

UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO

CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Ygor Santos Barros

**Ensino de programação no ensino médio por
meio de uma abordagem lúdica, autoral e
colaborativa**

Rio de Janeiro

2023

Ygor Santos Barros

Ensino de programação no ensino médio por meio de uma abordagem lúdica, autoral e colaborativa

Dissertação apresentada no Programa de Pós-Graduação em Informática do Centro de Ciências Exatas e Tecnologia como requisito parcial para obtenção do título de Mestre.

UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO

Orientador: Prof. Dr. Mariano Pimentel

Rio de Janeiro

2023

SB277e Santos Barros, Ygor
Ensino de programação no ensino médio por meio de uma
abordagem lúdica, autoral e colaborativa / Ygor Santos Barros. – Rio de
Janeiro, 2023.
187

Orientador: Prof. Dr. Mariano Pimentel.
Dissertação (Mestrado) - Universidade Federal do Estado do Rio de
Janeiro, Programa de Pós-Graduação em Informática, 2023.

1. ensino de programação. 2. abordagem construtivista de
programação. 3. ludicidade. 4. autoria. 5. colaboração.
I. Pimentel, Mariano, orient. II. Título.

Ygor Santos Barros

Ensino de programação no ensino médio por meio de uma abordagem lúdica, autoral e colaborativa

Dissertação apresentada no Programa de Pós-Graduação em Informática do Centro de Ciências Exatas e Tecnologia como requisito parcial para obtenção do título de Mestre.

BANCA EXAMINADORA

Documento assinado digitalmente
 **MARIANO GOMES PIMENTEL**
Data: 06/11/2023 23:42:59-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Mariano Pimentel - Orientador

Documento assinado digitalmente
 **Tadeu Moreira de Classe**
Data: 28/10/2023 09:54:40-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Tadeu Moreira de Classe

Documento assinado digitalmente
 **SIMONE DE LUCENA FERREIRA**
Data: 24/10/2023 18:20:58-0300
Verifique em <https://validar.iti.gov.br>

Prof^ª. Dra. Simone de Lucena Ferreira

UFS
Documento assinado digitalmente
 **RICARDO RODRIGUES NUNES**
Data: 31/10/2023 08:48:25-0300
Verifique em <https://validar.iti.gov.br>

Nunes
UTAD

Dedico este trabalho aos meus pais, que são a razão de eu estar aqui hoje. Expresso minha gratidão por terem me encorajado a seguir em frente em todas as etapas da minha vida. Também dedico este trabalho de pesquisa à minha esposa, Debora, cujo apoio e suporte diário foram essenciais para a concepção deste projeto. Obrigado por estar sempre ao meu lado, me motivando e encorajando a alcançar meus objetivos. Este trabalho é dedicado a você, com todo o meu amor e apreço. Por fim, gostaria de dedicar este trabalho à minha irmã Ingrid, que sempre me incentivou a perseguir meus sonhos e nunca desistir. Agradeço pelo seu apoio inabalável e por ser uma fonte de inspiração e motivação. Esta pesquisa é dedicada a você, com gratidão e carinho.

Agradecimentos

Gostaria de expressar minha gratidão ao meu orientador, o Professor Mariano Pimentel, por todo o apoio e tempo dedicado em minha orientação. Com sua orientação e incentivo, pude me desenvolver como pesquisador, e contribuir para que esta pesquisa fosse concluída. Além disso, agradeço pelas reflexões propostas continuamente, que me permitiram desconstruir e romper paradigmas, em busca de ser uma pessoa e um professor cada vez melhor. Obrigado por tudo desde já.

Também agradeço aos membros da banca, Professores Ricardo Nunes, Simone Lucena e Tadeu Classe, por aceitarem participar da avaliação do meu trabalho.

Aos estudantes do curso de Informática da Escola Técnica Estadual Maria Mercedes Mendes Teixeira que cursaram a disciplina Linguagem de Programação 1 no primeiro semestre de 2022, gostaria de agradecer por participarem e contribuírem para os estudos desta pesquisa, possibilitando sua realização. Agradeço também aos profissionais da ETEMMMT, em especial à coordenadora do curso Viviane de Abreu e a equipe da direção escolar composta por Leandro Chasse, Renata Gama, Sheila Venancia e Priscila Maisano, que apoiaram esta pesquisa com a disponibilização dos dados históricos e autorização para realização da pesquisa com os estudantes da disciplina que leciono.

Aos colegas Carolina Sacramento e Wagner Silva (STI/COC), gostaria de agradecer pelo estímulo e apoio durante toda a jornada da formação. Aos colegas Frederico Orofino e Marcello Santana, agradeço por todo conhecimento compartilhado ao longo da minha trajetória profissional.

Agradeço também à direção da Casa de Oswaldo Cruz / FIOCRUZ pelo incentivo e autorização para cursar o Mestrado, contribuindo para a minha qualificação profissional e trajetória docente.

Por fim, gostaria de expressar minha gratidão aos professores do PPGI/UNIRIO Simone Bacellar, Sean Siqueira, Gleison Santos e Rodrigo Santos, pelos conhecimentos e ensinamentos compartilhados durante as disciplinas que ministraram, que contribuíram para a minha formação profissional e de pesquisador. Levarei estes ensinamentos ao longo da vida.

Agradeço a esta Universidade e a todas as pessoas que fazem parte dela pela experiência única que vivenciei.

Resumo

O ensino de programação é geralmente parte das disciplinas dos cursos de Ciência, Tecnologia, Engenharia e Matemática (STEM). Além disso, estudos da última década mostram que essas disciplinas têm altas taxas de reprovação e evasão em todo o mundo, especialmente no cenário nacional, chegando a cerca de 30%. Além disso, a falta de motivação dos alunos é um problema comum. Como professor de uma disciplina de programação em uma escola pública do Rio de Janeiro, o pesquisador enfrenta esses desafios em seu dia a dia. Nesta pesquisa, foi desenvolvida uma nova abordagem didática com o objetivo de reduzir esses problemas e promover uma aprendizagem atraente e prazerosa para os alunos. Foram conceituados e discutidos princípios que nortearam a pesquisa e utilizou-se o método Pesquisa-Formação. A pesquisa foi conduzida no cotidiano do pesquisador e dos alunos durante o primeiro semestre de 2022, e a avaliação proporcionou dados quantitativos e qualitativos de diferentes dispositivos (diário de pesquisa, diário de classe, questionário *online* e grupo focal). Concluiu-se que a solução proposta proporcionou aos alunos um processo de aprendizagem interessante, motivador, prazeroso e engajante, aumentando seu comprometimento e contribuindo na redução da curva de aprendizado. No entanto, deve-se considerar que a evasão e a reprovação podem ser afetadas por variáveis externas e dependem do contexto em que estão inseridos. Como trabalhos futuros, pretende-se realizar mais estudos em diferentes contextos educacionais e a disseminação da solução entre outros docentes.

Palavras-chave: ensino de programação; abordagem construtivista de programação; ludicidade; autoria; colaboração.

Abstract

The teaching of programming is generally part of the subjects in Science, Technology, Engineering, and Mathematics (STEM) courses. Furthermore, studies from the last decade have shown that these subjects have high failure and dropout rates worldwide, especially in the national scenario, reaching around 30%. Additionally, student motivation is a common issue. As a professor of a programming course in a public school in Rio de Janeiro, the researcher faces these challenges on a daily basis. In this research, a new didactic approach was developed with the aim of reducing these issues and promoting engaging and enjoyable learning for students. Principles that guided the research were conceptualized and discussed, and the Research-Formation method was used. The research was conducted in the researcher's and students' daily lives during the first semester of 2022, and the assessment provided quantitative and qualitative data from different sources (research diary, class diary, online questionnaire, and focus group). It was concluded that the proposed solution provided students with an interesting, motivating, enjoyable, and engaging learning process, increasing their commitment and contributing to reducing the learning curve. However, it should be considered that dropout and failure rates may be influenced by external variables and depend on the context in which they are embedded. As future work, it is intended to conduct further studies in different educational contexts and disseminate the solution among other educators.

Keywords: Programming education; constructivist approach to programming; playfulness; authorship; collaboration.

Lista de ilustrações

Figura 1.	Monitoramento do fechamento das instituições por covid-19 no mundo	19
Figura 2.	Percentual de reprovação de estudantes por país em programação . . .	21
Figura 3.	Meme da internet sobre expectativa dos estudantes em programação . .	22
Figura 4.	Ilustração elaborada por estudante ao final da disciplina	29
Figura 5.	Desenho de uma estrela por meio da língua LOGO	31
Figura 6.	Vida e obra de Papert	32
Figura 7.	O pensador computacional	34
Figura 8.	Tradução de língua na plataforma Google Translator	40
Figura 9.	Do pensamento a realidade na Filosofia x Computação	43
Figura 10.	Ilustração de programação de blocos em Robótica	45
Figura 11.	O processo de alfabetização e letramento	47
Figura 12.	Processo iterativo do <i>Design Thinking</i>	54
Figura 13.	Abordagem da Aprendizagem Colaborativa	56
Figura 14.	Técnica de programação em pares em sala de aula	59
Figura 15.	Exemplo de ABP na prática	61
Figura 16.	Representações do processo de autoria	62
Figura 17.	Atividade lúdica realizada com os alunos	64
Figura 18.	Ilustração da interface do jogo Lightbot	66
Figura 19.	Matriz SWOT	71
Figura 20.	Exemplo de Canvas para um negócio	72
Figura 21.	Planejamento estratégico para a disciplina LP1	73
Figura 22.	Planejamento da disciplina LP1 apresentada aos alunos	75
Figura 23.	Exemplo de código gerado no Scratch	76
Figura 24.	Distribuição dos usuários Scratch por idade	77
Figura 25.	Primeiro desafio da atividade Estacionamento Algorítmico	79
Figura 26.	Avaliação da solução proposta na pesquisa	84
Figura 27.	Alunos durante a dinâmica do Estacionamento Algorítmico	99
Figura 28.	Captura de tela do jogo no formato PACMAN	101
Figura 29.	Captura de tela do jogo no formato PIMBALL	101
Figura 30.	Captura de tela do jogo no formato TIRO	102
Figura 31.	Captura de tela do jogo desenvolvido no Calango	106
Figura 32.	Dispositivos utilizados pelos estudantes durante a disciplina	108
Figura 33.	Captura da tela do diário no mês de maio de 2022	113
Figura 34.	Captura da tela do diário na planilha de consolidado	113
Figura 35.	Grupo Focal em ação	116
Figura 36.	Categorias do Domínio Cognitivo	125

Figura 37. Zona de Desenvolvimento Proximal	127
---	-----

Lista de tabelas

Tabela 1.	Relações possíveis no processo da aprendizagem	33
Tabela 2.	Alfabetização x Letramento	48
Tabela 3.	Nova organização da disciplina LP1	74
Tabela 4.	Quadro comparativo entre os dispositivos adotados na pesquisa	94
Tabela 5.	Sumário das respostas coletadas na importância de cada atividade/recurso	109
Tabela 6.	Análise estatística da situação da turma	114

Lista de abreviaturas e siglas

BNCC	Base Nacional Comum Curricular
CECIERJ	Centro de Ciências e Educação Superior à Distância do Estado do Rio de Janeiro
COC	Casa de Oswaldo Cruz
ETEMMMT	Escola Técnica Estadual Maria Mercedes Mendes Teixeira
ETER	Escola Técnica Estadual República
FAETEC	Fundação de Apoio à Escola Técnica
FIOCRUZ	Fundação Oswaldo Cruz
FP	Fundamentos de Programação
LP1	Linguagem de Programação 1
MEC	Ministério da Educação
OMS	Organização Mundial da Saúde
PC	Pensamento Computacional
PDA	Projeto e Desenvolvimento de Algoritmos
PWEB	Programação para Web
RC	Raciocínio Computacional
SBC	Sociedade Brasileira de Computação
SECTI	Secretaria de Ciência Tecnologia e Inovação do Estado do Rio de Janeiro
STEM	<i>Science, Technology, Engineering and Math</i>
UNESA	Universidade Estácio de Sá
UNESCO	Organização das Nações Unidas para a Educação, a Ciência e a Cultura
UNIRIO	Universidade Federal do Estado do Rio de Janeiro

Sumário

1	INTRODUÇÃO	14
1.1	Itinerância do pesquisador	14
1.2	A pandemia da covid-19	18
1.3	Problematização: dificuldades em programação	20
1.4	Questões de pesquisa	23
1.5	Visão da pesquisa	23
2	QUADRO TEÓRICO: FUNDAMENTAÇÃO DA PESQUISA	25
2.1	Programação e Codificação	25
2.2	Pensamento e Raciocínio Computacional	30
2.3	Linguagem e Língua de Programação	38
2.4	Ensino e Aprendizagem de Programação	44
2.5	Alfabetização e Letramento em Programação	46
2.6	Metodologias Ativas no Processo de Ensino e Aprendizagem	50
3	PROPOSTA DE SOLUÇÃO: UMA NOVA FORMA DE ENSINAR PROGRAMAÇÃO	53
3.1	<i>Design thinking</i>	54
3.2	Etapas de Ideação	55
3.2.1	Princípios Norteadores - Aprendizagem Colaborativa	56
3.2.2	Princípios Norteadores - Programação em Pares	57
3.2.3	Princípios Norteadores - Aprendizagem Baseada em Projetos	60
3.2.4	Princípios Norteadores - Autoria	61
3.2.5	Princípios Norteadores - Ludicidade	63
3.2.6	Princípios Norteadores - Jogos digitais	65
3.2.7	Princípios Norteadores - Do concreto ao abstrato	66
3.2.8	Princípios Norteadores - Letramento em programação	67
3.2.9	Princípios Norteadores - Pensamento Criativo	68
3.2.10	Princípios Norteadores - Princípio do <i>design</i>	69
3.2.11	Matriz SWOT	69
3.2.12	Planejamento Estratégico	71
3.3	Etapas de Prototipação	74
3.4	Etapas de Testes	83
4	QUADRO EPISTÊMICO-METODOLÓGICO	85
4.1	Método Pesquisa-Formação	85

4.2	Dispositivos de Pesquisa	87
4.2.1	Diário de Pesquisa	87
4.2.2	Questionário <i>online</i>	88
4.2.3	Diário de classe	91
4.2.4	Projeto de Avaliação	91
4.2.5	Histórico de notas	92
4.2.6	Grupo Focal	92
4.3	Cotidiano pesquisado	94
5	RESULTADOS	97
5.1	Diário de Pesquisa	97
5.2	Questionário <i>online</i>	106
5.3	Diário de Classe	112
5.4	Grupo Focal	115
6	DISCUSSÕES	123
6.1	Recomendações práticas	132
7	CONCLUSÃO	135
7.1	Contribuições da pesquisa	137
7.2	Limitações e trabalhos futuros	137
	REFERÊNCIAS	139
	APÊNDICE A – COMPONENTE CURRICULAR LP1	147
	APÊNDICE B – NOVO PLANEJAMENTO DA DISCIPLINA LP1 .	149
	APÊNDICE C – ATIVIDADE ESTACIONAMENTO ALGORÍTMICO	153
	APÊNDICE D – TERMO DE CONSENTIMENTO LIVRE E ES- CLARECIDO	161
	APÊNDICE E – QUESTIONÁRIO	163
	APÊNDICE F – TERMO DE AUTORIZAÇÃO - IMAGEM E SOM	168
	APÊNDICE G – APRESENTAÇÃO GRUPO FOCAL	169
	APÊNDICE H – TRANSCRIÇÃO DO GRUPO FOCAL	177

1 INTRODUÇÃO

O objetivo desta seção é apresentar a jornada formativa de um pesquisador iniciante, bem como uma visão geral da pesquisa que investigou como uma nova abordagem didática, combinada com mudanças nas práticas de ensino, pode ajudar os alunos a superarem as dificuldades encontradas no aprendizado de programação de computadores. O estudo visou proporcionar maior engajamento, motivação e tornar o aprendizado mais atraente e prazeroso.

1.1 Itinerância do pesquisador

No ano de 1996, aos 13 anos de idade, recebi de presente dos meus pais o meu primeiro computador, um Pentium 233Mhz ¹. Até aquele momento, meu contato com computadores se limitava aos encontros de grupos de estudo na casa de colegas. O computador serviu como incentivo para continuar meus estudos e, no mesmo ano, fui aprovado no concurso público para ingressar no ensino médio profissionalizante do curso técnico de informática na Escola Técnica Estadual República (ETER), uma das unidades da rede Fundação de Apoio à Escola Técnica (FAETEC), no Rio de Janeiro. No ano seguinte, com o início da popularização do acesso à internet no país, comecei a navegar na rede mundial de computadores por meio de conexão via linha telefônica.

Até aquele momento, meus conhecimentos de informática se resumiam a operações básicas, como a elaboração e edição de documentos de texto ou planilhas. Conforme as disciplinas do curso técnico avançavam, ampliei meus conhecimentos nas diferentes áreas oferecidas pelo curso. Fiquei fascinado quando me deparei com a área de programação de computadores e como a escrita de um código resultava em um programa que realizava as operações para as quais foi designado. Depois de ter contato com as línguas e plataformas de programação Pascal, Clipper, Delphi e Java, decidi que seguiria a área de programação em minha trajetória profissional.

A FAETEC exigia dos alunos a realização de um estágio técnico obrigatório para obtenção do diploma. Em 1999, participei de processos seletivos e entrevistas até ser aprovado para compor o recém-criado Núcleo de Informática da Casa de Oswaldo Cruz (COC), na Fundação Oswaldo Cruz (FIOCRUZ), onde iniciei minha trajetória profissional. Hoje, 24 anos depois, permaneço ativo no quadro da instituição e tenho vivência em diferentes vínculos e atuações nas mais diversas áreas de Tecnologia da Informação e Comunicação (TIC), com responsabilidades que se modificaram ao longo dos anos.

¹ <https://pt.wikipedia.org/wiki/Pentium>

Durante o período de estágio, atuei nas áreas de suporte ao usuário, manutenção de *hardware*, *software* e redes. Com o término do estágio, fui contratado para minha primeira experiência profissional como bolsista em um programa de aperfeiçoamento profissional ofertado pela instituição. Durante o período de vigência da bolsa, continuei trabalhando na área de suporte ao usuário e manutenção, com o acréscimo de atividades relacionadas ao desenvolvimento e manutenção de *sites* e sistemas. Com o tempo, fui aperfeiçoando meus conhecimentos em programação e, com a experiência adquirida, migrei definitivamente para a área de desenvolvimento de sistemas. Neste momento, já havia iniciado o curso de Bacharelado em Informática na Universidade Estácio de Sá (UNESA), conciliando a formação acadêmica com a área profissional.

O início da graduação me gerou a expectativa de que as disciplinas de programação do curso seriam de pouca complexidade, baseada em minha experiência profissional prévia. Para a minha surpresa, estas disciplinas apresentaram dificuldades das quais não imaginava. O aprendizado de programação no nível superior ocorreu da mesma forma que no ensino técnico, onde os professores apresentaram os conceitos de linguagem de programação e utilização da língua Pascal e C. Para superar estas dificuldades, o primeiro passo foi a formação de um grupo de estudo com outros colegas de turma. Percebi então que os colegas ingressantes no curso que não carregavam nenhuma bagagem prévia em programação, apresentavam dificuldades muito maiores quando comparadas aos demais. Parte destes alunos, ao final do primeiro semestre, não alcançaram a aprovação e acabaram abandonando o curso. Outra parte abandonou o curso antes mesmo do fim do semestre. Para ilustrar este cenário, no primeiro semestre a unidade em que estava matriculado formou 4 turmas com aproximadamente 40 alunos cada, totalizando em torno de 160 estudantes iniciantes. No segundo semestre, as turmas se reduziram a apenas 2, com aproximadamente 40 alunos cada uma.

Em 2003, fui convidado pela FIOCRUZ para participar da formação de *Webmaster*, oferecida em parceria com o NCE/UFRJ. Em 2004, minhas atividades foram totalmente direcionadas para a área de desenvolvimento de sistemas de informação para a *web*, atuando na elaboração e manutenção de *sites* e sistemas. Já em 2009, com a conclusão da graduação, ingressei no curso de especialização em Engenharia de *Software* na Universidade Estácio de Sá, buscando aprimorar meus conhecimentos na área de desenvolvimento de sistemas. No segundo semestre de 2010, fui convidado pela direção da COC/FIOCRUZ para assumir a coordenação do Serviço de Tecnologia da Informação (STI), função na qual me dedico até os dias atuais. Essa função envolve a gestão das áreas de suporte ao usuário, infraestrutura em TIC e desenvolvimento de sistemas.

Minha experiência acadêmica teve início em 2011, quando fui selecionado como tutor presencial no curso de Tecnologia em Sistemas de Computação da UFF, integrado ao consórcio CEDERJ. O objetivo desse consórcio é democratizar o acesso ao ensino superior

público, gratuito e de qualidade na modalidade de Educação a Distância (EaD) por meio de um acordo de cooperação técnica entre o Governo do Estado do Rio de Janeiro, a Secretaria de Ciência, Tecnologia e Inovação do Estado (SECTI), a Fundação Centro de Ciências e Educação Superior à Distância do Estado do Rio de Janeiro (CECIERJ) e as Instituições de Ensino Superior (IES) públicas sediadas no Estado do Rio de Janeiro.

