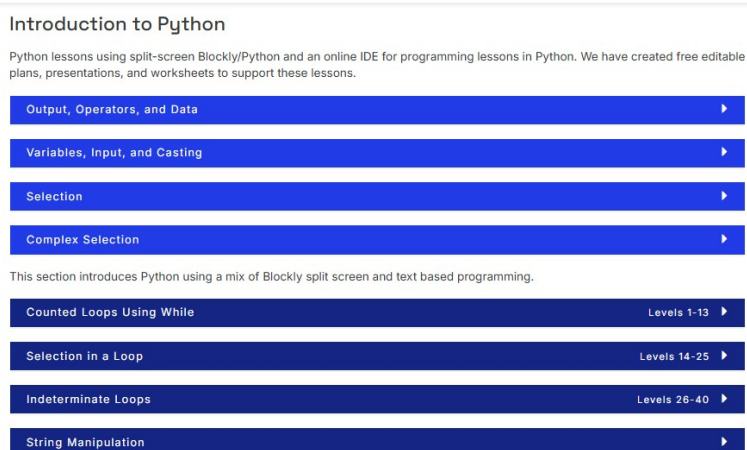
Figura 15 - Scratch: estrutura de repetição



Fonte: Próprio Autor (2025)

Nas aulas seguintes, foi introduzido o jogo Python Den, da plataforma Code for Life, também utilizado para trabalhar estrutura de repetição em blocos de programação e posteriormente a linguagem de programação Python. O jogo oferece vários níveis separados por conteúdo, conforme ilustra a Figura 16.

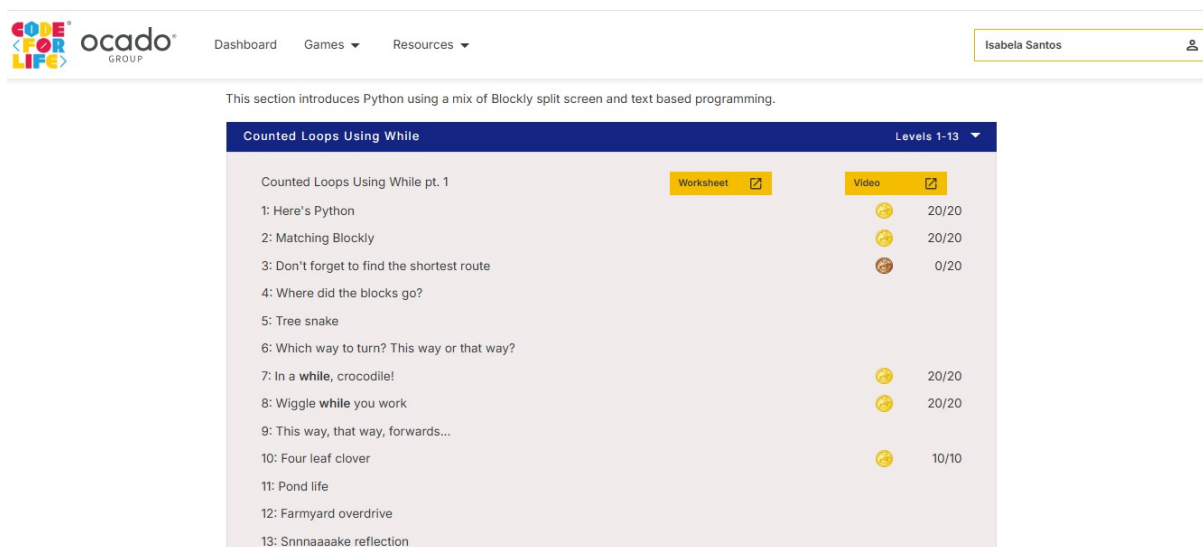
Figura 16 - Python Den : conteúdo programático



Fonte: Plataforma Code for Life (2025)

Cada conteúdo é dividido em vários níveis, iniciou-se o uso do jogo Python Den na sessão *Counted Loops Using While* nível 1, conforme ilustra a Figura 17.

Figura 17 - Python Den : divisão de níveis



The screenshot shows the Code for Life platform interface. At the top, there are navigation links for 'Dashboard', 'Games', and 'Resources', along with a user profile for 'Isabela Santos'. Below the navigation, a section titled 'Counted Loops Using While' is displayed, with a dropdown menu for 'Levels 1-13'. The main content area lists 13 levels with their respective completion status:

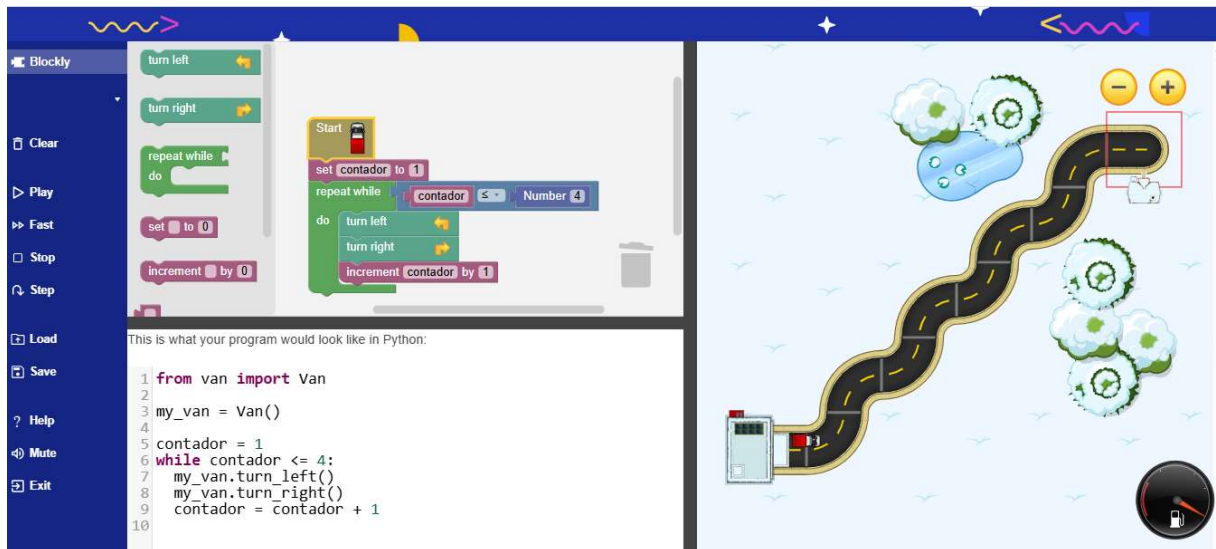
| Level | Completion Status |
|---|-------------------|
| 1: Here's Python | 20/20 |
| 2: Matching Blockly | 20/20 |
| 3: Don't forget to find the shortest route | 0/20 |
| 4: Where did the blocks go? | |
| 5: Tree snake | |
| 6: Which way to turn? This way or that way? | |
| 7: In a while , crocodile! | 20/20 |
| 8: Wiggle while you work | 20/20 |
| 9: This way, that way, forwards... | |
| 10: Four leaf clover | 10/10 |
| 11: Pond life | |
| 12: Farmyard overdrive | |
| 13: Snnnaaaake reflection | |

Fonte: Plataforma Code for Life (2025)

Nesta seção, o aluno explora o conceito de estruturas de repetição, nas quais determina previamente quantas vezes uma instrução ou conjunto de instruções será executado.

Conforme ilustra a Figura 18, tela do jogo nível 8; o aluno utiliza bloco de programação com estrutura de repetição (bloco *repeat while*) para mover o veículo até alcançar o objetivo. Conforme move os blocos, automaticamente, são convertidos pelo próprio jogo em linguagem de programação Python, correspondendo a cada sequência definido pelo jogador e visualizado em tempo real.

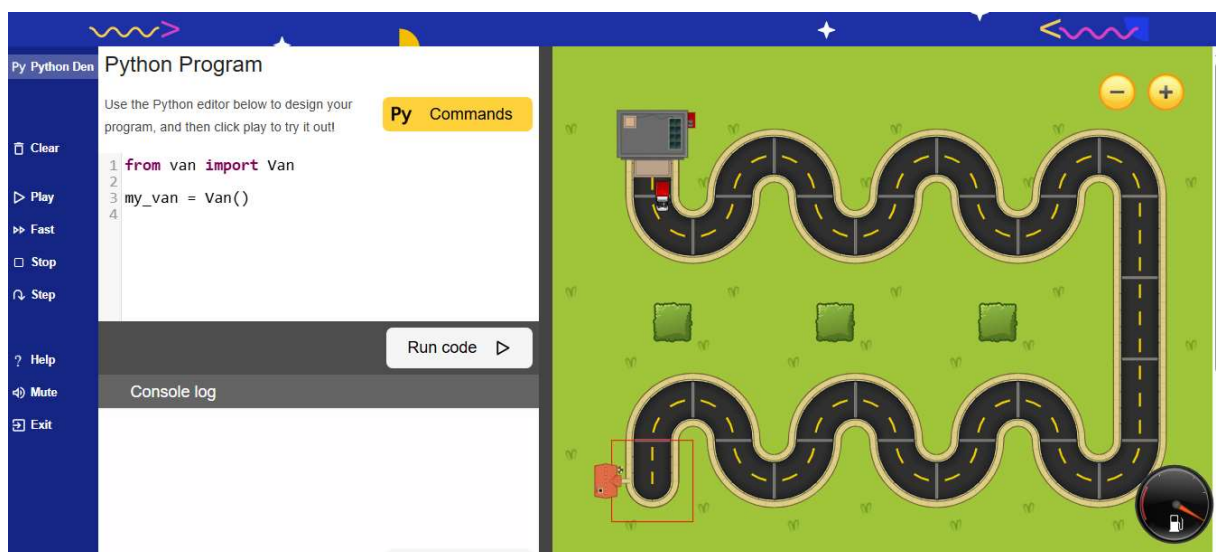
Figura 18 - Python Den: tela do jogo nível 8



Fonte: Próprio Autor (2025)

Ao avançar os níveis, o jogador deixa de ter a opção de programação em blocos, passando a programar exclusivamente em Python, conforme ilustrado na Figura 19, que apresenta a tela do jogo nível 13.