Como tutor presencial, minhas atividades consistiam em aplicar provas, corrigir atividades dos alunos e conduzir sessões de tutoria para os estudantes matriculados no polo Nova Iguaçu. As disciplinas sob minha responsabilidade eram Construção de Páginas *Web* (CPW) e Fundamentos de Programação (FP). Com o passar do tempo, a disciplina Fundamentos de Programação foi substituída pela disciplina Programação e Desenvolvimento de Algoritmos (PDA). A disciplina PDA, presente no primeiro semestre do curso, é de suma importância para apresentar aos alunos os conceitos de programação e linguagem de programação, sendo essencial para a continuidade do curso no contexto da programação.

Meu desejo de ser tutor responsável por esta disciplina foi motivado pela repetição de um mesmo cenário ao longo dos semestres: os alunos aprovados em PDA ao cursar FP apresentavam enormes dificuldades na disciplina. Durante as sessões de tutoria, percebi que a dificuldade não estava relacionada às línguas Pascal ou Python utilizadas na disciplina, mas sim a conceitos fundamentais de programação, como o uso de estruturas condicionais ou de repetição. Os alunos não conseguiam elaborar um algoritmo que resolvesse o problema proposto pelo professor, mesmo com o auxílio de pseudocódigo. Esse problema acarretava um alto índice de reprovação na disciplina, tornando-a um dos maiores obstáculos do curso.

Em minha avaliação, identifiquei duas causas principais para esse problema. A primeira é que a disciplina introdutória de programação (PDA) não enfatizava a necessidade de prática constante de programação para fortalecer o aprendizado. A segunda causa é o formato de avaliação, que não estimulava os alunos a elaborar códigos nas avaliações, independentemente de ocorrerem à distância ou presencialmente. As avaliações eram compostas por questões que apresentavam trechos de código e opções de respostas para o estudante selecionar.

Como tutor, percebi que nas sessões de tutoria os alunos encontravam dificuldades em programar, de maneira similar aos meus colegas de graduação. Eu pude contribuir e amenizar esta situação com o ensino de programação por meio da prática, fazendo com que os estudantes “colocassem a mão na massa” e escrevessem códigos na ferramenta indicada pelo professor da disciplina, o PETEQS. Com essa pequena modificação nas sessões de tutoria, notei que os alunos aprovados na disciplina PDA conseguiram cursar a disciplina FP melhor preparados.

Em 2014, ampliei minha atuação como docente ao receber a convocação para atuar

como professor de informática na rede FAETEC. Inicialmente, lecionei disciplinas que não eram minha especialidade, como rede de computadores e aplicativos gráficos. Com o passar dos semestres, fui designado para as disciplinas de programação, como Programação para *Web* (PWEB) e Linguagem de Programação 1 (LP1). A disciplina LP1 é responsável pela introdução dos conceitos de programação aos estudantes e, de acordo com a matriz curricular, o professor deve lecionar partindo de algoritmos até a língua Java, que será utilizada nos demais semestres do curso. O Apêndice A apresenta a matriz curricular atualizada da disciplina disponibilizada pela FAETEC para que todos os docentes das unidades escolares a sigam.

A disciplina LP1 sempre foi considerada pelos professores e funcionários da escola como uma disciplina difícil, que causa apreensão nos alunos e tem um alto índice de reprovação em todos os três turnos do curso. Historicamente, a disciplina também apresenta um alto índice de evasão, pois muitos alunos se sentem incapazes de aprender programação, o que leva à desmotivação e conseqüentemente ao abandono do curso antes do final do semestre. Além disso, parte dos alunos ingressantes no curso de informática também não possuem experiência com computadores ou programação.

O curso é oferecido na modalidade técnico de informática subsequente ou concomitante, o que significa que está disponível para qualquer pessoa que já tenha concluído o ensino médio ou para estudantes que cursam o nível médio a partir do segundo ano. A maioria dos alunos matriculados no curso vem do ensino público e apresentam muitas dificuldades em sua formação, especialmente em conceitos matemáticos e interpretação de textos.

A cada semestre, busquei lecionar a disciplina de forma diferente, com práticas novas e menos complexas, na esperança de reduzir a evasão e a reprovação. No entanto, esses esforços não foram suficientes para melhorar significativamente o desempenho dos alunos na disciplina.

Em 2019, candidatei-me a uma vaga do mestrado em Informática do Programa de Pós-Graduação em Informática (PPGI) da Universidade Federal do Estado do Rio de Janeiro (UNIRIO), na linha de pesquisa de Conversação em Redes na Educação (CRED), coordenada pelo Professor Mariano Pimentel. Essa linha de pesquisa estava alinhada às minhas expectativas, pois estava voltada para a Informática na Educação e poderia proporcionar experiência e aprendizado como docente e pesquisador.

Inicialmente, planejava pesquisar e trabalhar com uma ferramenta de bate-papo para apoiar a mediação da aula virtual em um ambiente suportado por tecnologia digital. No entanto, após conversas com o orientador, decidi mudar o tema da minha pesquisa e investigar as dificuldades enfrentadas pelos alunos iniciantes no ensino de programação nos cursos de ensino médio. Essa mudança foi motivada pela minha vivência em sala de aula, onde identifiquei os problemas relatados pelos alunos e a dificuldade em alcançar

resultados diferentes no ensino de programação.

1.2 A pandemia da covid-19

O ano de 2020 ficou marcado pelo surto da covid-19, uma doença causada pelo vírus SARS-CoV-2. Em janeiro daquele ano, a Organização Mundial da Saúde (OMS) declarou emergência de saúde pública de importância internacional (WELSH, 2020). Em 11 de março de 2020, a pandemia foi oficialmente declarada, devido ao surto da covid-19 em várias regiões do mundo, exigindo que as autoridades governamentais tomassem medidas para conter a disseminação da doença. Essas medidas foram implementadas através de políticas públicas emergenciais, publicadas em instrumentos legais e normas.

Entre as medidas sugeridas pela OMS para prevenir a covid-19, destacam-se o distanciamento físico, o isolamento e a quarentena. No Brasil, de acordo com a Lei nº 13.979², de 6 de fevereiro de 2020, que dispõe sobre as medidas para enfrentamento da emergência de saúde pública decorrente da covid-19, considera-se:

I - Isolamento: separação de pessoas doentes ou contaminadas, ou de bagagens, meios de transporte, mercadorias ou encomendas postais afetadas, de outros, de maneira a evitar a contaminação ou a propagação do coronavírus; e

II - Quarentena: restrição de atividades ou separação de pessoas suspeitas de contaminação das pessoas que não estejam doentes, ou de bagagens, contêineres, animais, meios de transporte ou mercadorias suspeitos de contaminação, de maneira a evitar a possível contaminação ou a propagação do coronavírus.

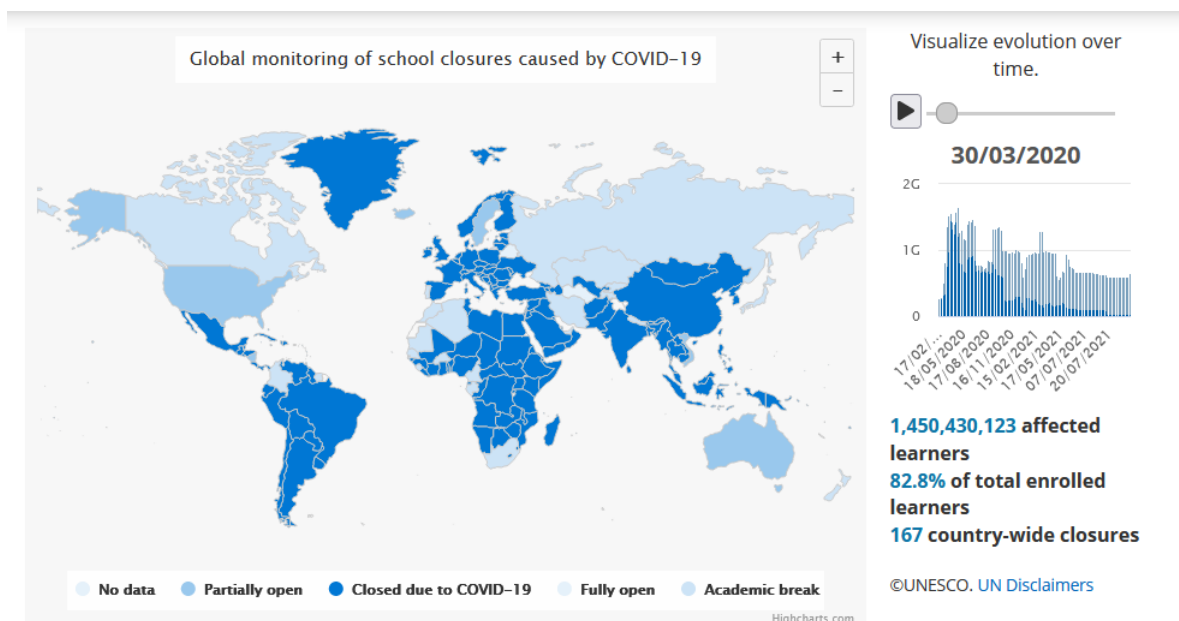
O distanciamento físico, de acordo com Wilder-Smith e Freedman (2020), é uma medida utilizada para reduzir as interações entre as pessoas de uma comunidade, que podem estar infectadas, mas ainda não foram identificadas com a doença. Juntamente com medidas de higienização, o distanciamento físico foi considerado uma das principais medidas de prevenção ao contágio da covid-19.

Como sugerido pela OMS e adotado na maioria dos países do mundo, o distanciamento físico levou ao fechamento das escolas e à suspensão das aulas presenciais na rede pública e privada em níveis básico e superior (ALMEIDA; ALVES, 2020). A UNESCO, órgão responsável pelo monitoramento dos impactos da pandemia na educação, apresenta dados sobre o fechamento das instituições de ensino e como isso afeta a população estudantil em todo o mundo. De acordo com o monitoramento, em março de 2020, no ápice da pandemia, cerca de 1,450,430,123 alunos foram afetados em todo o mundo. Esse número corresponde a 82,8% do total de alunos matriculados em todo o planeta, em 167 países,

² BRASIL. *LEI Nº 13.979, DE 6 DE FEVEREIRO DE 2020*. [S.l.], 2020. Disponível em: <<https://www.in.gov.br/en/web/dou/-/lei-n-13.979-de-6-de-fevereiro-de-2020-242078735>>. Acesso em: 12 ago. 2021.

como mostra a Figura 1.

Figura 1. Monitoramento do fechamento das instituições por covid-19 no mundo



Fonte: UNESCO (2020)

No contexto educacional do Brasil, o Ministério da Educação aprovou a Portaria nº 343³ em 17 de março de 2020, suspendendo as aulas presenciais durante a covid-19 e substituindo-as por atividades baseadas em meios digitais. Esse processo ficou conhecido como ensino remoto emergencial e provocou a adoção de tecnologias digitais sem a realização de um planejamento pedagógico completo, apenas adaptando o conteúdo antes veiculado em sala de aula para o ambiente *online*. É um ensino praticado com o uso de tecnologias, mas com os mesmos padrões e princípios do modelo tradicional (SANTOS et al., 2020).

O ensino remoto emergencial impactou estudantes e docentes de forma geral. Com a suspensão das aulas presenciais, os professores precisaram se adaptar rapidamente para oferecer aos estudantes a possibilidade de continuidade dos estudos na modalidade *online*. Na FAETEC, houve uma tentativa de padronizar o ensino para todas as unidades escolares. Os professores foram orientados a disponibilizar aos estudantes todo o material da disciplina e ofertar encontros semanais para discussão dos conteúdos. Esses encontros, com duração de 2 horas, funcionavam de maneira semelhante a uma sessão de tutoria, dada a limitação de tempo disponível.

A baixa adesão dos alunos foi identificada como uma das principais características do período do ensino remoto emergencial. Muitos alunos e suas famílias não estavam

³ BRASIL. PORTARIA Nº 343, DE 17 DE MARÇO DE 2020. [S.l.], 2020. Disponível em: <<https://www.in.gov.br/en/web/dou/-/portaria-n-343-de-17-de-marco-de-2020-248564376>>. Acesso em: 12 ago. 2021.

preparados para esse cenário, com ausência de banda larga e/ou computador em suas residências, o que os obrigava a desistir do curso naquele momento. Durante os 4 semestres afetados pela pandemia, as turmas eram compostas por no máximo 5 alunos, e muitos acabavam desistindo antes do encerramento do semestre letivo.

Os problemas percebidos pelo pesquisador antes da pandemia se intensificaram durante o ensino remoto emergencial, com baixa adesão dos alunos, o que provocou um cenário atípico. Esses fatores foram considerados fundamentais durante o planejamento da pesquisa e concluiu-se que os anos de ensino remoto emergencial, 2020 e 2021, não seriam incluídos no escopo do trabalho.

1.3 Problematização: dificuldades em programação

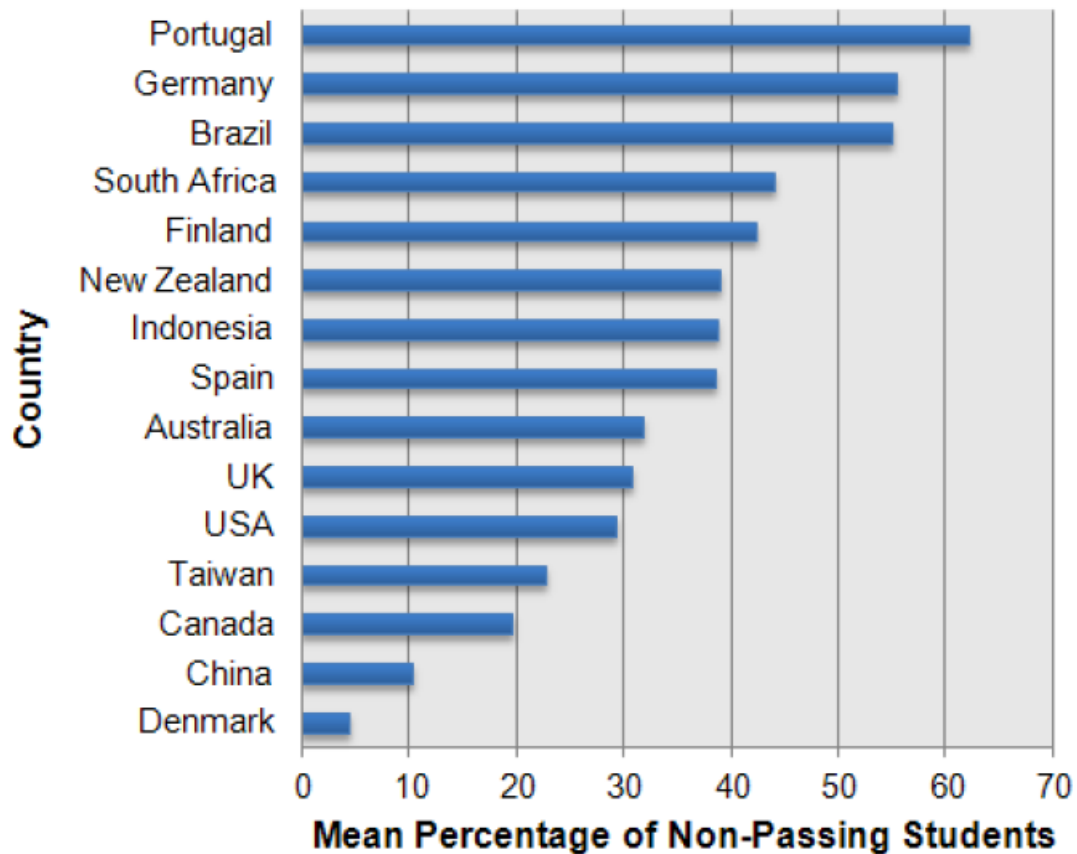
O ensino de programação de computadores está presente em disciplinas dos cursos de Ciência, Tecnologia, Engenharia e Matemática (STEM). Na disciplina introdutória, são apresentados os conhecimentos necessários para cursar as demais disciplinas no contexto de programação, tradicionalmente com um foco direcionado para o aprendizado por meio de algoritmos. No entanto, estudos recentes apontam que essas disciplinas têm apresentado um elevado grau de evasão e reprovação em todo o mundo, sendo ainda maior quando consideramos o cenário nacional (BENNEDSEN; CASPERSEN, 2019; MEDEIROS, 2019).

O processo de ensino e aprendizagem de programação de computadores é um desafio significativo para professores e estudantes, e pode ser considerado um dos principais obstáculos em cursos de computação, informática e áreas afins (GOMES et al., 2023; MABEL et al., 2023). A origem desse problema é ampla e pode envolver várias causas, como a falta de habilidades para resolver problemas, falta de habilidades matemáticas, baixo nível de abstração, dificuldades na interpretação do problema e dificuldades na compreensão de textos por parte dos estudantes. Além disso, parte do problema pode estar relacionada às práticas docentes utilizadas no processo de ensino e aprendizagem e à formação pedagógica (ou falta dela) desses docentes.

Além da reprovação e evasão, a falta de motivação dos estudantes também pode ser apontada como um dos problemas enfrentados por cursos de programação de computadores. Dados provenientes da década anterior revelam que a taxa média global de reprovação nesses cursos era de aproximadamente 30% (WATSON; LI, 2014). No contexto nacional, essa taxa ultrapassava a média mundial (BOSSE; GEROSA, 2015). Em 2018, (RABELO; MAIA; PARREIRAS, 2018) constataram que 43% dos alunos brasileiros em cursos de computação enfrentaram dificuldades em disciplinas introdutórias de programação, incluindo um preocupante índice de abandono de 25% da disciplina. A Figura 2 ilustra o percentual de alunos reprovados em programação por país no ano de 2014. O Brasil ocupava o terceiro lugar nesse ranking, com aproximadamente 55% de reprovação, praticamente empatado

com a Alemanha. Neste cenário, Portugal liderava a estatística. Contudo, vale ressaltar que metade dos países avaliados apresentava índices inferiores à média global.

Figura 2. Percentual de reprovação de estudantes por país em programação



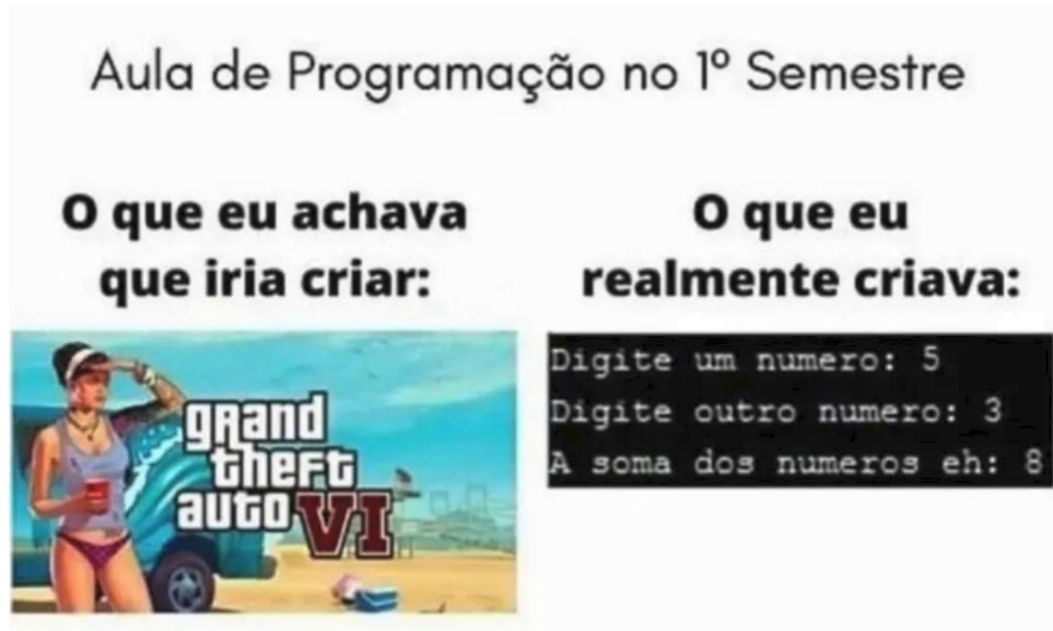
Fonte: Watson e Li (2014, p. 42)

Entre os fatores que possivelmente contribuem para essa situação, destacam-se: i) a carência de habilidades abstratas e raciocínio lógico necessários para desenvolver soluções algorítmicas; ii) a ausência de motivação por parte dos estudantes, que frequentemente encaram a disciplina como um grande obstáculo a ser superado; e iii) a abordagem de ensino que pode não ser capaz de estimular o interesse dos estudantes (JENKINS, 2002).

É comum que as disciplinas de introdução à programação sejam ensinadas por meio de uma abordagem tradicional, que se concentra em conceitos, algoritmos, fluxogramas, repetição e resolução de problemas baseados principalmente na matemática. No entanto, essa abordagem pode não proporcionar motivação e interesse por parte dos estudantes, que muitas vezes têm seu primeiro contato com a programação durante a disciplina. Como resultado dos desafios e dificuldades observados, professores e pesquisadores têm se dedicado, nos últimos anos, a estudar essas causas e a desenvolver soluções para tornar o aprendizado da programação mais atraente e prazeroso para os estudantes. Recentemente, recebi de um aluno um meme pelo WhatsApp, ilustrado na Figura 3, que demonstra como

essa abordagem tradicional pode ter um impacto negativo na experiência do aluno no universo da programação.

Figura 3. Meme da internet sobre expectativa dos estudantes em programação



Fonte: Grupo de WhatsApp

O pesquisador vivencia essa realidade como docente da disciplina Linguagem de Programação 1 (LP1). O curso de informática, com ênfase em desenvolvimento de sistemas, é organizado em três módulos com duração de seis meses cada. A disciplina LP1 é responsável pela introdução da área de programação aos estudantes que ingressam no curso. Como em outros cursos, a estrutura curricular proposta pela FAETEC, disponível no Apêndice A, direciona o corpo docente a adotar uma abordagem didática tradicional, em que o professor apresenta aos estudantes os conceitos de lógica de programação com o apoio de fluxogramas, pseudocódigo e propõe aos alunos basicamente a solução de problemas como cálculo de uma média escolar, área de uma figura geométrica ou verificar se um determinado número é primo.

Em grande parte, os estudantes que ingressam no curso de informática da Escola Técnica Estadual Maria Mercedes Mendes Teixeira (ETEMMMT) não possuem qualquer experiência prévia com programação e desconhecem o que o curso técnico oferece. Muitos dos alunos acreditam estar ingressando em um curso em que irão adquirir conhecimentos em informática básica, antigamente chamado de microinformática, como o uso de sistema operacional Windows e pacote de escritório, com a elaboração de textos, confecção de planilhas e apresentações com slides.

Historicamente, a cada semestre, cerca de 40% dos estudantes abandonam ou reprovam a disciplina LP1. Considerando que a FAETEC recomenda que as turmas do primeiro semestre sejam compostas por 30 estudantes, podemos considerar que, ao final de

cada semestre, 12 abandonam ou são reprovados. Parte desses alunos perde a motivação para permanecer no curso por acreditar que não serão capazes de adquirir os conhecimentos necessários para escrever algoritmos ou códigos de programação. Pode-se considerar que o uso de uma abordagem didática tradicional e práticas docentes conservadoras na disciplina LP1 contribui para a desmotivação dos estudantes e a manutenção do percentual elevado de abandono e reprovação na disciplina.

Portanto, como é possível sustentar a motivação e o engajamento dos estudantes, reduzir as taxas de reprovação e desistência em cursos de programação, e fornecer um processo de aprendizagem mais acessível e envolvente?

1.4 Questões de pesquisa

A partir do cenário identificado na seção de problematização (1.3), esta pesquisa propõe uma nova abordagem didática para o ensino de programação que seja capaz de manter o engajamento dos alunos, tornando o aprendizado simples e prazeroso. Para alcançar esses objetivos, apresentam-se as questões de pesquisa que serão respondidas ao final do trabalho. São elas:

1. Uma nova abordagem didática no ensino de programação pode proporcionar aos estudantes um processo de aprendizagem mais simples, interessante e motivador?
2. A adoção da abordagem didática proposta pode tornar a aprendizagem prazerosa e engajante?
3. A adoção da abordagem didática proposta pode reduzir o percentual de evasão e reprovação?
4. A adoção da abordagem didática proposta pode reduzir a curva de aprendizagem em programação?
5. A adoção da abordagem didática proposta pode potencializar o comprometimento dos estudantes?

1.5 Visão da pesquisa

A seção 2 abordará o quadro teórico que embasou a pesquisa, abrangendo conceitos que estimularam reflexões conceituais relacionadas a temas como Programação e Codificação, Pensamento e Raciocínio Computacional, Linguagem e Língua de Programação, Ensino e Aprendizagem de Programação, bem como Alfabetização e Letramento em Programação.

Com o intuito de solucionar o problema descrito na seção 1.3, propomos uma nova abordagem didática para o ensino de programação, substituindo a abordagem tradicionalmente utilizada. Na seção 3, apresentaremos as técnicas e o processo percorrido durante essa jornada, abrangendo todo o planejamento realizado, que incluiu a identificação e seleção de técnicas, práticas e metodologias ativas, a utilização de artefatos de planejamento estratégico, a (re)organização da disciplina Linguagem de Programação 1, o novo planejamento desenvolvido e a avaliação da solução proposta.

Na seção 4, descrevemos a abordagem epistemológica-metodológica da pesquisa e os métodos e procedimentos científicos utilizados para conduzir a avaliação, produção e interpretação dos dados coletados. Utilizamos o método Pesquisa-Formação para planejar e aplicar a nova abordagem proposta e as práticas docentes utilizadas para o ensino de programação com estudantes da disciplina de introdução a programação do curso técnico de informática em uma das unidades da FAETEC/RJ.

A seção 5 apresentará os resultados obtidos por meio dos dispositivos de avaliação utilizados na pesquisa. Em seguida, na seção 6, discutiremos os resultados anteriormente apresentados, avaliando os dados coletados e aprofundando a análise das respostas para as questões de pesquisa. Encerraremos a seção com uma conclusão dos achados e possíveis implicações para o ensino de programação.

Na seção 7, apresentamos as conclusões desta pesquisa, ressaltando suas principais contribuições, bem como abordando suas limitações e possíveis direções para trabalhos futuros.

2 QUADRO TEÓRICO: FUNDAMENTAÇÃO DA PESQUISA

O objetivo desta seção é apresentar reflexões e discussões sobre os conceitos fundamentais que orientaram esta pesquisa, incluindo Programação e Codificação, Pensamento e Raciocínio Computacional, Linguagem e Língua de Programação, Ensino e Aprendizagem de Programação, Alfabetização e Letramento em Programação e Metodologias Ativas no Processo de Ensino e Aprendizagem. O pesquisador espera que, ao discutir esses conceitos, os leitores sejam instigados a refletir sobre o conteúdo que fundamentou a proposta de solução para a pesquisa, uma vez que ensinar vai além de transmitir informações sistematizadas ou arquitetar situações instrucionais que orientem o processo de ensino e aprendizagem.

2.1 Programação e Codificação

Ao buscar o termo “o que é programação” no Google, encontramos definições que descrevem a programação como um processo de criação de instruções que direcionam o computador a realizar uma tarefa. No entanto, essa definição tradicional é frequentemente ensinada de forma robotizada em cursos de programação, tanto presenciais quanto *online*. Esta seção busca refletir sobre um novo viés do ensino de programação, que envolve um processo criativo.

Como professor, profissional e pesquisador da área de Computação, posso afirmar que durante todo o meu processo formativo fui ensinado a programar de forma robotizada. Eu simplesmente recebia os conceitos compartilhados pelo professor em uma aula expositiva e os repetia sem refletir sobre o motivo de realizá-los daquela maneira. Isso resultava em aulas tediosas e pouco produtivas, onde os estudantes simplesmente decoravam os códigos construídos em sala de aula e os replicavam nos exercícios e avaliações.

Em uma ocasião durante a graduação, um professor descontou pontos da minha avaliação por eu não ter nomeado as variáveis e funções como ele havia indicado em sala de aula. Essa experiência ilustra como o ensino de programação tradicional muitas vezes se preocupa com detalhes irrelevantes em vez de incentivar a criatividade e a compreensão dos conceitos fundamentais.

Os exemplos de ensino de programação que vivenciei anos atrás ainda são comuns hoje em dia e são replicados por muitos colegas docentes que ensinam programação. Essa é uma crítica que surgiu a partir de uma auto-reflexão durante a pesquisa e que

precisa ser compartilhada com outros docentes. Programar é muito mais do que o que é geralmente proposto pelos professores. Embora seja uma disciplina relacionada às exatas, a programação pode ser vista como um processo criativo, em que cada aluno precisa experimentar, errar e tentar novamente até atingir seu objetivo.

Como atleta amador e praticante de esportes, encontro uma interessante metáfora ao comparar a programação com a prática da corrida. Assim como estabeleço metas e traço um plano de treinamento para alcançar um objetivo específico em uma competição de corrida, na programação também é essencial criar um planejamento estruturado. Ambos os processos requerem disciplina, rotinas específicas, esforço contínuo, e até mesmo ajustes durante o percurso para atingir o melhor resultado. Assim como no esporte, programar envolve tanto a busca por aprimoramento contínuo quanto a capacidade de adaptar-se a novos desafios, reavaliando e corrigindo a rota, caso necessário, para alcançar os melhores resultados.

Um indivíduo que está dando seus primeiros passos na programação, sem qualquer conhecimento prévio, precisa contar com o auxílio de um professor que o apoie nessa jornada através de tentativas, erros e descobertas sucessivas. Ao lecionar programação para meus alunos, procuro estabelecer um plano gradual que lhes permita atingir níveis crescentes de maturidade na área ao longo do tempo.

Fazendo uma analogia com a prática esportiva, o aprendizado de programação pode ser comparado a uma corrida, em que o nível de conhecimento equivale à distância percorrida. Assim como alguém que inicia sua jornada de corrida com caminhadas curtas e, gradualmente, alterna entre caminhadas e corridas até alcançar a marca de 5 km, nossos alunos também vão evoluindo passo a passo.

Posteriormente, eles poderão progredir para distâncias maiores, como 10 km, meia maratona (21 km) e, por fim, alcançar o tão almejado objetivo de uma maratona completa (42 km). O aprendizado da programação, de forma similar ao esporte, requer esforço e comprometimento para alcançar os objetivos propostos.

Dessa forma, ao encorajar nossos alunos a persistirem e enfrentarem os desafios do aprendizado, estaremos capacitando-os a conquistar uma sólida base de conhecimento em programação e, conseqüentemente, a alcançarem suas metas e aspirações nessa área.

Além da tendência de se tornar “robotizado” outra característica comum na aprendizagem da programação é a ênfase exclusiva em exemplos matemáticos. Isso é algo que frequentemente vivenciamos como alunos e, sem perceber, acabamos reproduzindo como professores em nossas salas de aula. É verdade que muitos professores de programação utilizam uma lista de exercícios com problemas matemáticos, como calcular a média dos alunos, a área de uma figura geométrica ou o índice de massa corporal (IMC) de uma pessoa.

Entretanto, a crítica não está no uso de exemplos matemáticos, mas sim na limitação exclusiva a esse escopo. A programação é uma disciplina extremamente versátil, com aplicações em diversas áreas, e focar apenas em exemplos matemáticos pode deixar de explorar todo o potencial e a abrangência dessa habilidade.

Como professores, temos a responsabilidade de diversificar os exemplos e exercícios, buscando contextos do mundo real que possam despertar o interesse e a criatividade dos alunos. Isso não apenas tornará a aprendizagem mais envolvente, mas também permitirá que eles vejam a programação como uma ferramenta poderosa para resolver problemas em diferentes domínios, desde a ciência e tecnologia até a arte, saúde, entretenimento e muito mais.

Portanto, ao enriquecer o conteúdo da programação com exemplos variados e contextualizados, estaremos preparando nossos alunos para enfrentar desafios diversos e estimulando seu pensamento crítico e inovador, essenciais para o mundo atual e futuro.

Quando fui selecionado para ministrar a disciplina de Linguagem de Programação 1, preparei cuidadosamente meu material e busquei orientação de outros colegas em relação aos conteúdos abordados. No entanto, à medida que amadureci como professor, percebi que estava cometendo um equívoco em sala de aula, ao utilizar listas de exercícios focadas exclusivamente em problemas matemáticos. Reconheci a necessidade de abandonar essa abordagem fria e matemática, e adotar um processo mais criativo e envolvente.

Um ponto de reflexão importante surgiu: como poderíamos transformar a forma como ensinamos e motivamos nossos alunos? Nessa busca por uma metodologia mais interessante, deparei-me com um artigo inspirador sobre “Atividades Autorais” (CARVALHO; PIMENTEL, 2020), onde um dos autores relata como estimula os alunos a desenvolverem códigos para situações que vão além dos tradicionais problemas matemáticos. Ele propõe o uso de exemplos que façam parte do cotidiano dos alunos ou até mesmo desafios que incentivem a criatividade deles.

No artigo, o professor Pimentel (2020) relata uma abordagem pedagógica em que combinou o uso de um exemplo tradicional - uma calculadora de IMC - com a proposta de que cada estudante desenvolvesse sua própria calculadora. Os resultados obtidos foram surpreendentemente diversos, com os alunos criando programas para resolver problemas variados e interessantes.

Entre os projetos desenvolvidos, destacam-se um programa para calcular o custo de um churrasco com base no número de convidados, outro para avaliar o grau de vício em internet em relação ao tempo de conexão e ainda uma calculadora para estimar o tempo gasto no trajeto de casa para a faculdade ao longo da semana.

Essa abordagem demonstra como a programação pode ser uma ferramenta poderosa e versátil para resolver questões do cotidiano, extrapolando os limites tradicionais dos

exemplos matemáticos. Ao permitir que os estudantes explorem suas próprias ideias e interesses, a aprendizagem se torna mais envolvente e significativa para eles.

Além disso, essa metodologia estimula a criatividade dos alunos, encorajando-os a encontrar soluções originais para os problemas apresentados. Essa liberdade de criação também promove o desenvolvimento de habilidades de resolução de problemas e pensamento crítico, fundamentais para a formação de estudantes preparados para os desafios do mundo contemporâneo.

Em suma, ao incorporar exemplos do cotidiano dos alunos e incentivar a criação de projetos autorais, o professor Pimentel (2020) demonstrou como é possível tornar o ensino de programação mais atrativo e relevante, capacitando os estudantes a aplicarem seus conhecimentos de forma prática e inovadora.

Em minha disciplina, buscando uma abordagem mais abrangente e envolvente no ensino de programação, decidi não apenas utilizar os exemplos matemáticos tradicionais, mas também inserir jogos autorais como parte do processo de aprendizagem. Essa abordagem lúdica visa estimular o interesse e o engajamento dos alunos, tornando o aprendizado mais divertido e prático.

Além disso, incentivo os estudantes a explorarem soluções voltadas para o contexto da computação, como a criptografia de mensagens de texto ou o desenvolvimento de um programa de correção automática de provas. Esses projetos mais específicos da área de computação permitem que eles se familiarizem com conceitos e técnicas essenciais para a prática da programação.

Outra maneira pela qual busco tornar o aprendizado mais relevante e aplicável é encorajando a utilização de exemplos do cotidiano das pessoas. Um exemplo disso é o cálculo do valor de multas por atraso no pagamento de boletos. Ao aplicar os conhecimentos de programação em situações reais, os alunos compreendem a importância e o impacto da programação no mundo ao seu redor.

Essa abordagem diversificada permite que os estudantes desenvolvam suas habilidades de programação de forma mais completa, indo além dos conceitos teóricos e exercitando sua criatividade na resolução de problemas práticos. Através da criação de jogos, da exploração de temas específicos da computação e da aplicação em exemplos do dia a dia, eles se tornam mais preparados para enfrentar desafios variados e se tornarem profissionais competentes e inovadores na área da programação.

Acredito que as reflexões e críticas apresentadas aqui possam servir de inspiração para outros colegas docentes refletirem sobre seus próprios processos de ensino de programação. É gratificante observar a felicidade dos estudantes ao apresentarem seus projetos, sejam eles programas ou jogos, como avaliação final da disciplina.

Para encerrar esta seção, gostaria de compartilhar a Figura 4, desenvolvida por um

dos estudantes mencionados no artigo, que ilustra a relação especial que ele estabeleceu com a programação por meio da autoria e criatividade.

Figura 4. Ilustração elaborada por estudante ao final da disciplina

magicódigo

fazer código é mágica!



*hoje foi o último dia das minhas férias da faculdade,
e eu não poderia esperar mais pra ver meus amigos de lá!*

Fonte: Facebook¹

A ilustração criada pelo aluno retrata de forma brilhante que a programação e codificação não precisam ser vistas como atividades chatas e monótonas, mas sim como

¹ <https://www.facebook.com/raenkumi/posts/1077721395765272>

um processo criativo e divertido. Essa representação visual ressalta a importância de (re)pensarmos os conteúdos e abordagens utilizadas no ensino de programação, com o objetivo de promover o engajamento dos alunos e tornar a aprendizagem mais simples e prazerosa.

Como docentes, é essencial refletir sobre a forma como apresentamos os conceitos e as situações de aprendizagem relacionadas à programação. Ao adotar métodos mais envolventes e interativos, podemos despertar o interesse dos alunos e estimular sua criatividade na resolução de problemas.

2.2 Pensamento e Raciocínio Computacional

A discussão sobre o termo Pensamento Computacional (PC) emergiu na primeira década do século XXI e reconhece-o como um conjunto de habilidades que potencializam a capacidade humana de abordar problemas de forma científica em diversas áreas do conhecimento, estendendo-se além do campo específico da computação (WING, 2016; SELBY; WOOLLARD, 2013; RIBEIRO et al., 2019). Embora essa discussão seja relativamente recente, pode-se considerar Seymour Aubrey Papert como um dos pioneiros, pois ele proporcionou aos estudantes um método para resolver problemas utilizando soluções algorítmicas e computadores (PAPERT, 1980). Com a publicação do estudo intitulado “Computational Thinking” realizado por Wing (2006), o assunto ganhou maior interesse e atenção de pesquisadores ao redor do mundo.

Seymour Aubrey Papert (1928-2016), renomado matemático e educador, foi um visionário ao antecipar o uso de computadores por estudantes em salas de aula, muito antes de eles se tornarem populares. Ele compreendeu o potencial dessas tecnologias para revolucionar o processo de ensino e aprendizagem e também para estimular a criatividade infantil. Além de suas contribuições no âmbito educacional, Papert foi um verdadeiro pioneiro no campo da inteligência artificial e desfrutou de grande influência no desenvolvimento de tecnologias educacionais inovadoras.

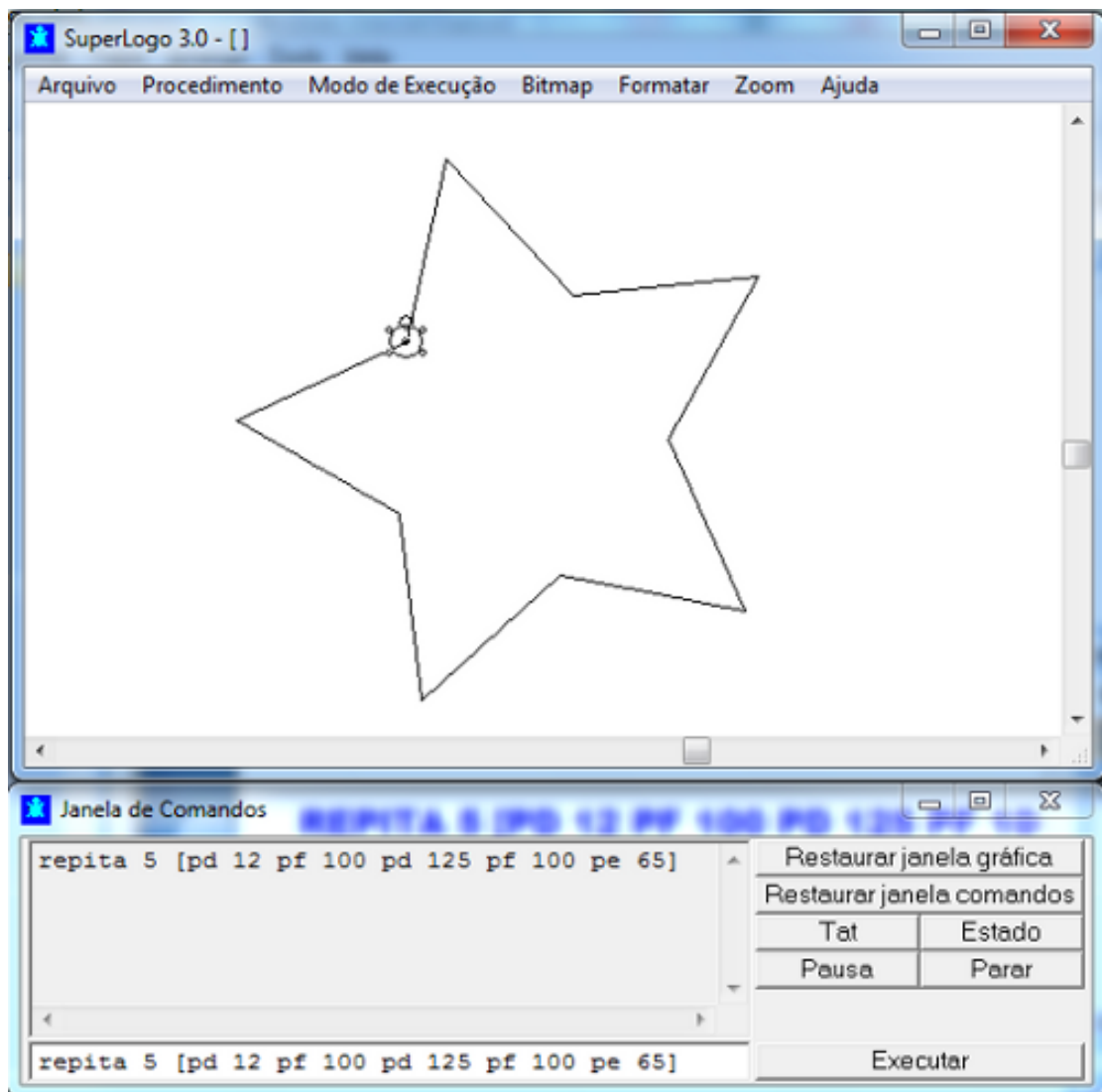
Papert é uma figura fundamental no campo da tecnologia educacional, especialmente no que diz respeito ao uso de computadores no processo de aprendizagem. Ele percebeu que os computadores possuíam um potencial único para permitir que as crianças experimentassem, explorassem e se expressassem, ao mesmo tempo em que forneciam informações e orientações valiosas.