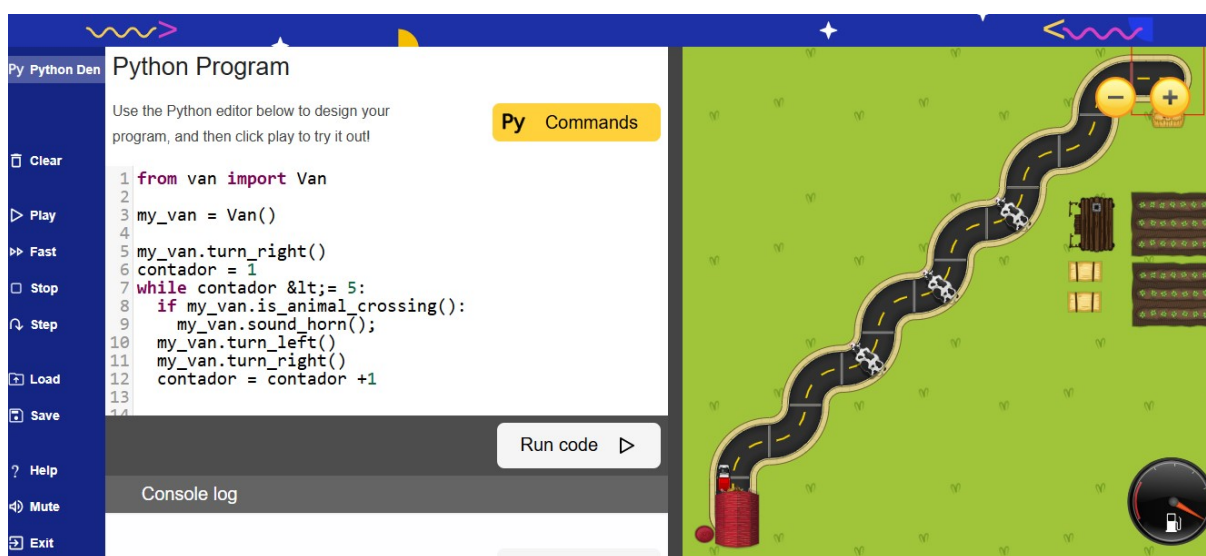
Figura 19 - Python Den: tela do jogo nível 13



Fonte: Próprio Autor (2025)

Nas aulas seguintes, trabalhou-se com os alunos a seção *Selection in a Loop*, que explora o conceito de estrutura de decisão inserida em uma estrutura de repetição, conforme ilustrado na Figura 20.

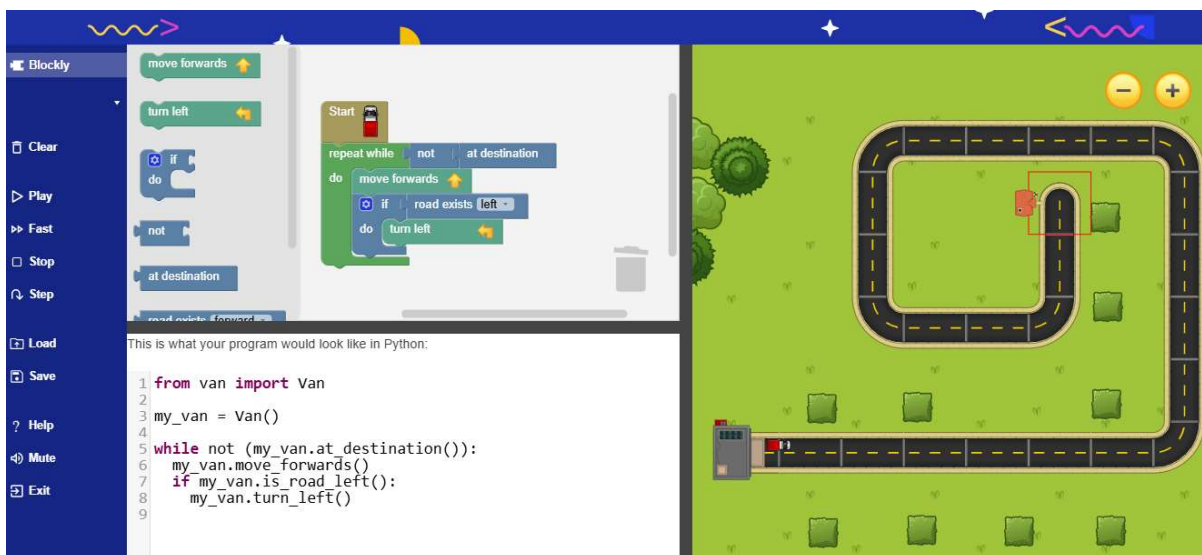
Figura 20 - Python Den: tela do jogo nível 16



Fonte: Próprio Autor (2025)

E a seção *Indeterminate Loop* trabalha o conceito de repetição em que o número de execuções não é definido antecipadamente, sendo controlado por uma condição lógica, conforme ilustra a figura 21.

Figura 21 – Python Den - Tela do jogo nível 31



Fonte: Próprio Autor (2025)

Apresenta-se a seguir a Figura 22, registro fotográfico dos alunos durante a aula da disciplina Lógica e Linguagem de Programação, realizada no laboratório de Informática da Escola Estadual Professor Murillo do Amaral, na qual participaram da atividade com o jogo Python Den.

Figura 22 – Alunos jogando Python Den



Fonte: Próprio Autor (2025)

Conforme ilustra a Figura 23, mais um registro fotográfico dos alunos durante a aula da disciplina Lógica e Linguagem de Programação, realizada no laboratório de Informática da Escola Estadual Professor Murillo do Amaral.

Figura 23 – Alunos jogando Python Den



Fonte: Próprio Autor (2025)

Ao dominar estrutura de repetição, o aluno passa a reconhecer padrões de repetição em problemas reais, conseguindo traduzi-los em soluções computacionais mais eficientes e escaláveis. Além disso, o aprendizado dos laços contribui para a internalização de conceitos mais avançados, como a manipulação de coleções de dados, o processamento de grandes volumes de informações e a construção de algoritmos de maior complexidade.

A estrutura de repetição foi trabalhada nos três jogos: Scratch, Cargo Bot e Python Den. A Figura 24 demonstra a aplicação da estrutura de repetição no ambiente Scratch.

Figura 24 – Estrutura de repetição: Scratch



Fonte: Próprio Autor (2025)

Na sequência, utilizou-se o editor de código Visual Studio Code para o estudo das estruturas de repetição na linguagem de programação Python. Conforme ilustra a Figura 25, estabeleceu-se uma comparação com o *sprite* previamente desenvolvido no Scratch, para maior entendimento do aluno.

Figura 25 – Estrutura de repetição: Scratch x Python



```

exemplo11_while.py > ...
1  numero = int(input('Digite um número: '))
2  while (numero != 0):
3      print(numero)
4      numero -= 1
5

```

Fonte: Próprio Autor (2025)

3.6.4 Listas

O último conceito trabalhado nesta pesquisa com os alunos foi Listas. A aprendizagem e o estudo de listas em Python são fundamentais para a formação de qualquer programador.

Em Python, uma lista é uma estrutura que permite armazenar vários valores em uma única variável. Esses valores podem ser de diferentes tipos (números, textos, booleanos, etc.), e cada elemento pode ser acessado por meio de um índice.

A Figura 26 apresenta um exemplo de criação de lista na linguagem de programação Python.

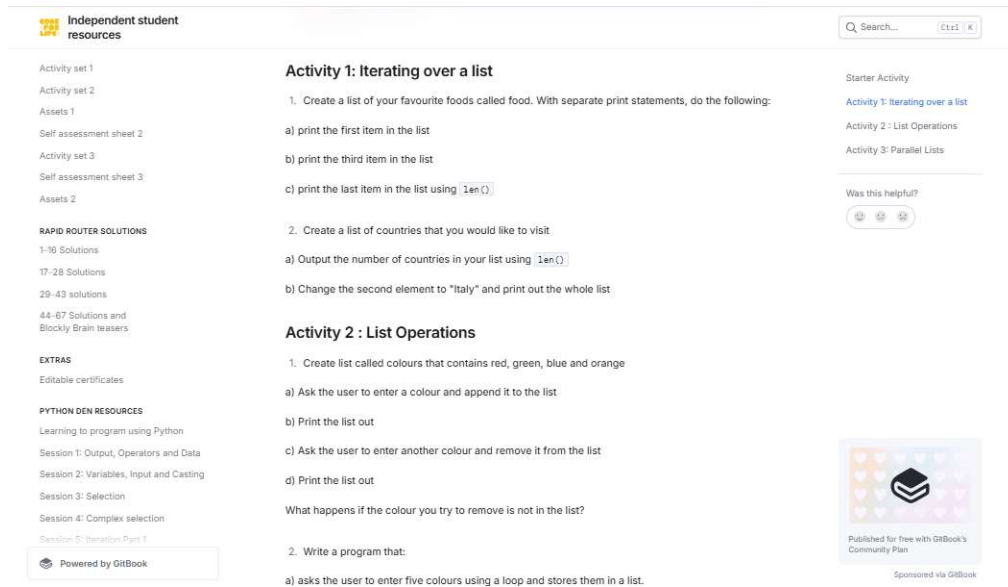
Figura 26 – Exemplo Lista

```
exemplo12_lista.py > ...  
1  # EXEMPLO LISTAS  
2  lista_frutas = ['morango', 'laranja', 'maçã', 'uva', 'caqui', 'limão']  
3  print(lista_frutas)  
4  
5  print('-----')  
6  print('          LISTA DE FRUTAS          ')  
7  print('-----')  
8  indice = 0  
9  while indice < len(lista_frutas):  
10 |     print(lista_frutas[indice])  
11 |     indice += 1  
12  print('-----')
```

Fonte: Próprio Autor (2025)

Utilizou-se a plataforma Code for Life, o jogo Python Den para trabalhar com Listas, na seção *Lists*. Nesta seção não há jogos apenas a opção *Work/sheet* e Video disponível. Trabalhou-se no editor de código Visual Studio Code com as atividades desta seção, conforme ilustra Figura 27.

Figura 27 – Seção Lists – Worksheet



The screenshot shows a GitBook page for 'Independent student resources'. The page is divided into three main sections: 'Activity 1: Iterating over a list', 'Activity 2: List Operations', and a sidebar with navigation links. The main content includes two activities with numbered tasks and sub-tasks. Activity 1 involves creating a list of favorite foods and printing specific items. Activity 2 involves creating a list of countries and performing operations like counting and modifying elements. The sidebar lists various resources like 'Activity set 1', 'Assets 1', and 'Rapid Router Solutions'. A search bar is visible at the top right, and a 'Was this helpful?' feedback section is at the bottom right.

Independent student resources

Search...

Activity set 1
 Activity set 2
 Assets 1
 Self assessment sheet 2
 Activity set 3
 Self assessment sheet 3
 Assets 2

RAPID ROUTER SOLUTIONS
 1-16 Solutions
 17-28 Solutions
 29-43 solutions
 44-67 Solutions and Blockly Brain teasers

EXTRAS
 Editable certificates

PYTHON DEN RESOURCES
 Learning to program using Python
 Session 1: Output, Operators and Data
 Session 2: Variables, Input and Casting
 Session 3: Selection
 Session 4: Complex selection
 Session 5: Iteration Part 1

Powered by GitBook

Activity 1: Iterating over a list

1. Create a list of your favourite foods called food. With separate print statements, do the following:

- print the first item in the list
- print the third item in the list
- print the last item in the list using `len()`

2. Create a list of countries that you would like to visit

- Output the number of countries in your list using `len()`
- Change the second element to "Italy" and print out the whole list

Activity 2 : List Operations

1. Create list called colours that contains red, green, blue and orange

- Ask the user to enter a colour and append it to the list
- Print the list out
- Ask the user to enter another colour and remove it from the list
- Print the list out

What happens if the colour you try to remove is not in the list?

2. Write a program that:

- asks the user to enter five colours using a loop and stores them in a list.

Starter Activity
 Activity 1: Iterating over a list
 Activity 2 : List Operations
 Activity 3: Parallel Lists

Was this helpful?
 👍 👎

Published for free with GitBook's Community Plan
 Sponsored via GitBook

Fonte: Próprio Autor (2025)

4. Coleta de Dados

Este capítulo apresenta os resultados dos instrumentos de coleta de dados desta pesquisa. São descritos os resultados das avaliações práticas e questionário aplicado aos alunos.

4.1 Instrumentos de Coleta

Os instrumentos de coleta de dados utilizados nesta pesquisa foram: questionário aplicado aos alunos, observação do pesquisador, atividades e avaliações práticas.

Para a aplicação do questionário aos alunos menores de idade, a pesquisa foi submetida à apreciação do Comitê de Ética em Pesquisa (CEP) do Centro Universitário Teresa D'Ávila – UNIFATEA. Após análise, o projeto foi aprovado sob o parecer consubstanciado nº 7.620.452, garantindo que todos os procedimentos metodológicos respeitassem os princípios éticos, incluindo o respeito à autonomia, o sigilo das informações e o uso do Termo de Consentimento Livre e Esclarecido (TCLE), disponível no Apêndice A.

Quatorze responsáveis pelos alunos assinaram o TCLE, garantindo que compreendiam os objetivos da pesquisa, seus direitos e a confidencialidade das informações fornecidas bem como todos os quatorze alunos menores assinaram o Termo de Assentimento Livre e Esclarecido (TALE), disponível no Apêndice B. Somente após essa etapa, os participantes responderam ao questionário, disponível no Apêndice C, que foi elaborado através da ferramenta *Google Forms* e disponibilizado ao final da pesquisa.

Durante as aulas da disciplina de Lógica e Linguagem de Programação as atividades práticas realizadas pelos alunos foram documentadas. Foram aplicadas três avaliações práticas durante o período da pesquisa com o objetivo de verificar o desenvolvimento das competências trabalhadas.

Os dados qualitativos coletados por meio da observação, do relato do pesquisador e das produções dos alunos foram analisados com base na técnica de análise de conteúdo, utilizando a categorização temática como método de interpretação.

4.2 Análise dos Dados: Avaliações Práticas

Foram aplicadas três avaliações práticas durante a pesquisa. A primeira no 2º bimestre, disponível no Apêndice D, treze alunos participaram e um aluno faltou. O conteúdo trabalhado foi o conceito de estrutura de decisão. A avaliação contou com três questões, que deveriam ser resolvidas em dois ambientes distintos: inicialmente no Scratch, por meio de blocos de programação, e posteriormente em Python, utilizando o editor de código Visual Studio Code.

Dessa forma, buscou-se avaliar a compreensão dos alunos tanto na lógica visual quanto na aplicação da mesma estrutura em uma linguagem de programação. Essa abordagem teve como propósito favorecer a transição gradual do pensamento computacional visual para o textual.

Verifica-se que os alunos alcançaram bons resultados em ambos os ambientes de programação, em bloco: Scratch e textual: Python, conforme apresentado na Tabela 8.

Tabela 8 – Comparativo de acertos *Scratch X Python*

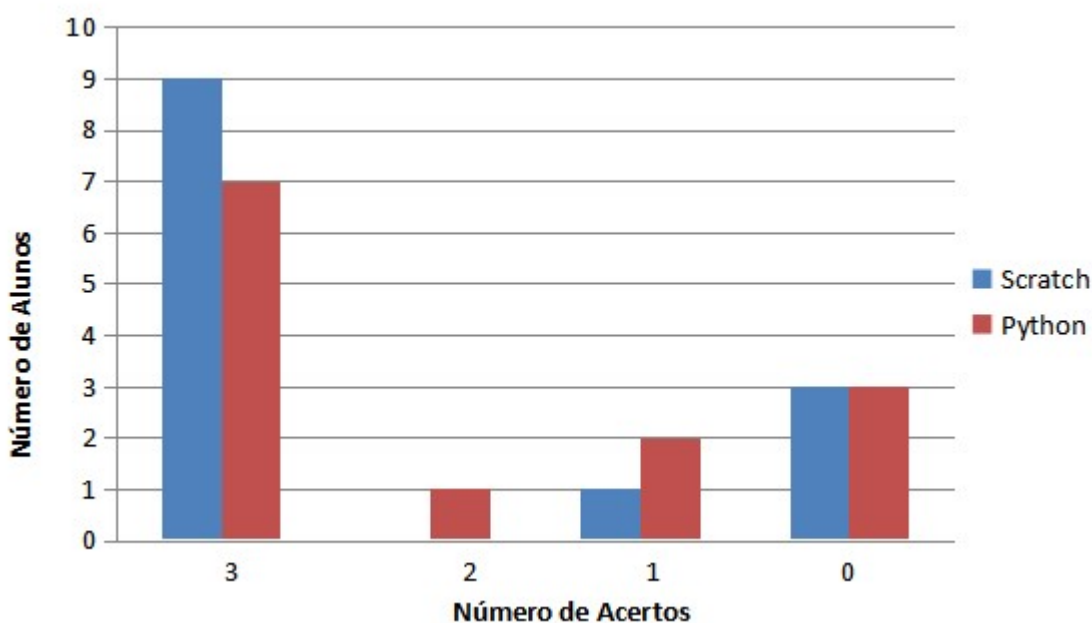
| Número de Acertos | Número de Alunos (13 alunos presentes e 1 aluno faltou) | |
|-------------------|--|--------|
| | Scratch | Python |
| 3 | 9 | 7 |
| 2 | 0 | 1 |
| 1 | 1 | 2 |
| 0 | 3 | 3 |

Fonte: Próprio Autor (2025)

Observa-se que a maioria dos alunos alcançou o desempenho máximo (três acertos) em ambos os ambientes — nove alunos no Scratch e sete no Python, evidenciando que grande parte da turma conseguiu compreender e aplicar os conceitos de programação propostos nas atividades, assimilar dos conteúdos e realizar a transição entre a programação em blocos e a programação textual.

Conforme apresentado no Gráfico 1, os resultados da avaliação indicam que os alunos obtiveram um desempenho satisfatório, devido a quantidade de acertos ser maior ou igual a dois acertos por ambiente de programação, representando mais da metade dos itens avaliados.

Gráfico 1– Resultado avaliação 2º bimestre



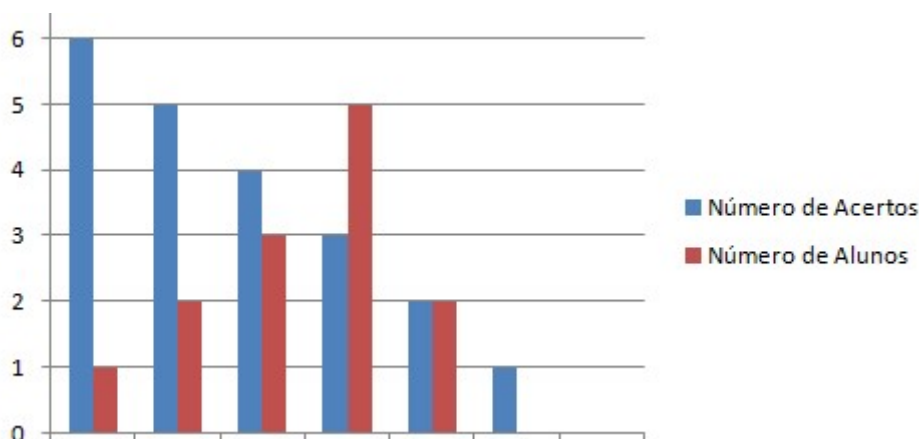
Fonte: Próprio Autor (2025)

A segunda avaliação prática, disponível no Apêndice E, foi aplicada no 3º bimestre, treze alunos participaram e um aluno faltou, na qual foi trabalhado o conceito de estrutura de repetição. A avaliação foi composta por seis questões, todas desenvolvidas na linguagem de programação Python, utilizando o editor de código Visual Studio Code. Essa avaliação teve como finalidade consolidar o entendimento dos laços de repetição pelo qual foi trabalhado o raciocínio lógico no ambiente Scratch, jogo Cargo Bot e jogo Python Den.

Embora o conteúdo referente à estrutura de repetição exija um raciocínio lógico mais elaborado e complexo, o resultado da avaliação manteve-se satisfatório,

pois de seis questões aplicadas na avaliação, onze alunos acertaram três questões ou mais, conforme apresentado no Gráfico 2.

Gráfico 2– Resultado avaliação 3º bimestre



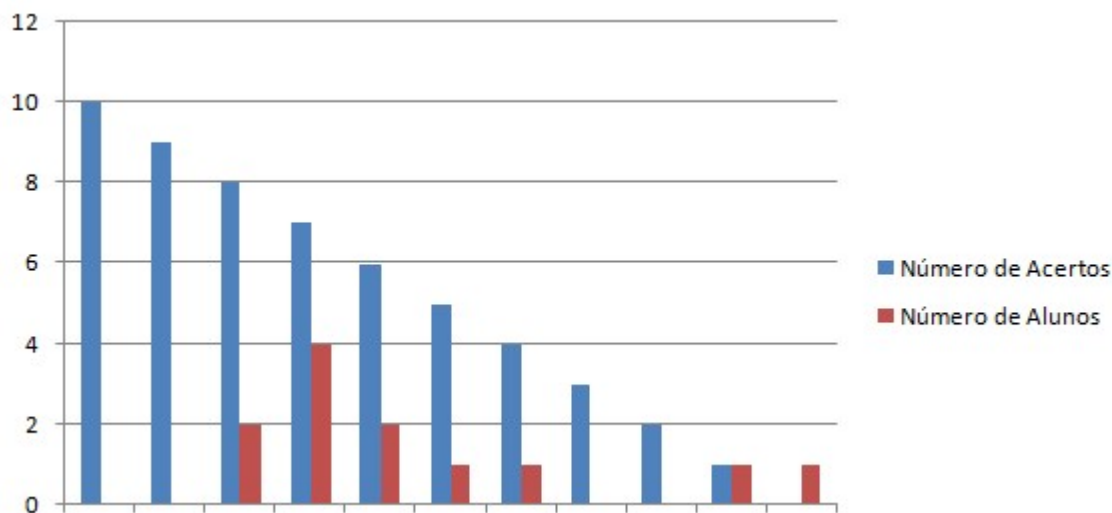
Fonte: Próprio Autor (2025)

A terceira avaliação prática, disponível no Apêndice F, foi realizada no 4º bimestre, doze alunos participaram e dois alunos faltaram. Foram trabalhados os conceitos de estrutura de decisão, repetição e listas, exigindo ainda mais do aluno um maior nível de complexidade e raciocínio lógico mais elaborado.

A avaliação foi composta por dez questões, todas desenvolvidas na linguagem de programação Python, utilizando o editor de código Visual Studio Code.