Além disso, Papert apresentou a ideia inovadora de que a programação e a depuração de computadores poderiam ser uma ferramenta poderosa para ajudar as crianças a refletirem sobre seu próprio pensamento e processo de aprendizado. Ao interagirem com o código e depurarem seus programas, os alunos desenvolvem habilidades metacognitivas, ou seja, tornam-se mais conscientes de seus próprios processos mentais e aprendem a identificar e

resolver problemas.

Durante o período de 1967 a 1968, Papert e uma equipe de pesquisadores desenvolveram a língua de programação LOGO com o objetivo de capacitar as crianças a controlar o computador. Naquela época, o computador era considerado a tecnologia mais avançada disponível, e a língua LOGO permitiu que as crianças pudessem programar a máquina, em vez de serem meramente controladas por ela. A imagem na Figura 5 ilustra o desenho de uma estrela obtido por meio de instruções na língua LOGO.

Figura 5. Desenho de uma estrela por meio da língua LOGO



Fonte: Informática e Conhecimento - Ferramentas da Educação²

Papert é uma figura renomada e uma influente fonte teórica para estudos e pesquisas no campo da educação, com destaque especial para a utilização da programação de computadores como ferramenta de ensino e aprendizagem para crianças na era digital. A Figura 6 ilustra uma seleção de obras elaboradas por Papert, incluindo o LEGO Mindstorms

² <http://infobook.6te.net/logo/logo.html>

Education EV3, os livros “Mindstorms: Children, Computers, and Powerful Ideas”, “As Máquinas das Crianças” e “LOGO”, bem como o projeto “One Laptop per Child”.

Figura 6. Vida e obra de Papert



Fonte: elaborado pelo autor

Papert defendia que a aprendizagem era mais eficaz quando envolvia um processo dinâmico de modelagem e assimilação, em que os modelos ajudavam a tornar ideias abstratas mais acessíveis e concretas. Por exemplo, um aluno poderia aprender o conceito de “variável” ao programar um computador (modelo) e, em seguida, compreender o conceito de “incógnita” em uma equação matemática (assimilação) com base nessa experiência.

Ele também enfatizava que a aprendizagem não era apenas uma questão cognitiva, mas também dependia de aspectos emocionais. Isso ocorria porque a aprendizagem significativa envolvia experiências vivenciadas e assimiladas de maneira pessoal, o que permitia que esses conhecimentos fossem aplicados em outros contextos de aprendizado.

As ideias de Papert tornaram-se referências teóricas importantes para pesquisas e produção científica em educação, especialmente no contexto do ensino e aprendizagem na era digital e na programação de computadores para crianças (PAPERT; VALENTE; BITELMAN, 1980).

Papert sustentava que a educação frequentemente adotava uma abordagem tecnicista, limitando o processo de aprendizagem a técnicas e relegando o papel do professor a um mero transmissor de informações. No entanto, ele viu a chegada do computador como uma

oportunidade para transformar esse cenário e promover uma verdadeira “megamudança” na educação, diminuindo a ênfase em abordagens puramente técnicas.

Para alcançar esse objetivo, Papert defendia que os educadores deveriam estudar materiais relevantes para o desenvolvimento intelectual dos alunos e adaptar-se às tendências tecnológicas da sociedade. Isso permitiria que eles realizassem intervenções significativas ao incorporar o computador como uma ferramenta valiosa no processo educacional. (CYSNEIROS, 2008)

Papert cunhou o termo “construcionismo”, uma teoria que tem como base o construtivismo de Jean Piaget. De acordo com essa teoria, os alunos são capazes de construir conhecimento por meio da manipulação de materiais concretos, em vez de se limitarem a proposições abstratas. O construcionismo enfatiza a importância do “aprender fazendo” e do “aprender a aprender”, permitindo que o aluno seja o protagonista de sua própria aprendizagem, construindo seu próprio conhecimento e compreendendo todo o processo de construção.

Para Papert, a teoria do construcionismo tem como princípio central que as pessoas constroem conhecimento de forma mais eficaz quando estão envolvidas ativamente na criação de coisas no mundo real. Em contraste, o instrucionismo, que ele compara ao construcionismo, refere-se à simples transmissão de informações, sem o engajamento ativo do aluno no processo de aprendizagem.

A Tabela 1 apresenta as principais características de cada conceito. O construcionismo destaca-se por colocar o aluno como o centro do processo educacional, estimulando sua criatividade, autonomia e capacidade de resolver problemas. Por outro lado, o instrucionismo enfatiza a transmissão de conhecimento de forma mais passiva, com o professor desempenhando um papel central na entrega das informações aos alunos.

Tabela 1. Relações possíveis no processo da aprendizagem

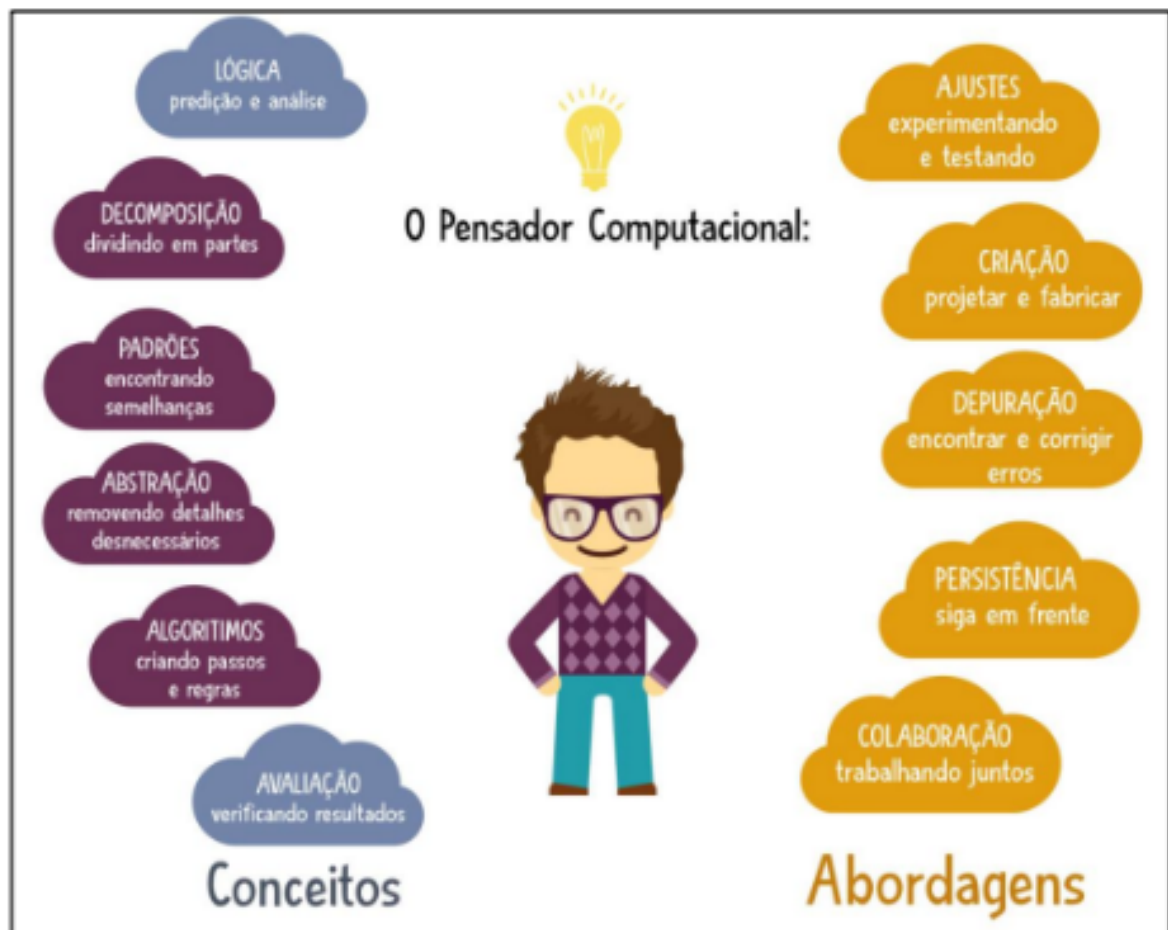
	INSTRUCIONISMO	CONSTRUCIONISMO
Conceito de Aprendizagem	Aprender por meio da reprodução de conteúdos recebidos de forma pré-estabelecida.	Aprender é pensar diferente de antes, é ver o mundo de outra forma, sob outros aspectos. Supõe a existência de várias alfabetizações.
Foco	Computador utilizado como máquina de ensinar.	Inserção do computador no espaço da sala de aula.
Elemento	Capacidade de reprodução do método tradicional de ensino.	Inserção do aprendente com o computador.
Processo	Conhecimento é transmitido por meio de perguntas e respostas.	Construir conhecimento quando entendem suas experiências.
Contexto	Espaço fechado, pré-programado com conteúdos previamente definidos.	Espaço para descobertas a partir das experiências; motivação ao diálogo.
Papel do sujeito	Receptor.	Realizar uma leitura do mundo e resolver problemas.

Fonte: adaptado de Silva, Kahlil e Nicot (2015, p. 15)

O Pensamento Computacional é uma abordagem que visa resolver problemas por

meio da decomposição em elementos menores e da busca por algoritmos que possam resolvê-los. Para isso, utiliza-se uma série de procedimentos, incluindo a decomposição, o reconhecimento de padrões, a abstração e o desenho algorítmico, como ilustrado na Figura 7. O PC está organizado em quatro pilares que são fundamentais para abordar problemas complexos de maneira organizada e eficaz. Ao aplicar essas estratégias, os indivíduos desenvolvem habilidades essenciais para a resolução de problemas não apenas na computação, mas também em diversas áreas do conhecimento e da vida cotidiana. O Pensamento Computacional é uma habilidade valiosa no mundo moderno, pois capacita as pessoas a enfrentar desafios de forma criativa, lógica e estratégica.

Figura 7. O pensador computacional



Fonte: Kampff et al. (2016, p. 3)

A decomposição é um processo de simplificação de um problema complexo, no qual ele é dividido em subproblemas menores e mais gerenciáveis. Isso torna o processo de resolução mais acessível e facilita a abordagem dos diferentes aspectos do problema separadamente.

O reconhecimento de padrões ocorre ao identificar e agrupar variáveis e dimensões do problema em diferentes categorias. Ao encontrar padrões, é possível perceber regularidades

e relações entre os elementos do problema, o que pode fornecer *insights* valiosos para a solução.

A abstração é uma etapa importante do Pensamento Computacional, pois consiste em ignorar informações desnecessárias e destacar apenas os elementos essenciais para a solução do problema. Isso permite que os aspectos mais relevantes sejam enfocados, tornando o processo de resolução mais eficiente e conciso.

Por fim, o desenho algorítmico é a etapa em que se escreve um processo eficiente para resolver o problema com o menor número de etapas possível. Isso envolve a criação de sequências lógicas e ordenadas de instruções, que guiarão a resolução do problema de forma sistemática e estruturada.

Em seu trabalho, Júnior (2023, p. 31) ao abordar o tema do Pensamento Computacional com a Educação Online no Ensino Médio, apresenta os conceitos dos quatro pilares do Pensamento Computacional em aplicação em uma atividade cotidiana, ultrapassando as fronteiras da computação. O autor narra a história de Joãozinho e a aplicação dos pilares do Pensamento Computacional na resolução do problema “fazer um bolo de aniversário para sua namorada”.

Nesse contexto, o autor destaca como a decomposição permitiu que Joãozinho dividisse o problema complexo em etapas menores e mais gerenciáveis. Ele desmembrou a tarefa de fazer o bolo em várias partes, como preparar a massa, o recheio e a cobertura, o que tornou a tarefa mais fácil de ser realizada.

Em seguida, o reconhecimento de padrões ajudou Joãozinho a identificar as variáveis importantes para a preparação do bolo. Ele observou que determinadas medidas e ingredientes se repetiam em cada etapa, o que o levou a agrupar esses elementos em categorias, facilitando o planejamento e organização da atividade.

A abstração permitiu que Joãozinho focasse apenas nos elementos essenciais do problema. Ele ignorou informações irrelevantes e concentrou-se nos passos fundamentais para a preparação do bolo, tornando o processo mais simples e eficiente.

Por fim, o desenho algorítmico representou a etapa em que Joãozinho elaborou um conjunto de instruções sequenciais para seguir durante o processo de preparo do bolo. Ele escreveu uma lista detalhada das ações necessárias em cada etapa, tornando o processo claro e orientado para a solução do problema.

Ao aplicar os pilares do Pensamento Computacional nessa situação cotidiana, Joãozinho conseguiu resolver o desafio de forma organizada, criativa e eficiente. Essa abordagem demonstra a versatilidade e a utilidade do Pensamento Computacional para solucionar problemas não apenas na computação, mas também em diversas outras áreas da vida diária. A narrativa de Gilson ressalta a importância de incorporar esses conceitos na educação, capacitando os estudantes a serem solucionadores de problemas proativos e

habilitados, independentemente do campo em que atuem.

Embora seja uma abordagem recente, o Pensamento Computacional tem sido reconhecido como um meio importante para disseminar a computação e o desenvolvimento do raciocínio lógico-computacional nas escolas, com o objetivo de resolver problemas e criar soluções. No entanto, percebeu-se que o Pensamento Computacional por si só não é suficiente para atender às diversas competências exigidas dos estudantes, como a habilidade de relacionar conceitos, categorias e valores de diferentes áreas de conhecimento, ou seja, habilidades interdisciplinares.

Desenvolver competências e habilidades relacionadas à Computação tem sido um desafio significativo no contexto educacional, especialmente porque a Computação é vista e tratada como uma ciência e não apenas como um método ou recurso tecnológico. Nesse sentido, surge um novo conceito na área da Ciência da Computação: o Raciocínio Computacional (RC), que pode contribuir não apenas para a programação ou para o uso de tecnologias digitais, mas também para outras áreas do conhecimento.

O Raciocínio Computacional (RC) é um conjunto de habilidades que fortalece o desenvolvimento de importantes competências humanas, como a tomada de decisões, criatividade, lógica, criticidade e colaboração. Essas habilidades são essenciais para a análise de situações, resolução de problemas e análise de dados pertencentes a diferentes categorias e complexidades. O RC é uma variação do Pensamento Computacional (PC), mas com um foco mais direcionado ao cotidiano dos estudantes.

E qual a diferença entre os termos? Em um estudo apresentado por Ferreira et al. (2015), os autores destacam que o termo “pensamento” se refere a uma atividade da mente na qual os objetos são tematizados ou decisões são tomadas em relação à realização de uma ação, enquanto “raciocínio” se refere a uma atividade do pensamento que envolve o encadeamento de juízos com o objetivo de estabelecer a verdade ou a falsidade de algo.

Pode-se afirmar que o Pensamento Computacional busca utilizar ferramentas, competências e habilidades da Computação para a resolução de problemas, enquanto o Raciocínio Computacional amplia o uso desses conceitos para o desenvolvimento de habilidades múltiplas. O objetivo não é apenas resolver problemas, mas também desenvolver e descobrir novas habilidades e competências que possam auxiliar o indivíduo em sua vida em sociedade.

Em 2017, foi lançada a versão final da Base Nacional Comum Curricular (BNCC) e, diferentemente do que ocorre em outros países, não houve a inclusão de uma disciplina específica referente ao Pensamento Computacional. Em vez disso, essa habilidade foi incluída de maneira difusa em diversas disciplinas já existentes. Insatisfeita com o conteúdo da BNCC, a Sociedade Brasileira de Computação (SBC) lançou um manifesto discordando da prática com tecnologia na educação. De acordo com a SBC, não foram

incluídas competências ou habilidades específicas que são desenvolvidas pelo Pensamento Computacional. Embora o currículo escolar já contemple o desenvolvimento de capacidades para a resolução de problemas cotidianos, a inclusão do Raciocínio Computacional na educação básica brasileira ainda está em discussão inicial.

Existem pesquisas que tratam da inclusão do Raciocínio Computacional no currículo educacional, mostrando que ele pode ser útil para o desenvolvimento cognitivo dos alunos. O Raciocínio Computacional pode contribuir com novas estratégias e metodologias para a prática pedagógica, permitindo que os alunos deem maior significado ao conteúdo estudado, já que o Raciocínio Computacional visa resolver problemas cotidianos de maneira prática, rápida e segura. Com a inclusão do Raciocínio Computacional na educação básica, é possível que a fronteira entre a escola e a sociedade seja reduzida por meio da integração das habilidades do Raciocínio Computacional com as disciplinas curriculares.

Ao refletir sobre o tema do Raciocínio Computacional, sua comprovada importância para todas as pessoas na sociedade e sua ausência explícita na estrutura curricular nacional, questiono como os alunos que ingressam no ensino médio enfrentam os desafios apresentados pela área de Computação. Especialmente os estudantes interessados na subárea de programação que precisam adquirir conhecimentos e habilidades específicas para programar um computador, que são diferentes da forma “natural” de pensar.

É comum que cursos e professores de programação não se preocupem ou não se sensibilizem com a ausência ou baixa capacidade dos alunos em responder às habilidades do Raciocínio Computacional, lecionando de forma tradicional. Isso resulta em enormes dificuldades no aprendizado de programação por parte dos estudantes, levando à reprovação e ao abandono. Esse cenário é preocupante. Compreender a importância do Raciocínio Computacional no cotidiano dos alunos em sala de aula é o passo inicial para elaboração desta pesquisa.

Para finalizar a discussão sobre a diferença entre os conceitos de PC e RC, apresento algumas iniciativas que fortalecem as reflexões discutidas nesta seção e demonstram a importância da adoção do RC como apoio às práticas de ensino.

O primeiro estudo, conduzido por Matos et al. (2021), relata as experiências vivenciadas em um curso de formação continuada de professores para a apropriação do Raciocínio Computacional. O resultado demonstra que os professores participantes do curso reconhecem a importância do Raciocínio Computacional no cotidiano escolar, auxiliando os estudantes na resolução de problemas tanto pessoais quanto profissionais. Esse estudo evidencia a necessidade de atualização das práticas de ensino dos docentes, que devem incorporar metodologias contemporâneas em suas disciplinas para que os alunos possam desenvolver habilidades relevantes para a vida em sociedade.

O estudo proposto por Silva et al. (2022) apresenta um relato de experiência sobre

como o uso do Raciocínio Computacional e de um jogo de tabuleiro contribuíram para o desenvolvimento de habilidades diretas e transversais do RC, bem como de habilidades da BNCC com alunos do 9º ano do Ensino Fundamental de uma escola pública.

2.3 Linguagem e Língua de Programação

É comum que profissionais e docentes da área de computação utilizem o termo “linguagem” para se referirem às línguas de programação existentes, tais como Java, PHP e Python. No passado, eu também utilizava essa terminologia em minhas aulas e conversas com colegas da área. No entanto, durante uma das reuniões de orientação, expus meu ponto de vista, o que nos levou a discutir sobre esse equívoco comum em nossa área e como a pesquisa poderia incentivar reflexões sobre o tema. Antes de abordar a relação entre linguagem e língua, foi necessário compreender o campo da semiótica, que trata das diferentes formas de linguagem.

Ao analisarmos os conceitos de linguagem e língua, podemos entender que linguagem é um sistema de signos que permite a comunicação entre as pessoas. Através da linguagem, elas expressam ideias, sentimentos e interações sociais. Já a língua é um sistema de signos convencionais e arbitrários utilizado por um grupo de pessoas para se comunicar. A língua é composta por elementos como palavras, expressões, regras gramaticais e sintáticas, e é transmitida de geração em geração por meio da educação e do convívio social. No mundo existem diversas línguas, cada uma com suas particularidades e variações regionais.

São exemplos de linguagem: A linguagem verbal, que utiliza palavras faladas ou escritas; a linguagem não-verbal, que se comunica através de gestos, expressões faciais, sons, imagens, cores, entre outros, como a linguagem corporal, a música, a pintura, etc. São exemplos de línguas: O português, o inglês, o espanhol, etc.

Signos são elementos que representam alguma coisa ou ideia e são utilizados para transmitir uma mensagem ou informação. Eles podem ser visuais, como ícones e símbolos, ou verbais, como palavras e frases. Os signos podem ser convencionais, ou seja, estabelecidos pela sociedade, como a língua e a escrita, ou podem ser criados de forma individual, como gestos e expressões faciais. Em resumo, os signos são elementos fundamentais para a comunicação e a expressão de ideias e informações entre os seres humanos.