Conforme apresentado no Gráfico 3, os resultados da avaliação indicam que os alunos alcançaram um desempenho satisfatório. Dos doze participantes, nove acertaram cinco ou mais questões, enquanto apenas três obtiveram menos de cinco acertos, valor correspondente a metade da avaliação. Esse resultado demonstra que a maioria conseguiu compreender e aplicar adequadamente as estruturas e comandos da linguagem Python, indicando uma assimilação dos conteúdos.

Gráfico 3– Resultado avaliação 4º bimestre



Fonte: Próprio Autor (2025)

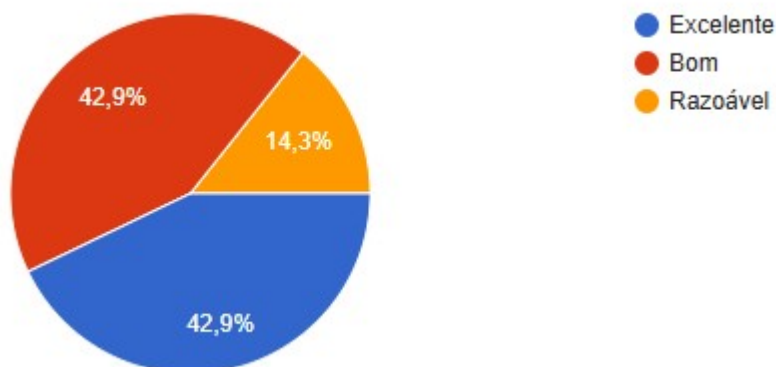
4.3 Análise dos Dados: Questionário

Os quatorze alunos receberam o convite para acesso ao questionário do *GoogleForms* por *e-mail*, e utilizaram os computadores do laboratório de Informática da escola estadual Professor Murillo do Amaral para responder durante o horário de aula.

As perguntas foram formuladas de maneira clara e objetiva, sendo sete perguntas fechadas e uma pergunta aberta.

A primeira pergunta realizada para os alunos foi: “Como você avalia o jogo digital Scratch?”, cujas respostas podem ser observadas no Gráfico 4.

Gráfico 4 - Como você avalia o jogo digital Scratch?

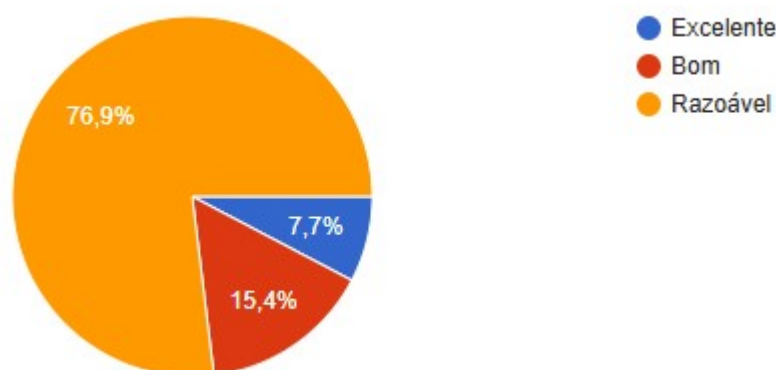


Fonte: Próprio Autor (2025)

Sobre a avaliação dos alunos em relação ao Scratch, 42,9% dos alunos responderam “Excelente”, 42,9% responderam “Bom” e apenas 14,3% dos alunos responderam “Razoável”. Esses resultados indicam uma avaliação positiva do jogo como ferramenta de apoio ao ensino e à aprendizagem de lógica e programação, devido o jogo ser simples, fácil e lúdico.

A segunda pergunta realizada para os alunos foi: “Como você avalia o jogo digital Cargo Bot?”, cujas respostas podem ser observadas no Gráfico 5.

Gráfico 5 - Como você avalia o jogo digital Cargo Bot?

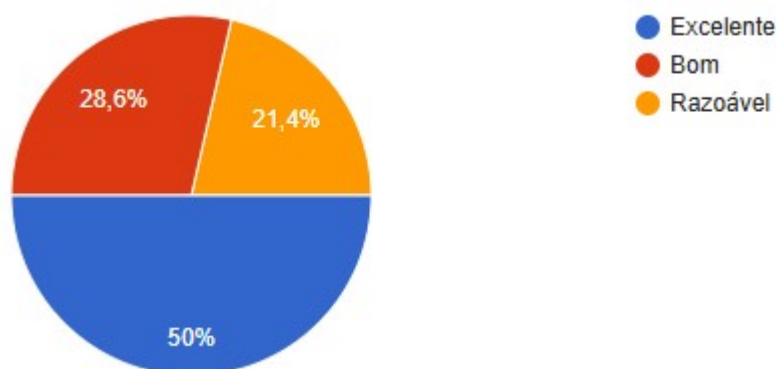


Fonte: Próprio Autor (2025)

Apenas 7,7% dos alunos avaliaram o jogo Cargo Bot como “Excelente”, 15,4% o classificaram como “Bom” e 76,9% o consideraram “Razoável”, resultado possivelmente associado ao nível de complexidade, à dificuldade e ao raciocínio lógico exigidos pelo jogo.

A terceira pergunta realizada para os alunos foi: “Como você avalia o jogo digital Python Den da plataforma Code for Life?”, cujas respostas podem ser observadas no Gráfico 6.

Gráfico 6 - Como você avalia o jogo digital *Python Den* da plataforma Code for Life ?



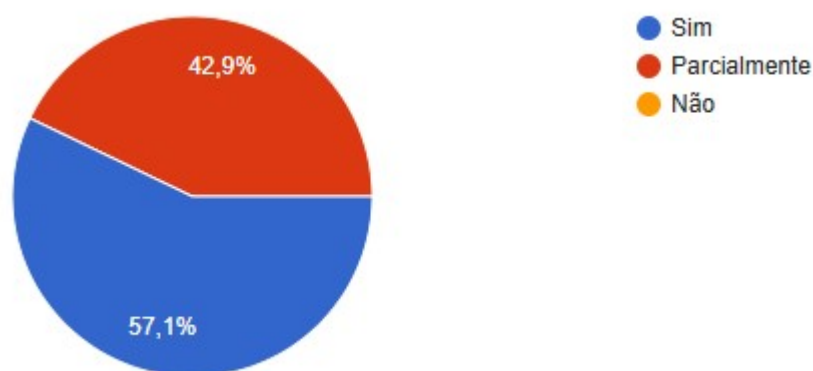
Fonte: Próprio Autor (2025)

Observa-se que 50% dos alunos avaliaram o jogo Python Den como “Excelente”, 28,6% como “Bom” e 21,4% como “Razoável”.

Observa-se a avaliação dos três jogos, que o Scratch foi o mais bem avaliado entre os participantes, indicando facilidade de uso e compreensão. O Cargo Bot, por sua vez, recebeu avaliação predominantemente “Razoável”, possivelmente em razão do maior nível de complexidade e das exigências de raciocínio lógico que o jogo demanda. Já o Python Den obteve resultados positivos, evidenciando uma boa aceitação dos alunos mesmo diante do desafio de programar em linguagem textual.

A quarta pergunta realizada para os alunos foi: “O uso do jogo digital facilitou a compreensão dos conceitos de programação?”, cujas respostas podem ser observadas no Gráfico 7.

Gráfico 7 - O uso do jogo digital facilitou a compreensão dos conceitos de programação?



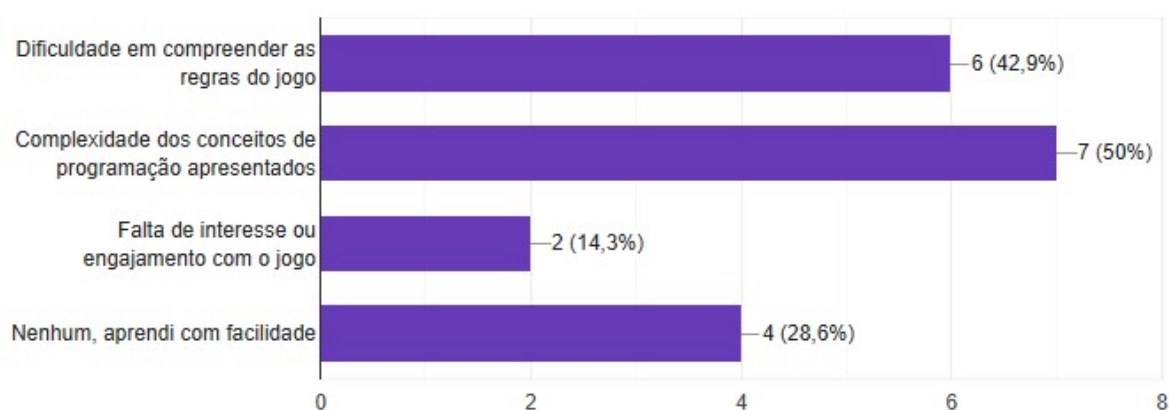
Fonte: Próprio Autor (2025)

Observa-se que a maioria dos alunos (57,1%) reconheceu que o uso do jogo digital facilitou a compreensão dos conceitos de programação, enquanto 42,9% indicaram que essa contribuição ocorreu de forma parcial. Nenhum aluno respondeu negativamente, o que demonstra que a utilização de jogos digitais foi percebida, em maior ou menor grau, como um recurso eficaz para apoiar o aprendizado de programação. Esses resultados reforçam a eficácia da metodologia adotada, evidenciando que o uso de jogos digitais pode favorecer a compreensão conceitual e o desenvolvimento do pensamento computacional de forma lúdica e significativa.

A quinta pergunta realizada para os alunos foi: “Quais desafios você encontrou ao aprender Lógica e Linguagem de Programação por meio dos jogos digitais?”. As opções de resposta disponíveis eram: “Dificuldade em compreender as regras do jogo”, “Complexidade dos conceitos de programação apresentados”, “Falta de interesse ou engajamento com o jogo” e “Nenhum, aprendi com

facilidade”, os alunos podiam marcar mais de uma opção, se fosse o caso, o que ocorreu. O Gráfico 8 apresenta as principais dificuldades relatadas pelos alunos durante a utilização dos jogos digitais como ferramenta ao ensino de programação.

Gráfico 8 - Quais desafios você encontrou ao aprender Lógica e Linguagem de Programação por meio dos jogos digitais?



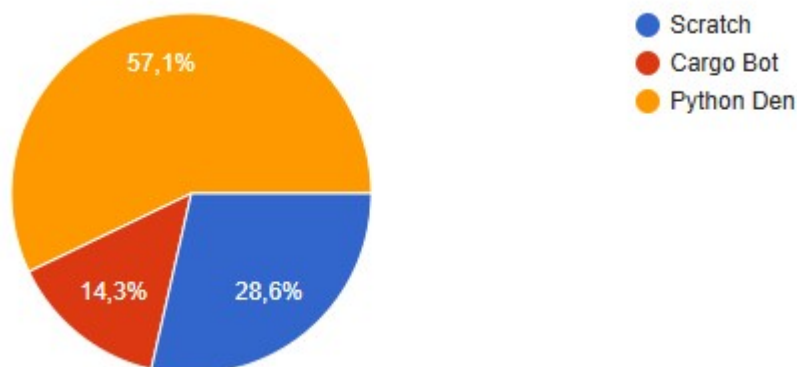
Fonte: Próprio Autor (2025)

Observa-se que a “Complexidade dos conceitos de programação apresentados” foi o aspecto mais citado, com 50% dos alunos (7 respostas), indicando que parte significativa dos participantes encontrou dificuldades na compreensão dos conteúdos abordados. Em seguida, 42,9% dos alunos (6 respostas) relataram “Dificuldade em compreender as regras do jogo”, o que sugere que, além dos conceitos de programação, a própria dinâmica do jogo demandou certo tempo de adaptação. Por outro lado, 14,3% dos alunos (2 respostas), mencionaram “Falta de interesse ou engajamento com o jogo”, o que demonstra que, de modo geral, o recurso foi bem aceito pelos participantes. Além disso, 28,6% dos alunos (4 respostas) afirmaram não ter enfrentado dificuldades, “Nenhum, aprendi com facilidade”, o que reforça o potencial do jogo como ferramenta pedagógica eficaz ao ensino de lógica e programação.

Os resultados indicam que, embora existam dificuldades em compreender as regras do jogo, o uso do jogo digital contribuiu para a aprendizagem ativa e o desenvolvimento do pensamento computacional. Segundo Wing (2006), o pensamento computacional envolve a capacidade de formular problemas e expressar suas soluções de modo que possam ser executadas por um agente computacional, competência que pode ser estimulada por meio de atividades lúdicas e baseadas em desafios.

A sexta pergunta realizada para os alunos foi: “Qual jogo você considera mais eficaz como ferramenta para o ensino e a aprendizagem da disciplina de Lógica e Linguagem de Programação?”, cujas respostas podem ser observadas no Gráfico 9

Gráfico 9 – Qual jogo você considera mais eficaz como ferramenta para o ensino e a aprendizagem da disciplina de Lógica e Linguagem de Programação?



Fonte: Próprio Autor (2025)

Observa-se que o Python Den foi apontado como o jogo mais eficaz por 57,1% dos alunos, seguido pelo Scratch, com 28,6%, e pelo Cargo Bot, indicado por 14,3%.

Esse resultado evidencia uma preferência pelo Python Den, o que pode estar relacionado ao fato de que esse jogo utiliza uma linguagem de programação (Python). O jogo possibilita ao aluno compreender de forma mais concreta a aplicação dos conceitos de lógica e estrutura de programação, favorecendo a transição do pensamento computacional para a escrita de código, conforme apontado por Wing (2006).

O Scratch, embora tenha sido o segundo mais citado, mantém uma papel relevante como ferramenta introdutória, pois oferece um ambiente visual e intuitivo que facilita o aprendizado inicial da lógica de programação.

O Cargo Bot apresentou menor índice de preferência, possivelmente devido a seu nível de dificuldade em função do raciocínio lógico exigido.

A sétima pergunta realizada para os alunos foi: “Qual conteúdo da disciplina de Lógica e Linguagem de Programação você encontrou maior dificuldade para aprender? Essa dificuldade foi superada com o uso dos jogos digitais? A maioria dos alunos encontrou dificuldades em geral na linguagem de programação Python e no jogo Cargo Bot, conforme observa-se nas respostas; apenas um aluno não respondeu.

O Cargo Bot eu apresentei uma dificuldade maior pois pedia um raciocínio lógico maior e eu não conseguia desenvolver com uso do jogo (Aluno 1).

Python, foi um pouco complicado e ainda é mais estou conseguindo (Aluno 2).

Eu tenho problema de raciocínio lógico e encontrei dificuldades na hora de executar as variáveis e a lista mais com a ajuda do Python Den eu consegui entender melhor a estrutura dos comandos o que facilitou bem o meu raciocínio e meu entendimento sobre tal (Aluno 3).

Python e vscode (Aluno 4).

Cargo Bot, pois é um jogo que exige uma certa complexidade que faz com que exige um pouco mais do que o normal do aluno/usuário do ao jogar ele (Aluno 5).

Python, sim me ajudou muito o uso dos jogos! (Aluno 6).

While foi o que eu tive mais dificuldade e graças ao Python Den fez com que eu entendesse melhor este comando (Aluno 7).

Cargo Bot, achei muito difícil de entender os comandos, então fiz o básico mais longo que deu certo (Aluno 8).

Code for Life (Aluno 9).

Eu tenho muita dificuldade com o código de programação Python. Mesmos com os jogos ainda não consegui compreender muito isso (Aluno 10).

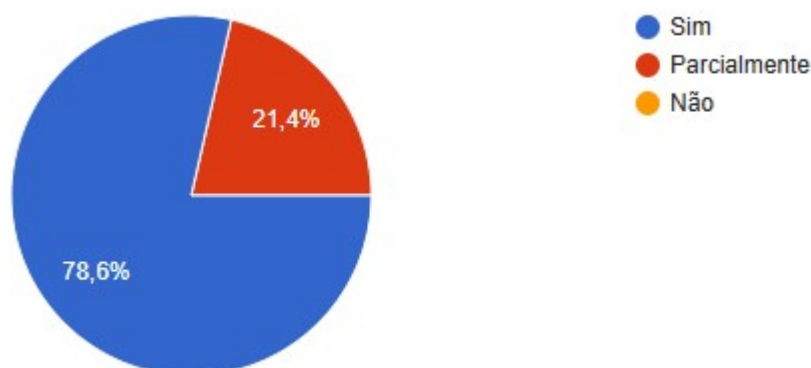
Python, sim (Aluno 11)

Nenhum (Aluno 12)

Python Den e Scratch, para mim há grandes dificuldades, pois não consigo acompanhar e entender tão bem, já o Cargo bot é razoável e tem técnicas além de ensina como praticar (Aluno 13).

A oitava pergunta realizada para os alunos foi: “Os jogos digitais tornaram as aulas mais dinâmicas e motivadoras? cujas respostas podem ser observadas no Gráfico 10.

Gráfico 10 - Os jogos digitais tornaram as aulas mais dinâmicas e motivadoras?



Fonte: Próprio Autor (2025)

Observa-se que a grande maioria dos alunos, 78,6% respondeu “Sim”, enquanto 21,4% afirmaram que “Parcialmente” e nenhum aluno respondeu “Não”.

Esses dados evidenciam que o uso de jogos digitais no ensino de lógica e programação foi amplamente percebido pelos alunos como uma estratégia motivadora e dinamizadora das aulas. A elevada porcentagem de respostas positivas demonstra que a inserção de recursos lúdicos contribui para tornar o processo de ensino-aprendizagem mais envolvente, favorecendo o interesse e a participação dos alunos.

De acordo com Valente (1999), as tecnologias digitais podem desempenhar um papel transformador no contexto educacional quando utilizadas como ferramenta pedagógica, promovendo aprendizagem significativa. O resultado do gráfico reforça essa concepção, ao indicar que os alunos percebem as aulas mediadas por jogos como experiências mais dinâmicas, interativas e desafiadoras.

Portanto, conclui-se que a utilização de jogos digitais contribuiu para aumentar a motivação e o engajamento dos alunos, para o desenvolvimento do pensamento computacional e para o fortalecimento do vínculo entre teoria e prática no ensino de programação.

5.0 Metodologia de Aprendizagem com Jogos Digitais

Com base nos resultados desta pesquisa, foi elaborada uma metodologia de aprendizagem baseada em jogos digitais, destinada a apoiar professores no ensino de programação. Essa metodologia utiliza jogos digitais como ferramenta pedagógica para auxiliar o processo de ensino e aprendizagem dos alunos.

5.1 Fundamentação

A proposta metodológica fundamenta-se na Metodologia Ativa, Aprendizagem Baseada em Jogos Digitais (*Game-Based Learning*), tendo como objetivo integrar o ensino de lógica e linguagem de programação de forma lúdica, interativa e contextualizada, favorecendo o desenvolvimento dos pilares: decomposição, reconhecimento de padrões, abstração e algoritmos, entre os alunos do ensino médio e técnico, por meio de atividades práticas mediadas por jogos. O método integra teoria e prática da programação em linguagem Python, promovendo o engajamento, a autonomia e a aprendizagem significativa dos alunos.

5.2 Estrutura Metodológica

A Tabela 9 apresenta a estrutura da metodologia de aprendizagem baseada em jogos digitais proposta nesta pesquisa, organizada em etapas progressivas que articulam os conteúdos curriculares da disciplina de Lógica e Linguagem de Programação ao desenvolvimento das habilidades do pensamento computacional. A metodologia foi concebida a partir da aplicação prática descrita nos Capítulos 3 e 4 desta pesquisa; considerando o contexto educacional, o perfil dos alunos, a linguagem de programação adotada (Python) e os jogos digitais selecionados como recursos pedagógicos.

Cada etapa da metodologia estabelece uma relação direta entre os conteúdos programáticos, os objetivos de aprendizagem e as estratégias didáticas adotadas, promovendo a transição gradual da programação em blocos para a programação textual. O uso dos jogos digitais Scratch, Cargo Bot e Python Den possibilita ao aluno construir o conhecimento de forma ativa, visual e progressiva, favorecendo a

compreensão de conceitos abstratos da programação e estimulando o desenvolvimento dos quatro pilares do pensamento computacional: decomposição, reconhecimento de padrões, abstração e algoritmos.