A semiótica é uma ciência que estuda os signos, tanto verbais quanto não verbais, que permitem a comunicação entre as pessoas, tornando-se um campo de estudo amplo e abrangente. Os signos estão presentes em todos os lugares e estão relacionados tanto aos elementos naturais quanto culturais, como uma cor, um gesto, uma palavra, um ruído, um cheiro, uma carta, um evento, uma música ou um olhar. Dependendo do contexto em que estão inseridos, esses signos são associados a um significado específico. A origem do termo “semiótica” é atribuída ao filósofo Charles Sanders Peirce, um importante pensador que

contribuiu significativamente para o desenvolvimento dessa ciência (BELLUCCI, 2019).

A semiótica pode ser dividida em três partes: i) a **sintaxe**, que é a relação entre signos; ii) **semântica**, que é a relação entre o signo e o que ele representa; e iii) **pragmática**, que é a relação entre os signos e seus intérpretes.

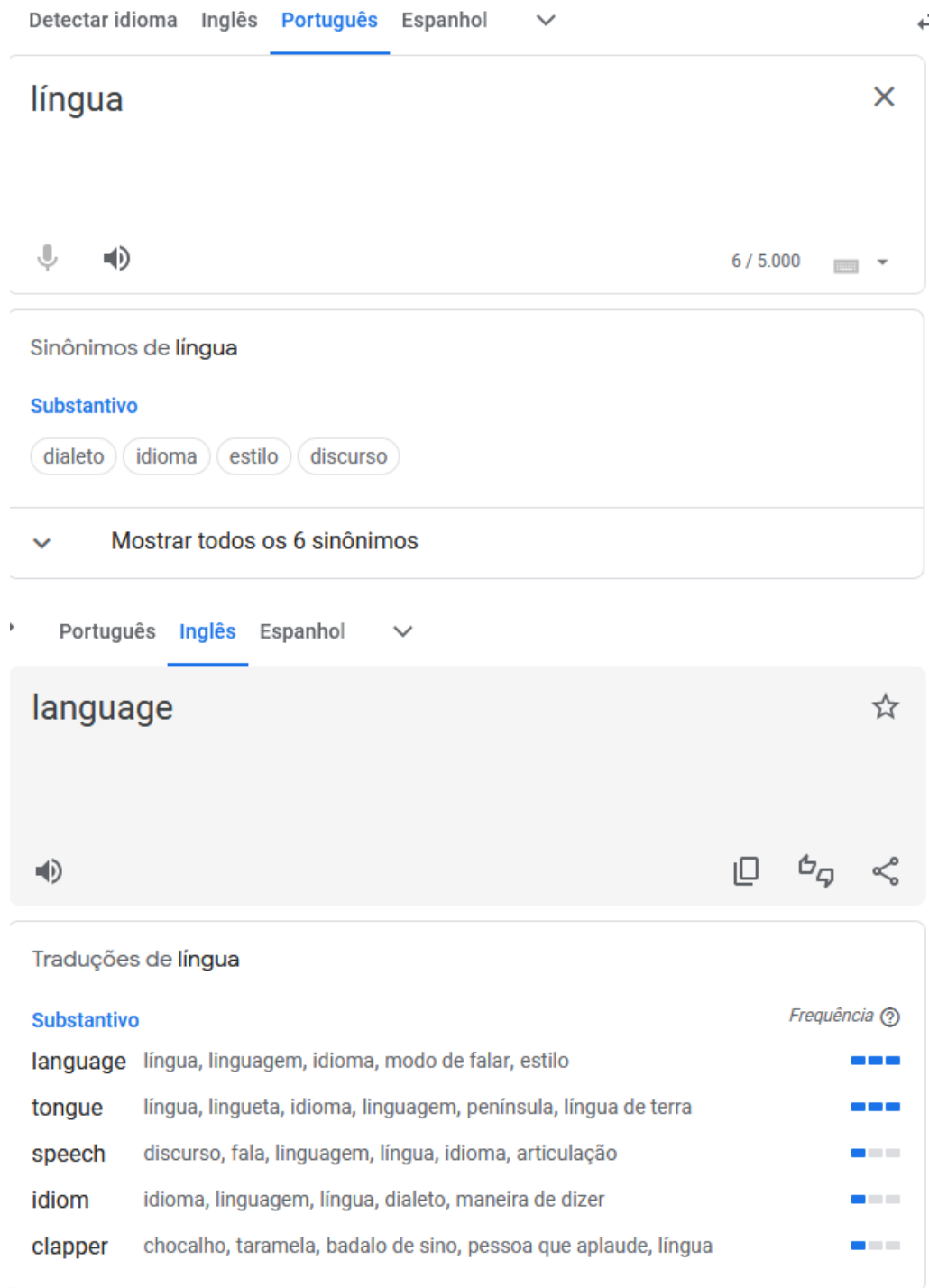
Através da semiótica, é possível investigar e interpretar como os signos são utilizados em diferentes contextos e culturas, revelando como são construídos e como adquirem significados específicos. Isso permite uma compreensão mais profunda da comunicação humana, contribuindo para o aprimoramento dos processos de interação e expressão.

Além de sua importância para a comunicação, a semiótica também é relevante para a compreensão do comportamento humano e da organização da sociedade. As linguagens e os sistemas de signos são fundamentais para a construção da cultura, das normas sociais e da identidade de um grupo, influenciando a forma como os indivíduos se relacionam e se expressam dentro de uma comunidade.

Um ponto relevante para compreender a confusão conceitual dos termos é abordado por Santaella (2022, p. 78), no contexto da tese linguística-cultural. Segundo a autora, a palavra inglesa “*language*” foi traduzida para o português como “linguagem”. No entanto, é importante considerar que, em inglês, o termo “*language*” é utilizado para se referir à “língua”. Ao contrário das línguas latinas, o inglês não possui duas palavras distintas para “língua” e “linguagem”. A confusão começou quando o termo “Java *language*” foi traduzido para “linguagem Java”, quando o correto seria “língua Java”. Essa tradução inadequada gerou um equívoco na interpretação dos termos e pode ter contribuído para a confusão entre os conceitos de “linguagem de programação” e “língua de programação” na área da computação.

A Figura 8, extraída da plataforma Google Tradutor, ilustra o problema mencionado. Ao solicitar a tradução da palavra “língua” do português para o inglês, podemos observar que a palavra *language* é a mais comumente utilizada, como evidenciado pelas barras de status na cor azul. Em seguida, podemos perceber que a segunda palavra mais utilizada é *tongue*. No português, a palavra “língua” possui uma sobrecarga semântica, uma vez que pode significar tanto um idioma (*language*) quanto o órgão (*tongue*). No entanto, em inglês, a palavra *tongue* não se refere a um idioma.

Figura 8. Tradução de língua na plataforma Google Translator



Fonte: elaborada pelo autor

Vale ressaltar também na reflexão que a plataforma de Inteligência Artificial ChatGPT³ foi utilizada para consultar trechos da música “Língua” de Caetano Veloso⁴. Foram feitas pesquisas na plataforma para obter informações relevantes sobre o tema em questão. Foi consultado no *prompt* de comando:

³ <https://openai.com/product/chatgpt>

⁴ <https://www.letras.mus.br/caetano-veloso/44738/>

O que Caetano Veloso quis dizer com as frases da música “Língua”:

“E deixe os Portugais morrerem à míngua

Minha pátria é minha língua”?

A resposta do ChatGPT foi a seguinte:

A música “Língua” de Caetano Veloso é uma reflexão sobre a importância e o poder da língua portuguesa para a identidade e cultura brasileira. A análise de cada uma das linhas solicitadas pode ter diferentes interpretações, dada a natureza subjetiva da arte, mas aqui está uma possível interpretação:

“E deixe os Portugais morrerem à míngua”: Esta linha pode ser interpretada como uma provocação lúdica, uma maneira de afirmar a autonomia e singularidade do português falado no Brasil, que se desenvolveu e evoluiu de forma distinta do português de Portugal. Poderia sugerir que, mesmo se o português em Portugal fosse desaparecer (“morrer à míngua”), o português do Brasil continuaria forte e vibrante.

“Minha pátria é minha língua”: Aqui, Caetano parece redefinir o conceito de pátria. Ao invés de associá-la a um território geográfico, ele a associa à língua portuguesa. Isso pode expressar a ideia de que a identidade e a cultura brasileira são intrinsecamente ligadas à língua que falamos, e que é através da língua que expressamos e experimentamos nosso senso de pertencimento e nacionalidade.

A interpretação do ChatGPT é uma das várias possíveis, e é importante ressaltar que a poesia e a música são formas de expressão artística que permitem diferentes camadas de significado. Cada pessoa pode extrair suas próprias interpretações e experiências pessoais ao apreciar uma obra, tornando a experiência única e enriquecedora. Ainda na plataforma, a ferramenta produziu o seguinte texto sobre a diferença entre idioma e língua:

Os termos “língua” e “idioma” são frequentemente usados intercambiavelmente, mas, dependendo do contexto, podem ter nuances distintas.

Língua: O termo “língua” é geralmente usado para se referir a um sistema completo de comunicação falada, escrita ou sinalizada usado por uma comunidade específica. Por exemplo, o português, inglês, espanhol, chinês, entre outros, são todos línguas. Uma língua tem sua própria gramática, vocabulário e sintaxe.

Idioma: O termo “idioma” também pode ser usado para se referir a um sistema completo de comunicação, como descrito acima. No entanto, em alguns contextos, “idioma” pode ser usado para se referir a uma variedade ou dialeto

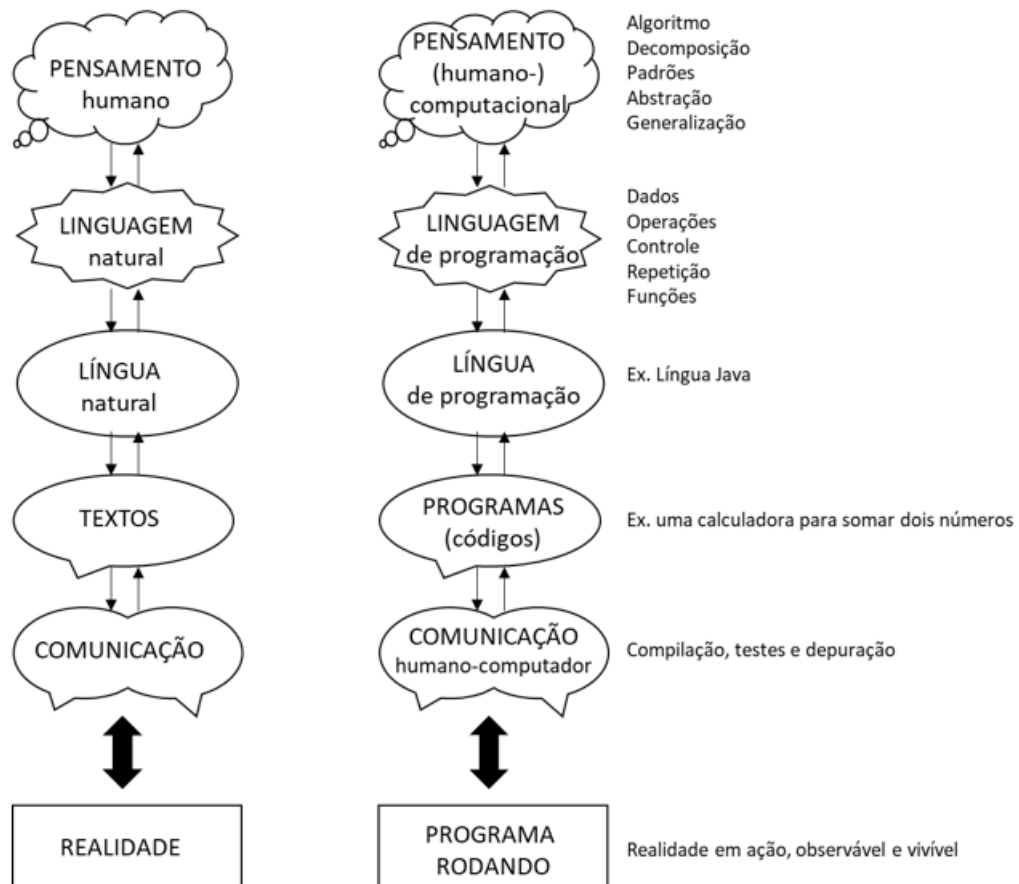
de uma língua. Por exemplo, as diversas formas de espanhol faladas na Espanha, México, Argentina, etc., podem ser referidas como diferentes idiomas do espanhol.

Na prática, no entanto, “língua” e “idioma” são frequentemente utilizados de forma intercambiável, e a distinção entre eles pode ser sutil e variar dependendo da região ou do campo de estudo.

Por exemplo, em português do Brasil, é mais comum utilizar o termo “língua” para se referir a sistemas de comunicação como português, inglês, espanhol, entre outros. Em contraste, em espanhol, é mais comum utilizar o termo “idioma” (“idioma” em espanhol) para se referir a esses mesmos sistemas. No entanto, essa distinção pode variar de região para região e de contexto para contexto.

No contexto da Computação, ao refletir sobre a diferença entre linguagem e língua, podemos considerar que Java, PHP e Python são línguas, e não linguagens, pois possuem uma sintaxe definida e utilizam palavras reservadas, assim como acontece com as línguas naturais, como o português, inglês e espanhol. Essas línguas de programação foram criadas pelos seres humanos com o objetivo de estabelecer uma comunicação entre o homem e o computador. Portanto, se utilizamos de forma equivocada o termo “linguagem” para nos referirmos a essas línguas, como podemos definir a linguagem de programação? Podemos compreender a linguagem de programação como a estrutura que define e caracteriza todas as línguas de programação existentes. Na Figura 9, podemos observar um paralelo entre o esquema filosófico que se inicia no pensamento humano e finaliza na realidade, e uma estrutura similar presente na computação.

Figura 9. Do pensamento a realidade na Filosofia x Computação



Fonte: elaborado pelo autor

Analisando a Figura 9, podemos estabelecer uma analogia entre a linguagem humana e a linguagem de programação. Assim, como a linguagem humana engloba a linguagem natural e a linguagem de sinais, a linguagem de programação possui estruturas comuns presentes em todas as línguas de programação, como dados, operações, estruturas de controle, estruturas de repetição e funções.

Ao avançarmos para o nível da língua, no contexto humano, podemos citar o português, inglês e espanhol como exemplos de linguagem natural, enquanto a Libras e a American Sign Language (ASL) são exemplos de linguagem de sinais. Já para a linguagem de programação, Java, PHP, Python, entre outras, são exemplos de línguas.

Ao analisarmos o próximo nível da figura, percebemos que a língua portuguesa, por exemplo, é composta por textos que seguem uma sintaxe definida e regulamentada. Da mesma forma, os “textos” codificados em uma língua de programação, como o Java, devem seguir as regras de sintaxe estabelecidas para aquela língua, resultando em códigos escritos. Esses códigos são avaliados quanto à sua estrutura e sintaxe estabelecidas pela língua, permitindo a comunicação entre o ser humano e o computador através da avaliação e compilação do código.

A última etapa proposta pela Filosofia é a realidade, o que cada indivíduo vivencia em seu dia a dia. As ações de vivência em sociedade são resultados de todo o processo experienciado pelo ser humano. Na computação, a experiência com a realidade pode ser observada com o sistema em pleno funcionamento, executando as funcionalidades aos quais foi proposto e proporcionando ao usuário a possibilidade de interação com o sistema.

Portanto, é importante que todos os docentes e profissionais da computação reflitam sobre esses conceitos e, caso percebam a existência de equívocos na definição das nomenclaturas utilizadas, sejam capazes de disseminar essas informações para seus alunos e demais colegas de profissão. Dessa forma, talvez seja possível, em um futuro não muito distante, romper com esse paradigma conceitual que tem prevalecido na área da computação por muito tempo.

2.4 Ensino e Aprendizagem de Programação

O elevado índice de insucesso nas disciplinas introdutórias de programação (reprovação e evasão) ao longo dos anos tem sido um desafio para a comunidade educacional, impulsionando pesquisas que buscam contribuir para a solução desse problema por meio da proposição de metodologias ou técnicas mais eficazes. Esta seção tem como objetivo compreender como o cenário nacional se apresenta nesse contexto, considerando os últimos anos.

Para isso, foram utilizados como referência dois estudos relevantes: o primeiro, apresentado por Holanda, Freire e Coutinho (2019), que busca compreender o panorama de pesquisa das publicações no Brasil sobre as estratégias adotadas no ensino de programação introdutória em cursos superiores, entre os anos de 2014 e 2018; o segundo, elaborado por Ribeiro, Martins e Berssanette (2022), que apresenta um panorama brasileiro do ensino e aprendizagem de programação de computadores na educação básica. Ambos os estudos realizaram uma revisão da literatura no que se refere ao ensino e aprendizagem de programação.

Os resultados apontados por esses estudos contribuíram para a escolha das técnicas e metodologias utilizadas na proposta de solução da pesquisa em questão, permitindo que sejam embasadas em evidências e experiências já estudadas e discutidas na área.

É notável que as instituições de ensino superior no Brasil têm adotado como principais metodologias para o ensino de programação a ludicidade e a gamificação, representando 35% do total de artigos identificados nas pesquisas. Além disso, destaca-se a utilização da aprendizagem baseada em projetos (PBL) e de linguagens e tecnologias específicas, como Scratch, Logo e Portugol.

Ao analisar o cenário da educação básica, percebe-se um aumento significativo no

uso de tecnologias de apoio, abrangendo um total de 74% das pesquisas. As tecnologias mais destacadas no ensino básico são o Scratch, o MIT App Inventor e o Code.org, que demonstram ser ferramentas importantes no estímulo ao aprendizado de programação desde as séries iniciais.

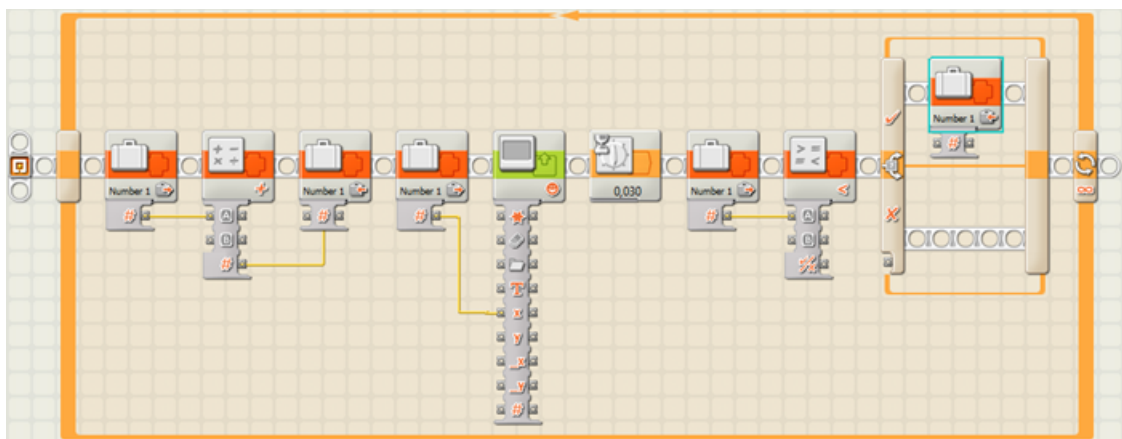
Em seguida, destacam-se as técnicas não tradicionais de ensino, que representam 32% dos resultados. Uma das técnicas não tradicionais em destaque é o uso da computação desplugada, que permite aos estudantes aprenderem conceitos computacionais sem a necessidade de computadores, promovendo uma abordagem mais interativa e criativa.

A utilização de tecnologias que possibilitam a programação com blocos tem se mostrado uma tendência crescente no ensino de programação. O objetivo dessas tecnologias é simplificar o contato dos estudantes com o universo da programação, tornando-o mais acessível e facilitando o processo de aprendizado. Por serem mais simples e intuitivas, as tecnologias de programação com blocos permitem que os estudantes pratiquem os conceitos de linguagem de programação de forma lúdica, estimulando sua criatividade.

Essa abordagem não se restringe apenas ao ensino básico, mas também é uma realidade no ensino superior, com estudos indicando que a programação de blocos está gradativamente substituindo a codificação tradicional. A adoção dessas tecnologias no contexto acadêmico possibilita que os estudantes desenvolvam habilidades essenciais de programação, ao mesmo tempo em que se familiarizam com conceitos e técnicas mais avançadas.

A Figura 10 ilustra um exemplo de como a programação com blocos pode ser utilizada na construção de aplicações de robótica pelos estudantes.

Figura 10. Ilustração de programação de blocos em Robótica



Fonte: Robótica – Movimento Mindstorms⁵

Potenciais metodologias não tradicionais para o ensino de programação incluem a computação desplugada e a utilização de jogos. A computação desplugada consiste

⁵ <https://imasters.com.br/tecnologia/robotica-movimento-mindstorms>

no aprendizado de programação por meio de atividades que não envolvem o uso de um computador, visando consolidar conceitos iniciais em ambientes familiares aos alunos. Nesse contexto, os estudantes podem aprender a lógica de programação e desenvolver habilidades de resolução de problemas utilizando materiais físicos, como cartas, quebra-cabeças ou tabuleiros, sem a necessidade de codificação em um computador. Essa abordagem proporciona uma experiência mais tangível e interativa, permitindo que os alunos pratiquem os conceitos de programação de forma concreta antes de aplicá-los em um ambiente digital.

Já o uso de jogos no ensino de programação pode criar um ambiente mais atraente e motivador para o aprendizado. Os jogos proporcionam desafios e recompensas, o que estimula o engajamento e a persistência dos estudantes. Além disso, os jogos permitem que os alunos aprendam de forma mais descontraída e prazerosa, o que pode reduzir o estresse e a ansiedade associados ao aprendizado de conceitos técnicos. Através de jogos educacionais, os estudantes podem resolver problemas, criar projetos e experimentar diferentes soluções, desenvolvendo habilidades práticas de programação de maneira divertida e criativa.

Um estudo recente realizado por Ribeiro, Martins e Berssanette (2022) identificou que o ensino de programação pode trazer diversos benefícios para os estudantes, aumentando o interesse e a motivação pela disciplina. Além disso, o aprendizado de programação pode contribuir para o desenvolvimento de habilidades importantes, como raciocínio lógico, criatividade, resolução de problemas, senso crítico, fluência digital e comunicação. Esses resultados positivos foram observados em aproximadamente 84% das evidências analisadas no estudo.

Os resultados apresentados no cenário nacional indicam uma busca constante por soluções que possam auxiliar os docentes nas práticas de ensino e aprendizagem em disciplinas introdutórias de programação. Essa é uma temática atual e de grande relevância, que vem sendo objeto de estudo por inúmeras instituições brasileiras.

2.5 Alfabetização e Letramento em Programação

Esta seção visa discutir a aplicação de conceitos da educação, alfabetização e letramento no contexto da computação, especialmente na programação de computadores. É importante destacar a diferença entre esses dois termos: a alfabetização refere-se ao processo de aprendizagem no qual um indivíduo adquire a habilidade de ler e escrever, enquanto o letramento envolve o desenvolvimento do uso competente da leitura e escrita em práticas sociais.

Dessa forma, a principal distinção entre alfabetização e letramento está relacionada à qualidade da habilidade do indivíduo em dominar a leitura e a escrita. Uma pessoa alfabetizada consegue compreender e produzir o sistema de escrita, enquanto um sujeito letrado vai além, demonstrando domínio da língua em seu cotidiano e em diversos contextos

sociais.

A Figura 11 ilustra o processo de alfabetização e letramento direcionado para a educação infantil. Essa etapa é fundamental para o desenvolvimento das competências de leitura e escrita nas crianças, e pode ser comparada com o processo de aquisição dos conceitos básicos na programação de computadores. Assim como a alfabetização é o ponto de partida para o letramento na língua, a compreensão dos conceitos fundamentais da programação é essencial para que os estudantes possam avançar em suas habilidades nessa área.

Figura 11. O processo de alfabetização e letramento



Fonte: Facebook⁶

A abordagem de Magda Soares (2004), uma referência nacional em alfabetização e letramento, é de extrema relevância para compreender a diferença entre esses dois conceitos. A pesquisadora, professora titular emérita da Faculdade de Educação da Universidade Federal de Minas Gerais (UFMG) e fundadora do Centro de Alfabetização, Leitura e Escrita (Ceale) na mesma instituição, destaca que a distinção fundamental entre alfabetização e letramento reside no grau de domínio que o sujeito possui sobre a leitura e escrita.

⁶ <https://web.facebook.com/rhemaeducacao/photos/a.109458775880683/1837654423061101/>

No contexto da alfabetização, o indivíduo aprende a ler e escrever, adquirindo as habilidades básicas de compreender e produzir o sistema de escrita. Entretanto, mesmo sendo capaz de decodificar palavras e textos, pode não estar habituado a utilizar essas habilidades em diferentes práticas sociais cotidianas.

Já no letramento, o sujeito vai além das habilidades básicas e demonstra um domínio mais completo da leitura e escrita em diversas situações e contextos sociais. Isso significa que o indivíduo letrado é capaz de utilizar a linguagem escrita de forma competente e eficaz em diferentes atividades e interações sociais, desenvolvendo uma compreensão mais ampla e profunda dos significados e usos da linguagem escrita.

Com base nas ideias defendidas por Magda Soares, podemos organizar as informações sobre alfabetização e letramento em relação a conceito, uso, indivíduo, atividades envolvidas e ensino, como ilustrado na Tabela 2.

Tabela 2. Alfabetização x Letramento

	ALFABETIZAÇÃO	LETRAMENTO
CONCEITO	Alfabetização é o processo de aprendizado da leitura e da escrita.	Letramento é o desenvolvimento do uso competente da leitura e escrita nas práticas sociais.
USO	Uso individual da leitura e escrita.	Uso social da leitura e escrita.
INDIVÍDUO	Alfabetizado é o sujeito que sabe ler e escrever.	Uma pessoa letrada sabe usar a leitura e a escrita de acordo com as demandas sociais.
ATIVIDADES ENVOLVIDAS	Codificar e decodificar a escrita e os números.	Organizar discursos, interpretação e compreensão de textos, reflexão.
ENSINO	Deixa o indivíduo apto a desenvolver os mais diversos métodos de aprendizado da língua.	Habilita o sujeito a utilizar a escrita e a leitura nos mais diversos contextos.

Fonte: adaptado das ideias de Magda Soares

A alfabetização e o letramento são processos distintos, porém complementares e interligados. Enquanto a alfabetização se refere ao aprendizado das habilidades básicas de ler e escrever, o letramento vai além, abrangendo a compreensão e o uso competente da leitura e escrita em diversas situações sociais.

Magda Soares destaca a importância de trabalhar simultaneamente a alfabetização e o letramento nas escolas, pois um processo complementa o outro. É essencial que o indivíduo não apenas decodifique palavras, mas também compreenda o que está lendo e seja capaz de produzir textos coerentes e adequados para diferentes contextos comunicativos.

No Brasil, o conceito de letramento tem ganhado destaque progressivamente, representando os processos que levam as pessoas a dominar adequadamente a leitura e a escrita, transcendendo a mera alfabetização. Esse entendimento é refletido nas matérias veiculadas na mídia, nas quais ser alfabetizado é compreendido como mais do que apenas saber ler e escrever um bilhete, algo que, até algum tempo atrás, era considerado suficiente para retirar alguém da condição de analfabeto.

Ao considerarmos os conceitos de alfabetização e letramento na educação em computação, é fundamental ir além da simples decodificação da língua de programação. É necessário ensinar aos estudantes não apenas as habilidades básicas de programação, mas também a compreensão profunda da língua e o uso adequado da programação em diferentes contextos e situações sociais.

Isso implica em incentivar a criatividade, a resolução de problemas e o pensamento crítico dos estudantes, além de desenvolver a capacidade de se comunicar e colaborar com outros programadores. O ensino de programação, portanto, deve contemplar os conceitos de alfabetização e letramento para ajudar os estudantes a desenvolver um domínio completo da linguagem de programação e sua aplicação em diversas práticas sociais.

Os professores de programação frequentemente adotam uma abordagem de ensino que prioriza a leitura e escrita de códigos de programação em uma determinada língua, geralmente em um contexto específico voltado para a matemática. Essa prática permite que o estudante seja alfabetizado em uma determinada língua de programação, tornando-se capaz de ler e escrever códigos de programação. No entanto, é importante lembrar que a alfabetização é apenas um dos componentes do processo de letramento e que esses componentes se complementam. Portanto, é fundamental que os professores de programação fortaleçam a necessidade de adotar o processo de letramento em sua abordagem de ensino, permitindo aos estudantes a aplicação das habilidades de leitura e escrita de códigos em práticas sociais e culturais diversas.

A inclusão de práticas em diferentes contextos para o ensino de programação pode contribuir para o letramento do estudante. Levar em conta o contexto social e as experiências de vida de cada aluno pode ajudar a facilitar o processo de aprendizagem. A inclusão de situações do cotidiano, como problemas que podem ser resolvidos por meio da programação, pode estimular o aluno a se manter engajado com o processo.

As reflexões sobre alfabetização e letramento, provenientes do campo da Educação, são fundamentais para serem incorporadas e discutidas no campo da Computação (BIS-SONI; MORETTO, 2023). É comum que os professores que lecionam em nossa área não possuam formação em docência, limitando-se apenas à sua formação técnica e experiência profissional. Somado a esse fator, está a resistência à mudança que é percebida em muitos dos colegas docentes. Esses fatores podem contribuir para uma fraca ou inexistente discussão sobre esses temas na Computação, o que acarreta na utilização equivocada de termos e/ou práticas docentes insuficientes que prejudicam os estudantes durante seu processo de aprendizagem em programação.

2.6 Metodologias Ativas no Processo de Ensino e Aprendizagem

A evolução contínua dos processos de ensino e aprendizagem reflete a adaptação à realidade presente. Isso resulta na abrangência de várias tendências e abordagens educacionais ancoradas nas demandas da sociedade e das necessidades reconhecidas tanto nas salas de aula quanto na vida cotidiana. Consequentemente, os métodos de ensino tradicionais deixaram de ser adequados para a atualidade (SEBOLD et al., 2010).

Em contraste com o modelo tradicional de ensino, que se caracteriza pela centralização do docente como detentor exclusivo do conhecimento e das ferramentas de ensino, algumas instituições estão explorando abordagens que promovam o desenvolvimento das capacidades intelectuais dos estudantes. Isso é alcançado por meio de práticas reflexivas, as quais se conectam aos desafios e experiências do dia a dia, assim como ao contexto social no qual os alunos estão inseridos. Isso também abrange a capacidade de encontrar soluções para os obstáculos que surgem ao longo da jornada acadêmica e, posteriormente, profissional (VEIGA; CASTANHO; AMARAL, 2000; FONSECA et al., 2005).

Dentro deste cenário, os métodos de ensino e aprendizagem contemporâneos sustentam a visão de Paulo Freire, o renomado educador, que enfatizou em suas reflexões e princípios pedagógicos a importância crucial da interação humana nas relações entre educador e educando.

Com o objetivo de promover uma educação caracterizada por abordagens mais dialógicas, críticas, reflexivas e libertadoras, baseadas nos métodos de conscientização, desalienação e problematização, surge uma oposição à visão educacional tradicional, muitas vezes chamada de “bancária”, na qual os alunos ou indivíduos eram considerados meros “recipientes de conhecimento ou informações” (FREIRE, 2011). Nesse sentido, a proposta é identificar as necessidades, obstáculos e lacunas nos processos de ensino (BERBEL, 2019).

Dessa forma, emergem abordagens inovadoras de ensino e aprendizagem, como as Metodologias Ativas (MAs). Estas abordagens colaboram para que o estudante assuma o papel principal no processo de aprendizagem, participando ativamente ao buscar, refletir e determinar qual caminho é mais apropriado para atingir o objetivo estabelecido. Isso é realizado por meio de experiências concretas ou dinâmicas que promovem o desenvolvimento cognitivo, socioemocional, autonomia, proatividade, resolução de problemas e autoestima. Essas práticas se fundamentam nos princípios pedagógicos dinâmicos. Essas metodologias abrangem uma variedade de modelos e estratégias para serem incorporadas à prática educativa. Isso permite que elas se conectem à realidade profissional, resultando em uma ampla gama de vantagens e desafios nos diversos níveis educacionais (MEDEIROS; RIBEIRO; SOUSA, 2020; LEITE et al., 2021).

Uma pesquisa conduzida por (ANGELO et al., 2023) examinou e apresentou as principais conclusões relacionadas às Metodologias Ativas (MAs). Para alcançar esse

objetivo, o estudo conduziu uma revisão integrativa da literatura, abrangendo um total de 43 artigos. A análise revelou que essas metodologias são extensivamente adotadas nas Instituições de Ensino Superior, particularmente nos cursos ligados à área da saúde e nos programas de pós-graduação.

Contudo, é importante notar que as MAs também estão sendo aplicadas na Educação Básica, abrangendo desde a Educação Infantil até o Ensino Fundamental, o Ensino Médio e a Educação de Jovens e Adultos (EJA), além da modalidade de ensino Técnico. O propósito fundamental dessas abordagens é romper com o formato tradicional de ensino, fomentar o desenvolvimento de habilidades críticas nos alunos e reforçar a colaboração em equipe, enquanto integram teoria e prática de maneira coesa.

Esse processo de aprendizado está diretamente vinculado à realidade dos estudantes e à habilidade dos educadores de relacionar o ensino com o contexto atual dos alunos.

As Metodologias Ativas (MAs) representam abordagens pedagógicas que têm o aluno como foco central, priorizando a obtenção de conhecimentos e habilidades práticas. De maneira semelhante ao contexto do ensino superior, as instituições de ensino médio também possuem a responsabilidade de preparar os estudantes para uma interação eficaz com o mundo profissional. Isso inclui a garantia da formação de cidadãos reflexivos e críticos, capacitados para enfrentar os desafios do mercado de trabalho.

Com base nesses elementos vinculados à abordagem de aprendizagem ativa, é evidente que uma mudança está ocorrendo nos meios e métodos de ensino, facilitando a compreensão, assimilação e aprendizado dos alunos em diferentes níveis educacionais. Originando-se inicialmente no método de aprendizagem proposto por William Glasser, um psiquiatra americano que promove a teoria prática da educação, essa abordagem estabelece que os alunos absorvem aproximadamente 10% do conteúdo ao ler, 20% ao escrever, 50% ao observar e escutar, 70% ao debater com outros, 80% ao praticar e incríveis 95% ao ensinar ou compartilhar o conhecimento adquirido.

Essa perspectiva é reforçada por Segura e Kalhil (2015), que afirmam que a aprendizagem ativa viabiliza um ensino integrador, inclusivo, orientado pelo estudante e auto-direcionado. Dessa maneira, é claro que os métodos mais eficazes estão incorporados nas Metodologias Ativas.

A adoção das práticas de metodologias ativas, por meio de abordagens educacionais criativas, audaciosas, descentralizadas e, acima de tudo, ativas, terá um impacto positivo no progresso dos alunos, tornando-os conscientes da realidade global e habilitando-os a reconhecer o alcance de suas ações e como elas reverberam na sociedade. Adicionalmente, isso posicionará o professor como um facilitador nesse processo de aprendizado e construção de conhecimentos relevantes. Dessa forma, a capacidade de transformação será abrangente, envolvendo a instituição, o professor, o aluno e o próprio desenvolvimento educacional.

Para colocar em prática as Metodologias Ativas, é essencial conduzir uma análise organizacional e estrutural, além de qualificar os professores atuantes nas instituições educacionais. Contudo, existem etapas sequenciais que orientam a aplicação e execução das MAs, que são as seguintes: delinear o tópico de conhecimento; definir as habilidades e competências de aprendizagem a serem desenvolvidas; estipular a estratégia de aprendizagem ativa a ser empregada; utilizar recursos e ferramentas que despertem o interesse dos alunos; ajustar constantemente ao objeto de conhecimento, à realidade dos alunos, às habilidades e competências; criar atividades significativas e avaliar a aprendizagem (ANGELO et al., 2023).

Uma pesquisa conduzida por Ferreira e Braga (2023) demonstra de maneira inequívoca uma convergência das estratégias pedagógicas no contexto brasileiro. Isso aponta para uma associação destacada entre o Ensino de Programação e as Metodologias Ativas, bem como a abordagem da aprendizagem centrada no aluno, como é reconhecida internacionalmente. Ambas as práticas têm exercido uma influência recíproca e colaborado na criação de ambientes de aprendizagem promissores, promovendo avanços de maior relevância no campo educacional do Brasil.

3 PROPOSTA DE SOLUÇÃO: UMA NOVA FORMA DE ENSINAR PROGRAMAÇÃO

Esta seção tem como propósito fornecer detalhes sobre o procedimento de desenvolvimento conducente à concretização da solução delineada na pesquisa: a formulação de uma nova abordagem didática para o ensino de programação. Esse esforço culminou na transformação das minhas práticas docentes. Contudo, para alcançar esse resultado, foi fundamental trilhar um caminho extenso, o qual abrangeu uma reflexão profunda e uma autoavaliação acurada da minha experiência e prática enquanto educador.

Durante meus nove anos de experiência em sala de aula, mais da metade dos quais incluíram o ensino de programação, e ao longo de minha vida acadêmica como estudante, vivenciei uma abordagem tradicional de ensino de programação. Todos os professores que tive seguiram um modelo semelhante, iniciando com a teoria da programação, seguido pelos conceitos de fluxograma e, em seguida, introduzindo a linguagem de programação usando uma língua já conhecida, como Pascal ou C. Para um aluno sem experiência em programação, além da dificuldade em entender as estruturas da linguagem, como uma estrutura de repetição por exemplo, era necessário compreender a língua, tornando o aprendizado duplamente difícil para o aluno.

O ensino da disciplina geralmente se concentrava em trabalhar em soluções matemáticas diversas, como o cálculo de uma média ou do fatorial de um número. O professor elaborava listas de exercícios baseadas exclusivamente em problemas matemáticos e as passava para os alunos codificarem. As avaliações também eram baseadas em papel, onde o aluno escrevia manualmente o código em um intervalo de tempo limitado, o que era desconfortável e angustiante.

Quando comecei a lecionar, me questionei sobre como poderia fazer diferente. Como professor iniciante e inexperiente, parti do princípio de repetir o que tinha vivenciado como aluno, mas logo percebi que essa abordagem não era satisfatória. Busquei a opinião de colegas mais experientes e descobri que a maioria deles usava a mesma abordagem tradicional. Assim, iniciei a minha trajetória docente baseada em uma abordagem tradicional, mas com o compromisso de modificar este cenário ao longo do tempo.

No início, minhas aulas seguiram o mesmo padrão que eu havia vivenciado como aluno. Preparei uma apostila contendo o conteúdo da disciplina, com listas de exercícios baseados exclusivamente em problemas matemáticos. As avaliações eram realizadas através de provas impressas, e as aulas consistiam principalmente em exposições teóricas. Com o passar dos semestres, percebi a necessidade de aprimorar minhas práticas em sala de aula,

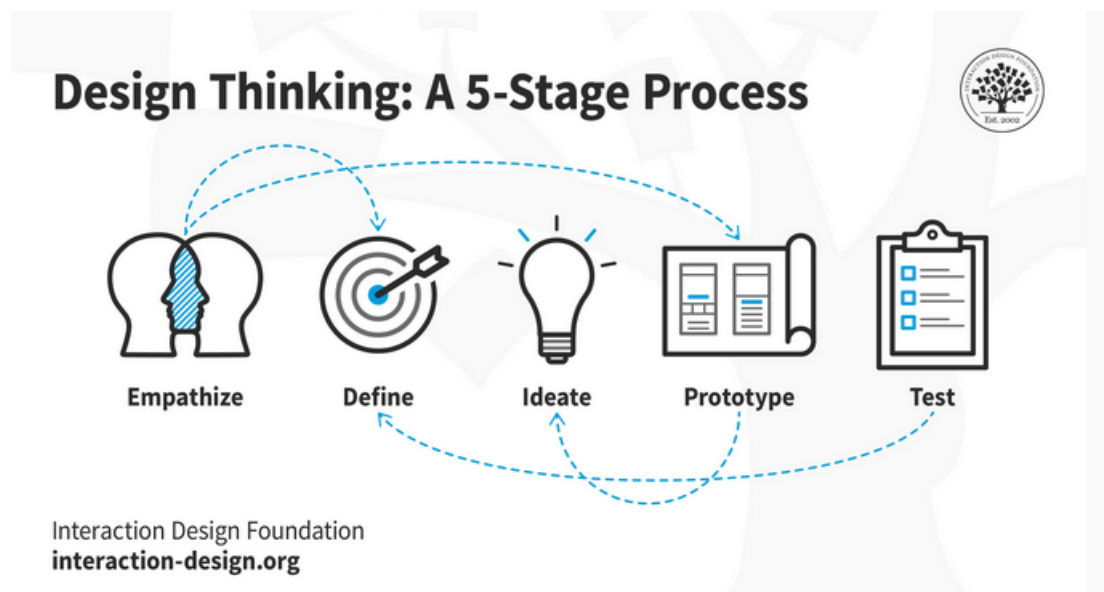
buscando suprir minhas dúvidas e desejos de uma maneira mais embasada teoricamente. Uma das primeiras mudanças que implementei foi substituir as provas impressas por provas práticas realizadas no computador. Posteriormente, abandonei completamente a aplicação de provas tradicionais e passei a adotar a prática de projetos como uma valiosa ferramenta de avaliação.

3.1 *Design thinking*

Com o propósito de adotar uma nova abordagem didática e aprimorar as práticas docentes, busquei embasamento na literatura para encontrar uma teoria que sustentasse esse processo. Minha escolha recaiu sobre o *Design thinking*, uma vez que sua proposta enfatiza a resolução de problemas e a geração de novas ideias, levando em consideração a empatia pelo usuário, a experimentação e a colaboração entre diferentes áreas de conhecimento. Essa abordagem é amplamente empregada em diversos setores, inclusive na educação, onde tem sido reconhecida como uma eficaz estratégia para desenvolver soluções inovadoras e centradas no aluno.

O processo do *design thinking* geralmente segue cinco etapas: empatia (compreender as necessidades do usuário), definição do problema (identificar claramente o desafio a ser solucionado), ideação (geração de ideias criativas para resolver o problema), prototipação (construção de protótipos para testar e aprimorar as soluções) e testes (avaliar a solução junto ao usuário final). Essas etapas formam um ciclo iterativo e podem ser adaptadas conforme necessário (OLIVEIRA, 2014). Na Figura 12, é possível visualizar a representação gráfica dessas cinco etapas.