Tabela 9 – Estrutura Metodológica da Aprendizagem com Jogos Digitais

| Etapa | Conteúdo Curricular | Jogo Digital Utilizado | Objetivos da Etapa | Estratégias, Recursos e Procedimentos Didáticos | Pilares do Pensamento Computacional e Aplicação no Jogo |
|---|---|---------------------------------|---|--|---|
| 1. Introdução à Lógica e Variáveis | Estrutura básica da programação: algoritmo, variáveis, entrada e saída de dados | Scratch | Desenvolver o raciocínio lógico inicial e introduzir o conceito de variáveis de forma visual e intuitiva. | Apresentação do ambiente Scratch; criação e programação de sprites; uso de variáveis visuais; resolução de problemas simples; transposição dos algoritmos em blocos para Python no Visual Studio Code. | Abstração: representação visual de dados por meio de variáveis do Scratch. Decomposição: divisão do problema em pequenas ações programáveis. Algoritmos: organização sequencial dos blocos para executar uma tarefa. |
| 3. Estrutura de Repetição (Loops) | Laços contados e condicionais | Scratch, Cargo Bot e Python Den | Desenvolver o raciocínio lógico sequencial e compreender a repetição de comandos. | Resolução de desafios no Cargo Bot; construção de laços com blocos “repita” no Scratch; exploração inicial de laços no Python Den; implementação de loops em Python. | Reconhecimento de padrões: identificação de ações repetitivas nos desafios. Algoritmos: criação de laços para otimizar a solução. Decomposição: separação das etapas repetidas do problema. |

| Etapa | Conteúdo Curricular | Jogo Digital Utilizado | Objetivos da Etapa | Estratégias, Recursos e Procedimentos Didáticos | Pilares do Pensamento Computacional e Aplicação no Jogo |
|---|--|------------------------|---|--|--|
| 4. Estruturas de Repetição e Decisão Integradas | Integração entre laços e estruturas condicionais | Python Den | Consolidar a aplicação conjunta de estruturas de decisão e repetição em Python. | Resolução de níveis progressivos no Python Den; escrita de código em Python; transcrição para o Visual Studio Code; discussão de soluções otimizadas. | Algoritmos: elaboração de soluções mais complexas com decisões e repetições integradas. Abstração: análise das condições necessárias para controle do fluxo. Reconhecimento de padrões: identificação de comportamentos recorrentes nos desafios. |
| 5. Listas e Manipulação de Dados | Estruturas de dados: listas | Python Den | Aplicar o pensamento computacional na organização e manipulação de dados. | Exploração de níveis do Python Den focados em listas; desenvolvimento de programas em Python para armazenamento e processamento de dados; atividades práticas avaliativas. | Abstração: representação de conjuntos de dados por meio de listas. Decomposição: divisão do problema em operações sobre os elementos da lista. Algoritmos: criação de procedimentos para percorrer e manipular dados. |

Fonte: Próprio Autor (2025)

5.3 Etapas da Aplicação

A aplicação da metodologia ocorreu em sete fases:

- ✓ seleção dos jogos de acordo com o conteúdo curricular;
- ✓ explicação introdutória do conteúdo;
- ✓ aplicação prática do conteúdo por meio do jogo;
- ✓ orientação do professor enquanto mediador;
- ✓ transposição para o código Python;
- ✓ discussão sobre a equivalência entre blocos visuais e comandos textuais;
- ✓ aplicação de atividades práticas.

5.4 Recursos Didáticos

- ✓ Laboratório de Informática com acesso à internet e notebooks individuais;
- ✓ Softwares empregados foram Scratch, Cargo Bot, Code for Life (Python Den) e Visual Studio Code.

5.5 Resultados Esperados

A aplicação da Metodologia de Aprendizagem com Jogos Digitais possibilita:

- ✓ desenvolvimento progressivo das habilidades de pensamento computacional;
- ✓ aprendizagem significativa dos conceitos de lógica e programação;
- ✓ maior engajamento e motivação dos alunos;
- ✓ melhora no desempenho em atividades e avaliações práticas; e
- ✓ formação de alunos autônomos e solucionadores de problemas.

6. Considerações Finais

A metodologia aplicada, fundamentada na Metodologia Ativa de Aprendizagem Baseada em Jogos Digitais (*Game-Based Learning*), proporcionou um ambiente de aprendizagem dinâmico, no qual os alunos puderam construir o conhecimento de forma prática, lúdica e interativa.

Os resultados obtidos evidenciaram que o uso de jogos digitais favorece o engajamento, estimula a participação e desperta o interesse dos alunos pela programação.

Os dados coletados por meio de atividades práticas e avaliações demonstraram que a maioria dos alunos apresentou melhora no desempenho e na autoconfiança ao lidar com desafios de lógica e programação.

Com base nos dados obtidos no questionário, o Python Den foi identificado como o jogo mais eficaz no processo de ensino-aprendizagem. Embora o Scratch tenha apresentado a melhor avaliação geral em termos de facilidade e usabilidade, o Python Den destacou-se por sua efetividade pedagógica: 57,1% dos alunos o escolheram como o recurso mais eficiente para aprender programação, além de ter recebido 78,6% de avaliações positivas entre “Excelente” e “Bom”. As respostas abertas reforçam esse resultado, evidenciando que o jogo contribuiu para superar dificuldades.

O jogo mais desafiador para os alunos foi o Cargo Bot, cuja complexidade contribuiu para dificuldades, mas também estimulou o raciocínio lógico dos alunos que conseguiram avançar nos desafios.

Conclui-se que a integração entre os conteúdos curriculares e os jogos Scratch, Cargo Bot e Python Den possibilitou o desenvolvimento gradual dos quatro pilares do pensamento computacional: decomposição, reconhecimento de padrões, abstração e algoritmo; enquanto a utilização sequencial desses recursos, acompanhando a progressão da disciplina, favoreceu a transição do raciocínio visual para a codificação em Python. Organizados em uma sequência didática estruturada, com desafios graduais e mediação docente adequada, os três jogos configuram uma estratégia eficaz para promover a aprendizagem ativa, fortalecer a relação entre

teoria e prática e consolidar a compreensão dos conceitos fundamentais da programação.

Esses resultados corroboram autores como Wing (2006), ao evidenciar que o pensamento computacional pode ser estimulado por meio de atividades desafiadoras e lúdicas, e Valente (1999), que destaca o potencial transformador das tecnologias digitais no processo de ensino-aprendizagem.

A pesquisa confirma que o uso de jogos digitais no ensino de programação não deve ser visto apenas como um recurso complementar, mas como uma metodologia pedagógica, capaz de integrar o conhecimento técnico, criatividade e resolução de problemas, contribuindo para o desenvolvimento de competências essenciais ao século XXI, como o raciocínio lógico, o pensamento crítico e a autonomia intelectual.

Por fim, a Metodologia de Aprendizagem com Jogos Digitais, apresentada neste trabalho, constitui-se como um modelo replicável e adaptável a diferentes contextos educacionais, oferecendo aos professores uma alternativa para o ensino de lógica e programação e contribuindo para a formação de alunos mais engajados, criativos e preparados para os desafios tecnológicos contemporâneos.

REFERÊNCIAS

ASSOCIAÇÃO Brasileira das Empresas de Tecnologia da Informação e Comunicação. *Até 2025 devem surgir quase 800 mil vagas de emprego para área de tecnologia*. Disponível em: <https://jconcursos.uol.com.br/noticia/empregos/ate-2025-devem-surgir-quase-800-mil-vagas-de-emprego-para-area-de-tecnologia-108898>.

Acesso em: 26 out. 2024.

BARBOSA, P. M. R. O construtivismo e Jean Piaget. *Revista Educação Pública*, Rio de Janeiro, v. 15, n. 12, 2015. Disponível em: <https://educacaopublica.cecierj.edu.br/artigos/15/12/o-construtivismo-e-jean-piaget>.

Acesso em: 06 mai 2025.

BATISTA, E. J. S. *Pensamento computacional: teoria e prática*. Campo Grande, MS: Ed. UFMS Digital, 2024.

BENDER, William N. *Aprendizagem baseada em projetos: educação diferenciada para o século XXI*. Porto Alegre: Penso, 2014.

BERGMANN, Jonathan; SAMS, Aaron. *Flip your classroom: reach every student in every class every day*. Washington, DC: International Society for Technology in Education, 2012.

BORGES, B. de O. *A linguagem Python como ferramenta no ensino básico*. 2021. 69 f. Dissertação (Mestrado Profissional em Matemática em Rede Nacional) – Universidade Federal do Triângulo Mineiro, Uberaba, 2021.

BRACKMANN, C. P. *Desenvolvimento do pensamento computacional através de atividades desplugadas na educação básica*. 2017. Tese (Doutorado em Informática na Educação) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2017.

BRASIL. Base Nacional Comum Curricular. Brasília: Ministério da Educação, 2018. Disponível em: <https://basenacionalcomum.mec.gov.br/>. Acesso em: 10 abr 2025.

BRASSCOM. *Relatório de Perspectivas do Mercado de Trabalho do Macrossetor TIC*. Disponível em notícias e relatórios do setor (2025).

BROWN, T. *Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation*. HarperBusiness, 2009.

CALDERON, I.; SILVA, W.; FEITOSA, E. Um mapeamento sistemático da literatura sobre o uso de metodologias ativas durante o ensino de programação no Brasil. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 32., 2021, Online. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2021. p. 1152–1161. DOI: <https://doi.org/10.5753/sbie.2021.217564>.

CARVALHO, C. V. de. Aprendizagem baseada em jogos – Game-based learning. In: WORLD CONGRESS ON SYSTEMS ENGINEERING AND INFORMATION TECHNOLOGY, 2., 2015. Anais [...]. [S.l.: s.n.], 2015. Disponível em: <https://pt.scribd.com/document/473220086/40>. Acesso em: 15 dez. 2024.

CASTRO JUNIOR, A. A. de; SANDIM, H. da C.; RIEDNER, D. D. T. *Pensamento computacional: teoria e prática*. Campo Grande: UFMS, 2023. Disponível em: <https://repositorio.ufms.br/bitstream/123456789/8876>. Acesso em: 18 maio 2025.

CODE FOR LIFE. 2025. Disponível em: <https://www.codeforlife.education/about>. Acesso em: 30 ago. 2025.

COHEN, L.; MANION, L.; MORRISON, K. *Research Methods in Education*. 5th ed. London and New York: Taylor & Francis e-Library, 2005.

CORREIA, G. G.; LOBANCO, A. J. M.; NASCIMENTO, T. C. F.; SOUZA, R. M. de. *William Glasser e a Pirâmide da Aprendizagem*. Anais do Fórum de Iniciação Científica do UNIFUNEC, v. 14, n. 14, 2024. Disponível em: <https://seer.unifunec.edu.br/index.php/forum/article/view/6279>. Acesso em: 10 abr 2025.

FALCÃO, T. P.; BARBOSA, R. "Aperta o play!": análise da interação exploratória em um jogo baseado em pensamento computacional. *Simpósio Brasileiro de Informática na Educação*, 2015. Disponível em: https://www.researchgate.net/publication/300236684_Aperta_o_Play_Analise_da_Interaçao_Exploratoria_em_um_Jogo_Baseado_em_Pensamento_Computacional. Acesso em: 02 set. 2025.

FREIRE, P. *Pedagogia da autonomia: saberes necessários à prática educativa*. 38. ed. São Paulo: Paz e Terra, 1996.

FUNDAÇÃO DOM CABRAL. *Future Jobs 2025: tendências do futuro do trabalho no Brasil*. Belo Horizonte: FDC, 2025. Disponível em: https://www.fdc.org.br/Documents/Future_Jobs_2025_Relat%C3%B3rio.pdf. Acesso em: 18 maio 2025.

GARLET, D.; BIGOLIN, N. M.; SILVEIRA, S. R. Ensino de programação de computadores na educação básica: um estudo de caso. *Revista Eletrônica de Sistemas de Informação e Gestão Tecnológica*, v. 9, n. 2, 2018.

GIGERENZER, G. *O poder da intuição: o que a ciência da decisão tem a dizer sobre como tomamos decisões*. Rio de Janeiro: Elsevier, 2008.

GIL, A. C. *Métodos e técnicas de pesquisa social*. 6. ed. São Paulo: Atlas, 2011.

GIL, A. C. *Como elaborar projetos de pesquisa*. 6. ed. São Paulo: Atlas, 2018.

GLASSER, W. *Teoria da escolha: uma nova psicologia da liberdade pessoal*. São Paulo: Summus, 2001.

GUARDA, G. F.; PINTO, S. C. C. O uso dos jogos digitais educacionais no processo no ensino-aprendizagem com ênfase nas habilidades do pensamento computacional: experiências no ensino fundamental. *Revista Brasileira de Pós-Graduação*, v. 17, n. 37, 2021.

HILES, D. *Heuristic Inquiry and Transpersonal Research*. London: CCPE, 2001.

MARTINS, E. P.; PRADO, M. E. B. B.; COSTA, M.; SILVA, A. F. G.; CAMBA, M. Pensamento computacional e jogos digitais: possibilidades para o ensino de matemática na educação básica. *Revista Científica a Educação a Distância Paidéia*, v. 15, n. 27, 2023. DOI: 10.5281/zenodo.10436162.

MARTINS, M. W. P.; MARIN, R. P. M.; ALVES, L. B. Desafios na aprendizagem de lógica de programação entre alunos de tecnologia: uma revisão sistemática da literatura. *Revista EDaPECI*, v. 24, n. 3, p. 12–24, dez. 2024. Disponível em: <https://www.researchgate.net/publication/386350791>. Acesso em: 06 maio 2025.

MORAN, J.; BACICH, L. *Metodologias ativas para uma educação inovadora*. Porto Alegre: Penso, 2018.

PAVLUS, J. Cargo-Bot, an addictive iPad game that teaches programming concepts. *Fast Company*, 25 maio 2012. Disponível em: <https://www.fastcompany.com/1669821/cargo-bot-an-addictive-ipad-game-that-teaches-programming-concepts>. Acesso em: 14 maio 2025.

PEREIRA, K. C.; SOUSA, R. R. Dificuldades de aprendizagem e fatores que motivam para o estudo para programação: o que mudou desde os estudos de 2018 no curso de TI da UFERSA, Campus Pau dos Ferros? *Recima21*, v. 5, n. 4, 2024. Disponível em: <https://www.researchgate.net/publication/379905259>. Acesso em: 06 maio 2025.

PIAGET, Jean. *A psicologia da criança*. Rio de Janeiro: Bertrand Brasil, 1973.

PINHO, G. et al. Proposta de jogo digital para dispositivos móveis: desenvolvendo habilidades do pensamento computacional. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, XXVII, 2016. Disponível em: <https://www.researchgate.net/publication/309887617>. Acesso em: 03 dez. 2024.

PRENSKY, M. *Aprendizagem baseada em jogos digitais*. Tradução de Karina Jannini. São Paulo: Senac São Paulo, 2012.

PLATTNER, H.; MEINEL, C.; LEIFER, L. *Design Thinking: Understand – Improve – Apply*. Berlin: Springer, 2010.

PRIETO, L. M.; TREVISAN, M. C. B.; DANEZI, M. I.; FALKEMBACH, G. M. Uso das tecnologias digitais em atividades didáticas nas séries iniciais. *Renote: Revista Novas Tecnologias na Educação*, v. 3, n. 1, 2005.

SAUR-AMARAL, I. *Revisão sistemática da literatura*. Lisboa: BUBOK, 2010.

SÃO PAULO (Estado). *Plano de curso – Técnico em Desenvolvimento de Sistemas*. São Paulo: Secretaria da Educação, 2024. Acesso em: 14 maio 2025.

SANTOS, A. dos. *Seleção do método de pesquisa: guia para pós-graduandos em design e áreas afins*. Curitiba: Insight, 2018.

SAVERY, John R. Overview of problem-based learning: definitions and distinctions. *Interdisciplinary Journal of Problem-Based Learning*, West Lafayette, v. 1, n. 1, p. 9–20, 2006.

SAVI, R.; ULBRICHT, V. R. Jogos digitais educacionais: benefícios e desafios. *Renote – Revista Novas Tecnologias na Educação*, v. 6, n. 1, 2008.

SCRATCH. Crie histórias, jogos e animações. 2025. Disponível em: <https://scratch.mit.edu/about>. Acesso em: 24 mar. 2025.

SEBESTA, Robert W. *Conceitos de linguagens de programação*. Porto Alegre: Bookman, 2010.

SGOBBI, S. Área de tecnologia gera 150 mil empregos por ano no Brasil, diz Brasscom. [S.l.]: Brasscom, 2023. Disponível em: <https://www.brasscom.org.br/noticia>. Acesso em: 06 maio 2025.

SIQUEIRA, I. C. P. et al. Normas sobre computação na educação básica – complemento à base nacional comum curricular (BNCC). *Relatório Técnico*. Conselho Nacional de Educação – Câmara de Educação Básica, 2022. Acesso em: 10 dez. 2024.

SILVA, I. R. S.; SILVA, R. O. da. Linguagem de programação Python. *Revista Tecnologias em Projeção*, Brasília, v. 10, n. 1, p. 55–61, 2019. Disponível em: <https://projecaociencia.com.br/index.php/Projecao4/article/view/1359>. Acesso em: 14 maio 2025.

SOUZA, A. L. A.; VILAÇA, A. L. A.; TEIXEIRA, H. J. B. Os benefícios da metodologia ativa de aprendizagem na educação. In: COSTA, G. M. C. (org.). *Metodologias ativas: métodos e práticas para o século XXI*. Quirinópolis, GO: Editora IGM, 2020.

SOUZA, C. *Desbravando a tecnologia da informação*. 2. ed. São Paulo: TechEditora, 2022.

THIOLLENT, M. *Metodologia da pesquisa-ação*. 18. ed. São Paulo: Cortez, 2011.

VALENTE, J. A.; FREIRE, F. M. P.; ARANTES, F. L.; D'ABREU, J. V. V.; AMIEL, T.; BARANAUSKAS, M. C. C.. Alan Turing tinha pensamento computacional? Reflexões sobre um campo em construção. *Tecnologias, Sociedade e Conhecimento*, v. 4, n. 1, p. 7–22, 2017.

VALENTE, J. A. O computador na sociedade do conhecimento. Campinas, SP: UNICAMP/NIED, 1999.

WING, J. M. Computational thinking. *Communications of the ACM*, v. 49, n. 3, p. 33–35, 2006. DOI: <https://doi.org/10.1145/1118178.1118215>

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.

APÊNDICE A – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

Estamos solicitando a você a autorização para que o menor pelo qual você é responsável participe da pesquisa: O USO DE JOGOS DIGITAIS COMO FERRAMENTA NO DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL APLICADO À DISCIPLINA DE LÓGICA E LINGUAGEM DE PROGRAMAÇÃO, que tem como pesquisador responsável Andresa Cristiane dos Santos e orientador Prof. Dr. José Adriano Sorbille, vinculada a instituição: UNIFATEA.

Esta pesquisa pretende-se investigar o uso de jogos digitais como ferramenta pedagógica no desenvolvimento das habilidades do pensamento computacional, para auxiliar a aprendizagem de estudantes do ensino médio e técnico nas disciplinas de programação de computadores.

A participação do menor é totalmente voluntária e a coleta de dados se dará por meio da aplicação de um questionário online estruturado, com perguntas objetivas e subjetivas, elaborado através da ferramenta GoogleForms. Será realizada durante o horário de aula no laboratório de Informática da escola Murillo do Amaral; sobre a contribuição do uso de jogos digitais educacionais como ferramenta pedagógica durante as aulas da disciplina de Lógica e Linguagem de Programação. Participarão da pesquisa os estudantes regularmente matriculados no 2º ano do ensino médio e técnico em Desenvolvimento de Sistemas, da escola da rede Escola Estadual Murillo do Amaral localizada na cidade de Aparecida-SP. Os alunos receberão o convite para acesso ao questionário do GoogleForms por e-mail e utilizarão os computadores da escola para responder durante o horário de aula.

As perguntas foram formuladas de maneira clara e objetiva, e a pesquisa foi projetada para ser breve, com um tempo estimado para conclusão de 15 a 20 minutos.

Há benefícios e riscos decorrentes da participação do menor na pesquisa. Como benefícios, o uso de jogos digitais como ferramenta pedagógica contribui para o desenvolvimento do pensamento computacional, tornando o aprendizado mais atrativo e eficaz. Essa abordagem lúdica favorece o ensino de programação no

ensino técnico, estimula o interesse dos alunos e diversifica a metodologia docente, promovendo habilidades cognitivas e técnicas importantes para o mercado de trabalho.

Os riscos são mínimos, inerentes ao ambiente virtual e ao uso de meios eletrônicos para a participação na pesquisa incluem a possibilidade de exposição não intencional dos dados fornecidos, mesmo com a adoção de medidas de segurança e anonimato. Além disso, há o risco de interceptação de informações por terceiros, ataques cibernéticos, falhas técnicas que comprometam a transmissão ou o armazenamento dos dados, bem como acessos não autorizados aos dispositivos utilizados para responder ao questionário. Riscos de privacidade e confidencialidade: os dados coletados serão anônimos, mas pode haver um risco relacionado ao vazamento ou acesso indevido às informações.

Mitigação: Todos os dados serão armazenados em plataformas seguras, e a confidencialidade será garantida conforme o Termo de Consentimento Livre e Esclarecido (TCLE). A pesquisa será conduzida em conformidade com as regulamentações de proteção de dados pessoais (como a LGPD). Os dados do formulário são encriptados em repouso e em trânsito, será adotada proteção interna contra phishing, fornecida pelo próprio fabricante do software. A ferramenta de coleta de dados utilizada tem controles de segurança para prevenir, detectar e responder a vulnerabilidades e violações de dados, além destas medidas recomenda-se que o participante utilize conexões seguras, dispositivos atualizados e evite redes públicas ou compartilhadas ao preencher o questionário. Destaca-se a importância do participante guardar uma cópia do documento eletrônico em seus arquivos. É possível obter uma cópia deste documento através do link

https://1drv.ms/w/s!AtTGAzpQ_icJvw1fz72rGEerE0Dd?e=dosaaq

Riscos psicológicos: embora a pesquisa envolva apenas o preenchimento de um questionário, existe a possibilidade de que alguns participantes possam sentir desconforto ao refletir e avaliar os jogos utilizados durante as aulas.

Mitigação: o questionário será aplicado em um ambiente confortável, com explicação clara dos objetivos e da participação voluntária, permitindo desistência sem prejuízo. Além disso, seu formato será objetivo e de fácil compreensão, reduzindo a fadiga

dos participantes.

Riscos de desconforto físico leve: como a pesquisa será realizada remotamente, o único risco potencial seria o desconforto físico relacionado ao uso prolongado de dispositivos como computadores ou celulares, o que pode causar cansaço visual ou postural.

Mitigação: o questionário pode ser completado no tempo que o participante achar mais conveniente, com intervalos, evitando que se sintam obrigados a fazer longos períodos de resposta seguidos.

Riscos de sobrecarga cognitiva: dependendo do tempo que o participante leva para responder, pode haver uma sobrecarga cognitiva, especialmente se as perguntas forem complexas ou exigirem reflexão profunda sobre práticas pedagógicas.

Mitigação: as perguntas foram formuladas de maneira clara e objetiva, e a pesquisa foi projetada para ser breve, com um tempo estimado para conclusão de 15 a 20 minutos.