Figura 12. Processo iterativo do *Design Thinking*



Fonte: What is Design Thinking?¹

Na área educacional, o *design thinking* pode ser empregado de diversas formas para criar soluções inovadoras e centradas no aluno. Algumas das maneiras de aplicar essa abordagem na educação incluem: identificar as necessidades dos alunos, definir claramente o problema a ser resolvido, gerar ideias criativas, realizar prototipagem, testar as soluções e, por fim, implementá-las. O uso do *design thinking* pode ser altamente efetivo para aprimorar o sistema educacional e desenvolver soluções inovadoras e personalizadas para as necessidades dos alunos (MOREIRA; SIMIÃO-FERREIRA; DE-CARVALHO, 2022).

As etapas iniciais do processo, conhecidas como empatia e definição do problema, têm como objetivo compreender as necessidades dos usuários (estudantes) e identificar o desafio a ser solucionado. Essas etapas foram compreendidas através das minhas observações e experiências diárias como professor e também como aluno em sala de aula. As informações referentes a essas etapas foram apresentadas na seção 1.3 da seção 1.

3.2 Etapa de Ideação

A elaboração da proposta de solução teve início no segundo semestre de 2021, quando iniciei o desenvolvimento do planejamento para a nova abordagem didática na disciplina LP1. A etapa de ideação, que consiste em gerar ideias criativas para solucionar o problema identificado, foi conduzida de forma estruturada. Inicialmente, busquei na literatura práticas, técnicas e metodologias ativas já conhecidas e disponíveis. Em seguida, elaborei uma matriz SWOT para compreender as variáveis internas e externas que poderiam impactar ou apoiar a solução. Por fim, um planejamento estratégico foi desenvolvido utilizando a ferramenta Canvas. A combinação desses esforços resultou no planejamento elaborado durante a etapa de prototipagem.

As metodologias ativas são estratégias de ensino que visam incentivar os estudantes a aprenderem de forma autônoma e participativa, através da resolução de problemas e situações reais, realizando tarefas que os estimulem a ir além, tomar iniciativa, debater e se tornarem responsáveis pela construção do conhecimento. Nesse modelo de ensino, o professor assume um papel de coadjuvante nos processos de ensino e aprendizagem, permitindo que os estudantes tenham protagonismo em seu próprio aprendizado.

Após avaliar as práticas, técnicas e metodologias disponíveis, bem como analisar os resultados de estudos apresentados na seção anterior, selecionei aquelas que se mostraram essenciais para orientar a pesquisa. O resultado dessa etapa foi uma lista composta por práticas, técnicas e metodologias que sustentam a fundamentação da solução proposta. Essa lista inclui as seguintes abordagens: aprendizagem colaborativa, programação em pares, aprendizagem por projeto, autoria, ludicidade, jogos digitais, concreto ao abstrato, letramento, pensamento criativo e design.

¹ <https://www.interaction-design.org/literature/topics/design-thinking>

3.2.1 Princípios Norteadores - Aprendizagem Colaborativa

A **aprendizagem colaborativa** é uma abordagem educacional que envolve a interação e colaboração entre os alunos com o objetivo comum de aprendizagem. Nessa metodologia, os estudantes trabalham juntos em tarefas, projetos ou discussões em grupo, contribuindo com suas habilidades, conhecimentos e perspectivas individuais. Essa abordagem oferece diversos benefícios aos alunos, incluindo:

Desenvolvimento de habilidades sociais importantes, como trabalho em equipe, comunicação, liderança e resolução de conflitos. Aprendizagem ativa, incentivando os alunos a serem participativos em sua própria educação, trabalhando juntos para resolver problemas e criar soluções. Compartilhamento de conhecimentos, criando um ambiente de aprendizado mais rico e diversificado. Desenvolvimento de pensamento crítico, permitindo que os alunos avaliem e analisem informações, discutam ideias e tomem decisões informadas. Aumento da motivação, pois os alunos trabalham juntos para alcançar um objetivo comum e se apoiam mutuamente no processo. Diversidade de perspectivas, envolvendo a participação de vários alunos, cada um com sua perspectiva única e experiência. Preparação para a vida profissional, uma vez que muitos trabalhos envolvem trabalho em equipe e colaboração para alcançar metas compartilhadas. A Figura 13 ilustra a abordagem da Aprendizagem Colaborativa.

Figura 13. Abordagem da Aprendizagem Colaborativa



Fonte: O que é aprendizagem colaborativa e por que está revolucionando a aprendizagem e o desenvolvimento?²

Apesar dos diversos benefícios da aprendizagem colaborativa, essa abordagem educacional também pode apresentar alguns desafios e problemas, tais como:

² <https://www.taqe.com.br/o-que-e-aprendizagem-colaborativa-e-por-que-esta-revolucionando-a-aprendizagem-e-o-desenvolvimento/>

Dependência excessiva dos demais membros do grupo, o que pode levar alguns estudantes a não assumirem a responsabilidade individual pelo aprendizado. Dificuldades na comunicação, especialmente quando os membros possuem opiniões divergentes ou enfrentam conflitos, podendo afetar a eficácia do trabalho em grupo. Dificuldades em lidar com conflitos, o que pode gerar tensões dentro do grupo e prejudicar a produtividade e o sucesso do projeto. Falta de responsabilidade, podendo ocasionar atrasos e até mesmo o fracasso do trabalho, caso alguns membros não cumpram com suas obrigações. Dificuldades em gerenciar o tempo, já que a aprendizagem colaborativa pode exigir muito tempo e esforço dos membros do grupo, o que pode impactar outros compromissos individuais.

O estudo de Torres e Irala (2014) enfatiza a importância da aprendizagem colaborativa como uma metodologia eficaz para promover a aprendizagem significativa dos alunos e apresenta alguns princípios e estratégias que podem ser adotados pelos educadores para implementar essa abordagem em sala de aula. Por outro lado, o artigo de Machado, Berkenbrock e Freitas (2019) discute a relevância da colaboração na aprendizagem de programação de computadores e propõe uma abordagem colaborativa baseada em dispositivos móveis. O estudo também destaca a importância da integração entre a teoria e a prática na aprendizagem de programação. Os autores concluem que a abordagem colaborativa baseada em dispositivos móveis pode ser uma ferramenta valiosa para melhorar a qualidade da educação em programação de computadores.

Em um estudo de caso apresentado por Alves, Rebouças e Scaico (2019), envolvendo estudantes de graduação em Ciência da Computação, a prática de *Coding Dojo* foi analisada para entender como ela pode ajudar os alunos a desenvolver habilidades de programação, trabalho em equipe e solução de problemas. Os resultados demonstram que a prática de *Coding Dojo* pode ser uma estratégia eficaz para a aprendizagem colaborativa de programação, pois os alunos tiveram a oportunidade de trabalhar em equipe e compartilhar conhecimentos e experiências, além de aumentar a motivação para aprender programação e aprimorar a qualidade do trabalho em equipe.

Diversas pesquisas apontam para evidências que comprovam a eficácia da aprendizagem colaborativa, incluindo melhorias na assimilação do conhecimento, desenvolvimento de competências sociais, aprimoramento da retenção de informações, maior envolvimento dos participantes e preparação para a carreira profissional.

3.2.2 Princípios Norteadores - Programação em Pares

A **Programação em Pares** é uma técnica de programação que envolve a colaboração de dois programadores trabalhando no mesmo código-fonte, utilizando apenas um computador. Durante o processo, um dos programadores escreve o código enquanto o outro observa, faz questionamentos, oferece sugestões e identifica possíveis erros. Após um período pré-determinado, eles trocam de posições, com o programador que estava

observando assumindo a tarefa de escrever o código, enquanto o outro passa a observar. A proposta por trás dessa abordagem é que a cooperação dos programadores pode resultar em um código de melhor qualidade, com menor incidência de erros e maior rapidez. Além disso, a técnica estimula o compartilhamento de conhecimentos e a manutenção de um alto nível de qualidade, uma vez que ambos têm a oportunidade de revisar e discutir o código criado. A Figura 14 ilustra um exemplo da técnica sendo utilizada em sala de aula.

A técnica de programação em pares é vantajosa em diversos aspectos, tais como:

Melhoria na qualidade do código: dois programadores trabalham em conjunto e revisam constantemente o código, resultando em menor quantidade de erros e maior qualidade. Aumento da produtividade: é possível resolver problemas mais rapidamente e trabalhar de forma mais eficiente. Oportunidade de aprendizado e compartilhamento de conhecimentos: os programadores podem aprender uns com os outros e aprimorar suas habilidades. Menor interrupção externa: um programador fica focado na escrita de código enquanto o outro monitora, minimizando as interrupções e permitindo que ambos trabalhem com mais eficiência. Entretanto, há também desvantagens na aplicação dessa técnica, tais como:

Resistência de alguns programadores em trabalhar em equipe: nem todos os profissionais estão dispostos a colaborar de forma tão próxima com outro colega. Sentimento de limitação criativa ou de ideias: alguns programadores podem sentir que sua criatividade ou liberdade para expressar ideias é restringida nesse modelo de trabalho. Possibilidade de desequilíbrio na carga de trabalho: se os programadores possuem habilidades muito diferentes, pode haver um desequilíbrio na distribuição das tarefas, gerando insatisfação. Críticas à técnica argumentam que a programação em pares pode ser uma perda de tempo e que o trabalho individual é mais eficiente. No entanto, muitos defensores destacam que os benefícios superam as desvantagens, e a abordagem de programação em pares continua sendo uma prática amplamente utilizada e bem-sucedida em diversas equipes de desenvolvimento de *software*.

³ <https://c2ti.com.br/blog/conheca-as-vantagens-e-os-desafios-do-pair-programming-tecnologia>

Figura 14. Técnica de programação em pares em sala de aula



Fonte: Conheça as vantagens e os desafios do Pair Programming³

Um estudo apresentado por Hernández et al. (2019) comparou a programação em pares e o trabalho individual, utilizando métricas de *software* para projetos de estudantes universitários. Os resultados mostraram que a equipe que trabalhou em pares produziu código com menor complexidade ciclomática, menor número de linhas de código e menor número de classes e métodos em comparação com a equipe que trabalhou individualmente. Além disso, os estudantes que trabalharam em equipe relataram maior satisfação com o processo de programação.

Em outro estudo elaborado por Estácio e Prikladnicki (2013), a eficácia da Programação em Par Distribuída (PPD) no ensino de programação foi avaliada em comparação com outros métodos de ensino, como aulas expositivas ou trabalho individual. Os resultados sugerem que a PPD pode ser uma abordagem efetiva e inovadora para o ensino de programação, proporcionando uma experiência mais ativa e colaborativa para os alunos.

A técnica de programação em pares tem sido extensivamente explorada em pesquisas acadêmicas, e estudos já comprovaram sua eficácia. Entre as principais evidências encontradas estão a melhoria da qualidade do código, aumento da produtividade, redução do tempo de revisão do código, maior satisfação e engajamento dos programadores, e aprimoramento das habilidades de comunicação e colaboração. No entanto, é importante ressaltar que a técnica pode não ser apropriada para todas as equipes e situações, e que os resultados podem variar de acordo com o contexto e a implementação da técnica.

3.2.3 Princípios Norteadores - Aprendizagem Baseada em Projetos

A **Aprendizagem Baseada em Projetos (ABP)** é uma abordagem que envolve a realização de projetos pelos alunos. Esses projetos geralmente estão relacionados à resolução de problemas reais ou desafios que exigem a aplicação de conhecimentos e habilidades específicas. A ABP tem como objetivo desenvolver habilidades cognitivas e socioemocionais, como pensamento crítico, resolução de problemas, comunicação, colaboração e autogestão. Além disso, a ABP ajuda a conectar a aprendizagem à vida real e a motivar os alunos a aprender, pois o projeto em si é algo que os alunos se sentem motivados a resolver.

A ABP apresenta diversos benefícios, como o aprimoramento do aprendizado, o desenvolvimento de habilidades socioemocionais dos alunos e a preparação para o mercado de trabalho. No entanto, há desvantagens, como a dificuldade de implementação, a limitação de tempo e a avaliação dos projetos. Além disso, críticas podem ser apontadas para a ABP, como o foco excessivo em projetos em detrimento do conteúdo teórico, a ineficácia para alunos com necessidades especiais e com dificuldades de aprendizagem, e a indicação restrita para disciplinas específicas, como ciência e tecnologia.

O estudo apresentado por Oliveira, Siqueira e Romão (2020) realizou uma comparação entre métodos de ensino, buscando avaliar a eficácia da ABP em relação a outros métodos no ensino médio. Os alunos foram distribuídos em três grupos distintos: um grupo que participou de um projeto de ABP, um grupo que recebeu aulas tradicionais e um grupo que participou de uma atividade prática, mas não relacionada a um projeto de ABP. Os resultados evidenciaram que o grupo que participou do projeto de ABP obteve melhores resultados em termos de desempenho acadêmico em comparação aos outros grupos. Além disso, os alunos desse grupo demonstraram maior motivação e engajamento no processo de aprendizagem.

Já Grotta e Prado (2018) discutem as diferenças entre a abordagem tradicional de ensino de programação de computadores e a ABP. O estudo apresenta exemplos de projetos de ABP em programação de computadores, como a criação de jogos, aplicativos móveis e programas para análise de dados. Os autores debatem como esses projetos podem auxiliar os alunos a desenvolver habilidades de pensamento crítico, resolução de problemas, trabalho em equipe, comunicação, criatividade e interesse pela programação. De acordo com os autores, a ABP pode ser uma abordagem mais eficaz para o ensino de programação de computadores, pois promove o desenvolvimento de habilidades além das habilidades técnicas. Entretanto, reconhecem que a ABP pode requerer mais tempo e recursos em comparação à abordagem tradicional de ensino de programação de computadores.

Os estudos mencionados apresentam evidências da eficácia da ABP na educação, demonstrando benefícios como o aumento do engajamento e motivação dos alunos, desenvolvimento de habilidades de pensamento crítico e resolução de problemas, aprimoramento das

habilidades sociais e de colaboração, aprendizado significativo e melhora do desempenho acadêmico.

A Figura 15 ilustra um exemplo prático de ABP onde alunos do ensino fundamental criaram em sala de aula uma placa de energia solar capaz de carregar um celular.

Figura 15. Exemplo de ABP na prática



Fonte: Aprendizagem baseada em projetos estimula alunos do ensino básico⁴

3.2.4 Princípios Norteadores - Autoria

A **autoria** refere-se à criação de uma obra original, que pode ser um livro, um filme, uma música, um poema, uma pintura ou qualquer outra forma de expressão artística, científica ou intelectual. Na educação, a autoria é definida como a habilidade dos alunos de produzir suas próprias criações intelectuais, utilizando seus conhecimentos e habilidades para criar algo novo e original. Isso pode incluir a produção de textos, projetos, apresentações, pesquisas e outras atividades que exigem a expressão criativa do pensamento e da experiência pessoal do aluno.

Promover a autoria na educação é vista como uma forma de desenvolver habilidades como pensamento crítico, criatividade, comunicação e autonomia dos alunos, incentivando-os a se envolverem mais ativamente em seu próprio processo de aprendizagem.

A autoria na educação pode trazer diversos benefícios para os alunos, como maior engajamento, motivação, autonomia e desenvolvimento de habilidades importantes para a vida pessoal e profissional, além de melhorias no desempenho acadêmico.

⁴ <https://desafiosdaeducacao.com.br/aprendizagem-baseada-em-projetos/>

No entanto, também existem críticas e desvantagens em relação ao processo de autoria, como a falta de interesse ou habilidade de alguns alunos em produzir criações intelectuais, a percepção da autoria como um privilégio de alunos com maiores recursos ou habilidades, a dificuldade de alguns alunos em lidar com a liberdade e a responsabilidade envolvidas na autoria e a subjetividade na avaliação da autoria, o que pode tornar a medição do desempenho desafiadora.

Figura 16. Representações do processo de autoria



Fonte: Atividades autorais online: aprendendo com criatividade⁵

De acordo com Veloso, Bonilla e Pretto (2016), a cultura da liberdade de criação é crucial para promover a autoria na educação, pois permite que os alunos expressem suas próprias ideias e opiniões, valorizando assim sua capacidade criativa e inovadora. As autoras ilustram essa questão com dois casos: o primeiro envolve um projeto de produção de vídeo em que os alunos tiveram total liberdade para criar seu roteiro e produzir seu filme. Já o segundo caso se trata de uma aula de redação em que os alunos foram orientados a seguir um modelo pré-determinado, sem muita liberdade para expressar suas próprias ideias e opiniões. Ao comparar os dois casos, as autoras concluem que a liberdade de criação é essencial para o desenvolvimento da autoria dos alunos e que limitações tecnológicas e normativas podem restringir essa liberdade e prejudicar o processo de aprendizagem.

A literatura apresenta evidências que comprovam a eficácia da autoria na educação, tanto em relação aos processos de aprendizagem dos alunos quanto aos resultados obtidos em termos de desempenho acadêmico. Dentre as evidências, destacam-se o desenvolvimento

⁵ <https://horizontes.sbc.org.br/index.php/2020/11/atividadesautorais/>

da criatividade e da capacidade de pensamento crítico, o aumento do engajamento e da motivação dos alunos, o favorecimento da aprendizagem significativa, o estímulo à autoavaliação e ao *feedback*, e a preparação para a trajetória profissional.

3.2.5 Princípios Norteadores - Ludicidade

A **ludicidade** se refere ao aspecto recreativo de uma atividade ou situação, associado à brincadeira, jogo e diversão. Sua presença pode ser notada em diversas áreas da vida, incluindo a educação, o lazer e o trabalho. A ludicidade é importante para o desenvolvimento cognitivo, social e emocional, principalmente das crianças. Contudo, ela não se limita a essa faixa etária, uma vez que adultos também podem desfrutar de atividades lúdicas que proporcionam benefícios como redução do estresse, aumento da criatividade e melhora do bem-estar em geral.

Na área da educação, o termo ludicidade se refere à utilização de jogos, brincadeiras e atividades lúdicas como método de ensino e aprendizagem. O objetivo é criar um ambiente mais dinâmico e envolvente para os alunos, fomentando a participação ativa, a criatividade e o interesse pelos conteúdos abordados. Essa abordagem pode ser aplicada em todas as etapas do processo educacional, desde a educação infantil até o ensino superior.

A introdução da ludicidade na educação tem como intuito tornar o aprendizado mais prazeroso, diminuindo a ansiedade e o estresse comuns ao processo de ensino e aprendizagem. Com o uso de atividades lúdicas, os estudantes são incentivados a desenvolver habilidades como a criatividade, imaginação, solução de problemas e trabalho em equipe, que são fundamentais para a formação de cidadãos críticos e participativos.

A ludicidade apresenta diversos benefícios quando aplicada de forma adequada no processo de ensino e aprendizagem, tais como contribuir para o desenvolvimento cognitivo, emocional e social dos estudantes, estimular a criatividade, a imaginação, a capacidade de resolver problemas e a trabalhar em equipe, tornar o processo de aprendizagem mais prazeroso e significativo, reduzindo a ansiedade e o estresse relacionados ao estudo, trabalhar diferentes habilidades e competências e diversificar o ensino, tornando-o mais dinâmico e atrativo.

Contudo, há desvantagens na sua utilização inadequada, como quando é utilizada apenas como um meio de entretenimento, sem considerar os objetivos educacionais, ou quando desvia o foco do ensino dos conteúdos fundamentais para as atividades lúdicas, não promovendo uma aprendizagem efetiva. Além disso, a ludicidade pode ser vista como uma forma de mascarar a falta de competência ou de preparo do professor.