Entretanto, com vistas a prevenir os possíveis danos gerados pela presente pesquisa os questionários poderão ser realizados conforme disponibilidade de tempo do participante, será garantido o anonimato da identidade do colaborador. O colaborador poderá abandonar a pesquisa a qualquer momento, deixar de responder qualquer questão que não desejar e solicitar a remoção de seus dados da pesquisa. Caso o participante queira que suas informações não sejam publicadas ou sejam excluídas, incluindo termos, anuências e consentimentos, basta formalizar o pedido pelo e-mail andresa.csf@gmail.com. Após a solicitação, todos os dados serão removidos da plataforma utilizada para a coleta.

Caso haja algum dano ao participante será garantido aos mesmos procedimentos que visem à reparação e o direito à indenização.

Para participar deste estudo o Sr.(a) não terá nenhum custo e não receberá nenhuma vantagem financeira.

O Sr.(a) receberá o esclarecimento sobre o estudo em qualquer aspecto que desejar e estará livre para recusar-se a participação do menor e a recusa em participar não acarretará nenhuma penalidade ou modificação na forma em que é atendido pelo pesquisador, que tratará a sua identidade com padrões profissionais de sigilo.

Os resultados da pesquisa estarão à sua disposição quando finalizada. O nome do menor estudante ou o material que indique sua participação não será liberado sem a sua permissão. O estudante não será identificado em nenhuma fase da pesquisa e nem em publicação que possa resultar. Todas as informações fornecidas pelo estudante serão tratadas de forma anônima e confidencial, sendo utilizadas apenas para fins acadêmicos.

Os dados serão armazenados de forma segura, conformidade com a resolução 466/2012 do CNS.

Os dados serão transcritos e utilizados na pesquisa, ficarão arquivados com o pesquisador responsável por um período de 5 (cinco) anos. As informações coletadas no decorrer da pesquisa, bem como os conhecimentos gerados a partir dos mesmos não serão usadas de maneira prejudicial às pessoas ou à instituição onde a pesquisa será realizada. Este termo de consentimento encontra-se impresso em duas vias, sendo que uma cópia será arquivada pelo pesquisador responsável, e a outra cópia impressa será fornecida ao senhor(a).

Para qualquer outra informação o(a) Sr.(a) poderá entrar em contato com o pesquisador pelo telefone (12) 99766-5900, ressalto que caso seja necessário estou à disposição e que pode realizar, inclusive, ligação a cobrar no número citado. Disponibilizo também meu e-mail andresa.csf@gmail.com, para demais esclarecimentos ou dúvidas.

Em caso de dúvidas com respeito aos aspectos éticos deste estudo, o(a) Sr.(a) poderá consultar o Comitê de Ética em Pesquisa no Centro Universitário Teresa D'Ávila – UNIFATEA (Av. Dr. Peixoto de Castro, 539 – Cruz, Lorena – SP, 12606-580, e-mail: cep@unifatea.edu.br)

Nome do Pesquisador: Andresa Cristiane dos Santos

Telefone e whatsapp: (12) 999766-5900

e-mail: andresa.csf@gmail.com

CONSENTIMENTO PÓS-INFORMAÇÃO

Eu, _____,

fui informado(a) dos objetivos da pesquisa “O uso de jogos digitais como ferramenta no desenvolvimento do pensamento computacional aplicado à disciplina de Lógica e Linguagem de Programação”, de maneira clara e detalhada e esclareci minhas dúvidas. Sei que a qualquer momento poderei solicitar novas informações sobre a pesquisa e me retirar da mesma sem prejuízo ou penalidade. Declaro que concordo em participar. Recebi uma cópia deste registro do consentimento livre e esclarecido e me foi dada a oportunidade de ler e esclarecer as minhas dúvidas.

APÊNDICE B – TERMO DO ASSENTIMENTO LIVRE E ESCLARECIDO

Título da pesquisa: O USO DE JOGOS DIGITAIS COMO FERRAMENTA NO DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL APLICADO À DISCIPLINA DE LÓGICA E LINGUAGEM DE PROGRAMAÇÃO

Pesquisadora: Andresa Cristiane dos Santos

Instituição: UNIFATEA.

Esta pesquisa tem como objetivo investigar o uso de jogos digitais como ferramenta pedagógica no desenvolvimento das habilidades do pensamento computacional, para auxiliar a aprendizagem de estudantes do ensino médio e técnico nas disciplinas de programação de computadores.

A sua participação é totalmente voluntária e a coleta de dados se dará por meio da aplicação de um questionário online estruturado, com perguntas objetivas e subjetivas, elaborado através da ferramenta GoogleForms, o link para acesso ao questionário será enviado por e-mail. Será realizada durante o horário de aula no laboratório de Informática da escola Murillo do Amaral; sobre a contribuição do uso de jogos digitais educacionais como ferramenta pedagógica durante as aulas da disciplina de Lógica e Linguagem de Programação. Participarão da pesquisa os estudantes regularmente matriculados no 2º ano do ensino médio e técnico em Desenvolvimento de Sistemas, da escola da rede estadual Murillo do Amaral localizada na cidade de Aparecida-SP. Os alunos receberão o convite para acesso ao questionário do GoogleForms por e-mail e utilizarão os computadores do laboratório de Informática da escola, no horário de aula para responder. As perguntas foram formuladas de maneira clara e objetiva, e a pesquisa foi projetada para ser breve, com um tempo estimado para conclusão de 15 a 20 minutos.

Há benefícios e riscos decorrentes da participação do menor na pesquisa. Como benefícios, o uso de jogos digitais como ferramenta pedagógica contribui para o desenvolvimento do pensamento computacional, tornando o aprendizado mais atrativo e eficaz. Essa abordagem lúdica favorece o ensino de programação no ensino técnico, estimula o interesse dos alunos e diversifica a metodologia docente,

promovendo habilidades cognitivas e técnicas importantes para o mercado de trabalho.

Os riscos da pesquisa são mínimos, podendo haver desconforto ao refletir e avaliar sobre a eficácia do uso de jogos digitais durante as aulas.

Todas as informações serão tratadas de forma anônima e confidencial, sendo utilizadas apenas para fins acadêmicos. Os dados serão armazenados de forma segura, em conformidade com a Resolução 466/2012 do CNS. A sua participação é voluntária. Caso decida não continuar, poderá se retirar a qualquer momento, se necessidade de justificativa e sem nenhum prejuízo. Caso tenha dúvidas sobre a pesquisa, entre em contato com a pesquisadora responsável pelo e-mail: andresa.csf@gmail.com. Se houver necessidade, o Comitê de Ética em Pesquisa da UNIFATEA pode ser contatado pelo e-mail cep@unifatea.edu.br e telefone (12)2124-28977.

Li e compreendi as informações apresentadas neste documento e concordo voluntariamente em participar desta pesquisa.

Assinatura do participante

Data: ___/___/___

APÊNDICE C – QUESTIONÁRIO APLICADO AOS ALUNOS

1-Como você avalia o jogo digital Scratch?

- a) Excelente
- b) Bom
- c) Razoável

2-Como você avalia o jogo digital Cargo-Bot?

- a) Excelente
- b) Bom
- c) Razoável

3-Como você avalia o jogo digital Python Den?

- a) Excelente
- b) Bom
- c) Razoável

4-O uso do jogo digital facilitou a compreensão dos conceitos de lógica de programação?

- a) Excelente
- b) Bom
- c) Razoável

5-Quais desafios você encontrou ao aprender lógica de programação por meio dos jogos digitais?

- a) Dificuldade em compreender as regras do jogo
- b) Complexidade dos conceitos de programação apresentados
- c) Falta de interesse ou engajamento com o jogo
- d) Nenhum, aprendi com facilidade

6-Qual jogo você considera mais eficaz como ferramenta para o ensino e a aprendizagem da disciplina de Lógica e Linguagem de Programação?

- a) Scratch
- b) Cargo-Bot
- c) Python Den

7-Qual conteúdo da disciplina de Lógica e Linguagem de Programação você encontrou maior dificuldade para aprender? Essa dificuldade foi superada com o uso dos jogos digitais?

8-Os jogos digitais tornaram as aulas mais dinâmicas e motivadoras?

- a) Sim
- b) Parcialmente
- c) Não

APÊNDICE D – AVALIAÇÃO PRÁTICA APLICADA NO 2º BIMESTRE

Crie um *sprite* no *Scratch* e um programa em *Python* que:

- Leia um número inteiro. Verifique e exiba uma mensagem se este número é negativo ou positivo.
- Uma loja de eletrônicos vende notebook no valor de R\$ 2.500,00 cada um. Se forem comprados mais de 5, o valor é de R\$ 2.300,00. Leia o número de notebooks comprados, calcule e escreva o custo total da compra.
- Ler nome e 2 notas escolares de um aluno. Calcular e exibir sua média e sua situação. Caso média maior ou igual a 6 sua situação é Aprovado caso contrário Reprovado.

APÊNDICE E – AVALIAÇÃO PRÁTICA APLICADA NO 3º BIMESTRE

- 1-Listar a sequência numérica: 10-20-30-40...100
- 2-Listar apenas os números pares no intervalo de 1 a 20.
- 3-Calcule a média de 6 números digitados pelo usuário.
- 4-Leia 5 números inteiros. Calcule o cubo de cada número.
- 5-Leia um número inteiro. Calcule o fatorial deste número.
- 6- Calcule a soma dos números de 1 a 100.

APÊNDICE F – AVALIAÇÃO PRÁTICA APLICADA NO 4º BIMESTRE

1-Liste apenas os números divisíveis por 5 de 1 a 50.

2-Lista todos os números no intervalo de 2 números digitados pelo usuário.

3-Crie uma repetição que pergunte ao usuário “Qual a sua comida preferida?” continue perguntando enquanto resposta igual a “sim” para a pergunta “Continuar respondendo ? sim ou não” .

4-Crie uma estrutura de repetição que leia um número inteiro encerre a leitura quando número digitado seja igual a 0. Ao final da leitura exiba a quantidade de números digitados e sua soma.

5-Crie um programa que leia 5 números digitados pelo usuário através de estrutura de repetição e faça:

- A soma dos números
- A média dos números

6-Crie uma lista para armazenar todos os dias da semana. Imprima todos os elementos da lista.

7-Crie um programa que leia 5 nomes digitados pelo usuário, armazene-os em uma lista. Faça:

- mostre todos os nomes cadastrados.
- Solicite um novo nome e troque pelo último nome da lista utilizando a função len
- mostre novamente todos os nomes cadastrados.

8-Crie listas paralelas:

produtos → nomes dos produtos;

preços → preços correspondentes.

Mostre todos os produtos com seus preços correspondentes lado a lado.

9- Crie uma lista com o nome de material, com os seguintes itens:*caderno, lápis, borracha e régua*.

- Insira o item “caneta” na primeira posição da lista
- Remova o terceiro item(lápis) da lista
- mostre o resultado imprimindo a lista.

10- Crie uma lista para armazenar a idade de 10 alunos. Imprima a lista e conte quantos alunos são maiores de 18 anos.