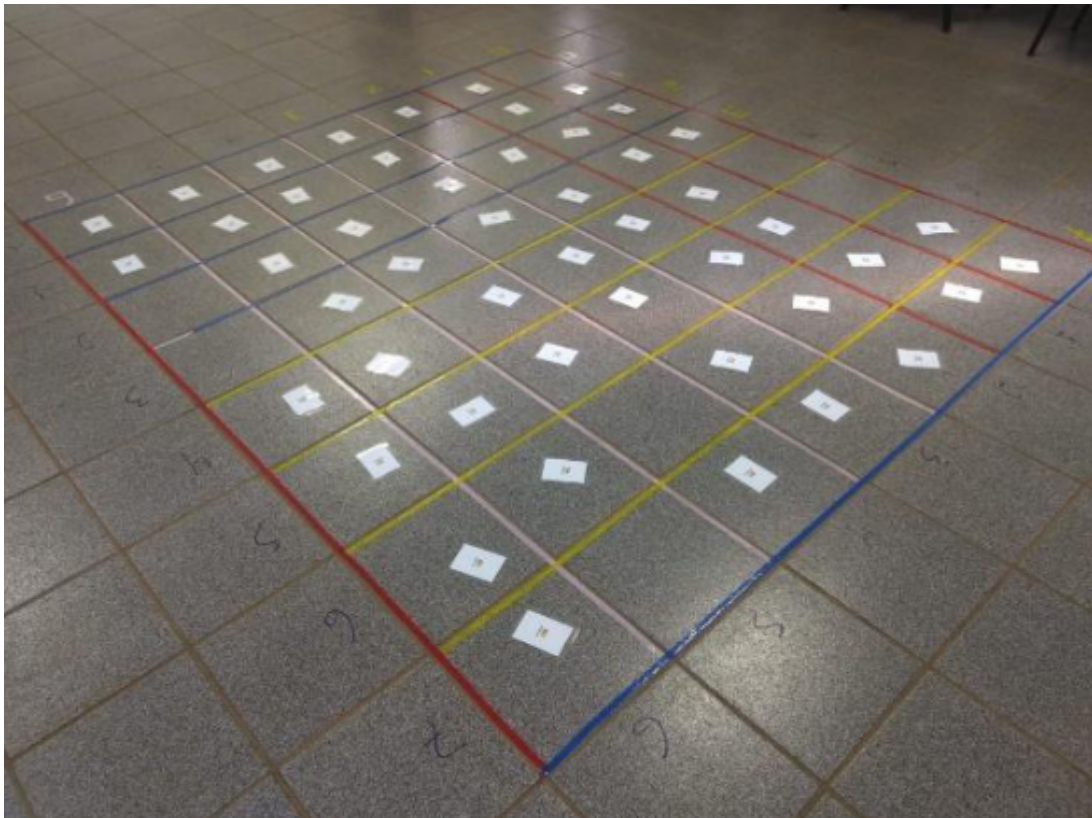
É importante destacar que há críticas quanto à efetividade da ludicidade como metodologia de ensino, argumentando que ela pode ser vista como uma forma de distração ou de fuga do aprendizado mais sério, além de ser uma forma de elitização da educação, já

que nem todos os estudantes têm acesso a atividades lúdicas e aos recursos necessários para realizá-las.

No artigo apresentado por Guarda, Rezende e Pinto (2022), foi descrita uma abordagem pedagógica lúdica utilizada para ensinar algoritmos de pesquisa sequencial e binária para 60 alunos de nível superior na disciplina de Algoritmos e Programação. Os resultados obtidos indicaram uma melhoria no desempenho acadêmico dos alunos, bem como maior engajamento e motivação no processo de aprendizagem.

A Figura 17 apresenta a atividade lúdica utilizada no estudo, que envolvia equipes em uma matriz humana de tamanho 8x8, onde cada espaço continha um desafio que deveria ser respondido pelas equipes para avançarem no jogo.

Figura 17. Atividade lúdica realizada com os alunos



Fonte: Guarda, Rezende e Pinto (2022, p. 3)

A abordagem pedagógica da ludicidade tem sido amplamente reconhecida como eficaz no contexto educacional, especialmente na educação infantil, devido ao aumento da motivação e do engajamento dos alunos, desenvolvimento cognitivo, social e emocional, melhoria do desempenho acadêmico e redução do estresse e ansiedade. Educadores defendem que a ludicidade é uma forma holística e humanizada de abordar a educação, permitindo que as crianças desenvolvam habilidades acadêmicas, sociais, emocionais e criativas.

Contudo, a eficácia dessa abordagem depende da adaptação ao contexto e às necessidades dos alunos, bem como a forma como é aplicada.

3.2.6 Princípios Norteadores - Jogos digitais

Um **jogo digital** é um tipo de jogo eletrônico que pode ser jogado em diferentes dispositivos eletrônicos, como computadores, consoles de videogame, *smartphones* e *tablets*. Caracteriza-se por ser um jogo interativo e imersivo que utiliza tecnologia digital para criar uma experiência de jogo envolvente para o usuário. Nos últimos anos, os jogos digitais tornaram-se cada vez mais populares, com uma grande variedade de títulos famosos e bem-sucedidos em todo o mundo, oferecendo diferentes tipos de desafios, experiências e diversão para jogadores de todas as idades e habilidades.

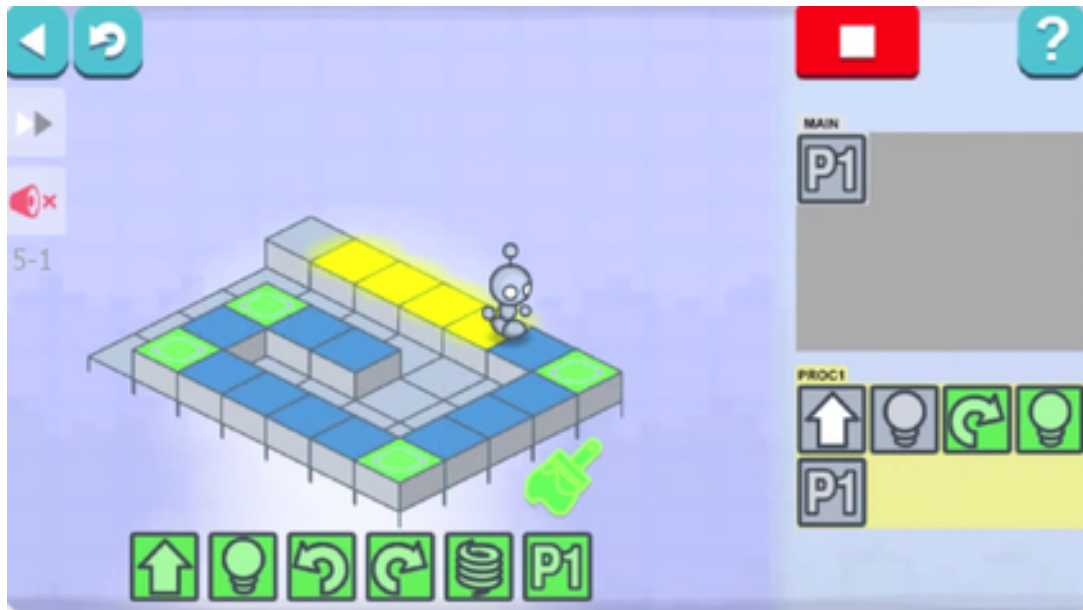
No contexto da educação, os jogos digitais são desenvolvidos com o objetivo de ensinar habilidades ou conhecimentos específicos, proporcionando um ambiente lúdico e interativo para a aprendizagem. Esses jogos são projetados para ajudar os estudantes a aprender de forma mais eficaz e envolvente, tornando o processo de aprendizado mais prazeroso e estimulante.

Os jogos digitais na educação apresentam diversas vantagens, tais como o desenvolvimento de habilidades cognitivas como resolução de problemas, pensamento crítico, criatividade e colaboração. Além disso, podem aumentar a motivação dos estudantes e despertar maior interesse pelo aprendizado.

O artigo de Monclar, Silva e Xexéo (2018) realizou uma revisão entre os anos de 2001 e 2018 para identificar jogos digitais que possam auxiliar no ensino de programação para estudantes. O estudo identificou 26 jogos divididos em duas categorias, que apresentam soluções válidas para melhorar o desempenho dos alunos. Um dos jogos identificados é o Lightbot, ilustrado na Figura 18, onde os estudantes encontram comandos pré-definidos e precisam escolher aqueles que devem ser utilizados para solucionar o problema apresentado.

⁶ <https://en.wikipedia.org/wiki/Lightbot>

Figura 18. Ilustração da interface do jogo Lightbot



Fonte: Lightbot⁶

Por outro lado, o artigo de Marques et al. (2011) apresenta uma experiência prática de ensino de programação para alunos do ensino médio, utilizando jogos e a língua de programação Python, com o objetivo de atrair mais estudantes para o campo da computação. A metodologia utilizada consiste em ensinar programação através de jogos, permitindo que os alunos aprendam de forma lúdica e prática. Os resultados mostraram que a abordagem de ensino por meio de jogos foi bem-sucedida em atrair a atenção dos alunos para a programação, aumentando o interesse e o engajamento na disciplina. Conclui-se, portanto, que o uso de jogos como estratégia de ensino pode ser uma maneira eficaz de atrair mais alunos para o campo da computação.

Há evidências que comprovam a eficácia dos jogos digitais no contexto educacional. Alguns exemplos incluem: aumento na retenção e no aprendizado, fomento ao pensamento crítico e à solução de problemas, possibilidade de aprendizado personalizado, aprimoramento da colaboração e do trabalho em equipe, utilização de *feedback* imediato e melhora da capacidade de resolução de conflitos e comunicação.

3.2.7 Princípios Norteadores - Do concreto ao abstrato

Diversos pesquisadores e professores defendem que o ensino de programação deve começar do abstrato para o concreto. No entanto, esta pesquisa propõe o oposto: que o aluno **comece com o mais concreto e caminhe em direção ao mais abstrato**. A ideia é começar com coisas palpáveis, como entender o conceito de comandos em programação por meio de jogos digitais ou atividades de computação desplugada. Desafios simples e concretos podem ser apresentados aos estudantes, permitindo a associação do conceito à

prática, sem a necessidade de escrever código em uma língua específica.

Ao iniciar uma disciplina de introdução à programação, é comum que alunos sem experiência prévia na área enfrentem dificuldades. A compreensão de conceitos como repetição pode ser facilitada ao utilizar noções presentes em jogos. Por exemplo, a noção de passos de movimentação de um personagem em um jogo pode tornar a compreensão do conceito mais simples para o aluno. Dessa forma, o estudante é capaz de compreender o resultado alcançado no jogo de forma imediata, o que pode facilitar o aprendizado em uma língua de programação específica, como C.

3.2.8 Princípios Norteadores - Letramento em programação

Letramento é o conjunto de habilidades e competências relacionadas à leitura, escrita e interpretação de textos, bem como à compreensão crítica do mundo que nos cerca por meio dessas práticas. O letramento não se limita apenas às habilidades técnicas de leitura e escrita, mas também envolve a compreensão de como a linguagem e as práticas de comunicação influenciam e são influenciadas pela cultura, política e sociedade. Ser letrado significa ser capaz de participar ativamente da vida social, política e cultural de uma comunidade, utilizando a leitura e a escrita como instrumentos para compreender e transformar o mundo.

O letramento em programação engloba as habilidades e competências necessárias para utilizar línguas de programação na resolução de problemas e criação de soluções tecnológicas. Isso abrange não apenas a compreensão de conceitos e técnicas de programação, mas também a capacidade de pensar logicamente, desenvolver algoritmos e trabalhar com tecnologias e plataformas tecnológicas.

Para ensinar programação, o letramento é transmitido por meio de uma variedade de atividades e estratégias, como a resolução de problemas, a criação de projetos, a utilização de tutoriais e exemplos, e a colaboração com colegas e professores. O objetivo é capacitar os alunos a se tornarem programadores competentes e criativos, capazes de aplicar essas habilidades de forma eficaz em suas vidas pessoais e profissionais.

O letramento em programação é uma abordagem abrangente que vai além da simples compreensão da sintaxe e estrutura das línguas de programação. Ele inclui o entendimento dos conceitos fundamentais da computação, como algoritmos, lógica de programação e pensamento computacional. Além disso, o letramento em programação enfatiza a importância da ética e da responsabilidade no uso da tecnologia. Ao adquirir essas habilidades, os alunos estarão melhor preparados para enfrentar os desafios da sociedade digital em constante evolução e poderão contribuir de forma significativa para o avanço tecnológico e social.

No ensino de programação, é comum que os alunos memorizem comandos e es-

truturas por meio da repetição. No entanto, o verdadeiro objetivo do letramento em programação é permitir que os estudantes desenvolvam a habilidade de escrever qualquer código necessário para alcançar seus objetivos. Isso significa que o foco deve estar na compreensão dos conceitos e técnicas de programação, e não apenas na memorização de códigos prontos. Ao desenvolver essa habilidade, os alunos se tornam capazes de criar soluções personalizadas e inovadoras para os problemas que encontram, em vez de depender de soluções pré-fabricadas.

3.2.9 Princípios Norteadores - Pensamento Criativo

O processo mental do **pensamento criativo** envolve a geração de novas ideias, conceitos, conexões e soluções para um problema ou desafio. Esse tipo de pensamento divergente busca explorar diversas possibilidades e abordagens, em vez de seguir um caminho único e previsível. Diversos fatores podem estimular o pensamento criativo, como a curiosidade, a imaginação, a flexibilidade cognitiva e a habilidade de buscar inspiração em diversas fontes e experiências.

No contexto educacional, o pensamento criativo refere-se à capacidade dos alunos de utilizar a imaginação e a originalidade para gerar ideias, resolver problemas e abordar questões de forma inovadora. Embora a promoção do pensamento criativo no ambiente escolar seja fundamental para o desenvolvimento das habilidades cognitivas e socioemocionais dos alunos, tais como o pensamento crítico, análise de informações, tomada de decisões e trabalho em equipe, existem problemas e críticas relacionados à sua adoção.

Dentre essas questões, destacam-se a falta de apoio institucional, a dificuldade em avaliar o pensamento criativo, a resistência por parte de educadores e alunos, a ausência de rigor acadêmico e o excesso de foco no indivíduo.

O artigo apresentado por Nunes et al. (2018) relata uma pesquisa que teve como objetivo avaliar o impacto do ensino de lógica de programação no desenvolvimento do pensamento criativo em crianças do ensino básico. A pesquisa incluiu atividades práticas de programação em sala de aula, estimulando as crianças a explorar diferentes soluções para os problemas propostos. Os resultados apontaram que o ensino de lógica de programação pode ter um impacto significativo no desenvolvimento do pensamento criativo em crianças, promovendo sua capacidade de gerar novas ideias, explorar abordagens diversas e encontrar soluções criativas, além de melhorar sua autoconfiança e motivação.

Em outro estudo conduzido por Bressan e Amaral (2015), foi avaliado o impacto do uso do Scratch na aprendizagem por meio da solução de problemas e no desenvolvimento do pensamento criativo em estudantes do ensino fundamental. O estudo envolveu a aplicação de atividades práticas utilizando o Scratch, encorajando os estudantes a criar projetos e soluções criativas para enfrentar diferentes desafios. Os resultados sugerem que o uso

do Scratch pode ser efetivo para o desenvolvimento do pensamento criativo, visto que os estudantes demonstraram habilidades de análise, síntese, resolução de problemas e criação de soluções originais.

Os estudos mencionados reforçam a existência de evidências que comprovam a eficácia do estímulo ao pensamento criativo no contexto da educação. Essas pesquisas evidenciam que o desenvolvimento do pensamento criativo resulta em melhorias na resolução de problemas, na autoconfiança e na autoestima dos alunos. Além disso, o estímulo à criatividade promove o desenvolvimento de habilidades de pensamento crítico, incentiva a curiosidade e a imaginação, e contribui para aprimorar a comunicação e a colaboração entre os estudantes.

3.2.10 Princípios Norteadores - Princípio do *design*

O **princípio do *design*** na computação se baseia nos mesmos conceitos fundamentais aplicados na área do *design* em geral, mas com um foco específico na criação de interfaces de usuário e experiências digitais. O objetivo é desenvolver interfaces que sejam fáceis de usar, intuitivas e agradáveis para o usuário, possibilitando uma navegação fluida pelas funcionalidades oferecidas pelo *software* ou aplicativo. Algumas diretrizes comuns do princípio do *design* na computação incluem: usabilidade, consistência, acessibilidade, *feedback* e estética.

O princípio do *design* no ensino de programação refere-se à aplicação dos conceitos de *design* na criação de materiais e recursos de ensino eficazes, que auxiliem os alunos a aprender programação de forma clara e compreensível. Essa abordagem é fundamental para o desenvolvimento de materiais de ensino que aumentem o engajamento e o sucesso no aprendizado dos alunos. Além disso, o uso de recursos de ensino bem projetados pode aprimorar a eficácia do processo de ensino e aprendizagem como um todo.

Muitos professores utilizam exercícios matemáticos para ensinar programação, mas essa prática pode dificultar ainda mais a compreensão dos alunos, especialmente quando a complexidade da solução aumenta de acordo com o conhecimento prévio (ou falta dele) em matemática. Por essa razão, é fundamental incluir problemas cotidianos da computação no ensino de programação, como a criptografia de mensagens ou o cálculo de juros em um boleto vencido. Esses exemplos estão diretamente relacionados às atividades que um profissional de Sistemas de Informação pode encontrar em seu trabalho.

3.2.11 Matriz SWOT

Após identificar e selecionar as práticas, técnicas e metodologias ativas orientadoras, iniciei a elaboração de uma matriz SWOT para analisar as variáveis que poderiam afetar a solução. A matriz SWOT é uma ferramenta estratégica que auxilia na identificação

das forças (*Strengths*), fraquezas (*Weaknesses*), oportunidades (*Opportunities*) e ameaças (*Threats*) de uma organização em relação ao seu ambiente interno e externo. Com base nessas informações, é possível desenvolver estratégias mais eficazes e tomar decisões mais fundamentadas.

A utilização da matriz SWOT pode ser uma ferramenta valiosa para o planejamento didático, pois permite avaliar os pontos fortes e fracos do plano de ensino, bem como as oportunidades e ameaças que podem impactar o sucesso da implementação. No entanto, é importante ressaltar que a matriz SWOT deve ser considerada como um guia para o planejamento didático e não como uma solução definitiva.

A figura apresentada na Figura 19 detalha todos os itens identificados em cada uma das áreas da matriz SWOT que podem afetar a solução proposta. No ambiente interno, foram identificadas diversas forças que podem influenciar positivamente a implementação da solução. Entre elas, destacam-se a experiência e qualificação do professor em programação, a disponibilidade de recursos tecnológicos na escola, como laboratórios para aulas e projetores, e o apoio da coordenação do curso de informática e da direção da escola para implementar melhorias na disciplina.

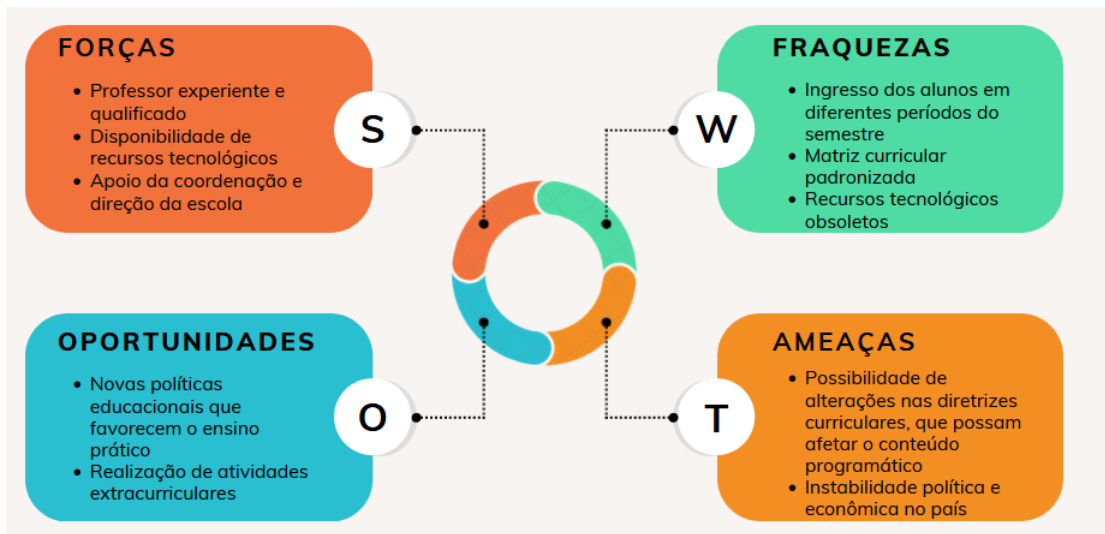
Por outro lado, também foram identificadas algumas fraquezas que representam desafios a serem superados. Entre as fraquezas identificadas, está o ingresso dos alunos em diferentes momentos do semestre, o que dificulta a sequência do planejamento da disciplina. Além disso, a existência de uma matriz curricular padronizada para toda a rede pode impedir a adaptação do conteúdo programático às necessidades específicas dos alunos e à abordagem de ensino proposta. Outra fraqueza apontada é a presença de recursos tecnológicos obsoletos, que podem dificultar ou interromper as atividades planejadas para as aulas.

As oportunidades identificadas incluem a criação de novas políticas educacionais que favorecem o ensino prático e crítico, fortalecendo a abordagem tradicional de ensino. Essas políticas podem abrir espaço para a implementação de metodologias ativas, como o ensino de programação de forma mais prática e voltada para a resolução de problemas reais. Além disso, a realização de atividades extracurriculares que permitem aos alunos vivenciar experiências do mercado de trabalho pode estimular o interesse pela disciplina e aumentar a compreensão da sua aplicação no contexto profissional.

Por outro lado, as ameaças identificadas são fatores que podem impactar negativamente a implementação da solução proposta. Entre elas, destacam-se possíveis mudanças nas diretrizes curriculares que afetem o conteúdo da disciplina, como a inclusão ou alteração de algum segmento específico. Essas mudanças podem exigir adaptações no plano de ensino, a fim de garantir a adequação aos novos requisitos. Além disso, a instabilidade política e econômica do país pode gerar impactos na educação, como a evasão dos alunos devido à violência no entorno da escola ou questões financeiras que afetem a continuidade dos

estudos.

Figura 19. Matriz SWOT



Fonte: Elaborada pelo autor

3.2.12 Planejamento Estratégico

Na última etapa da fase de ideação, foi realizada uma avaliação dos artefatos disponíveis para o planejamento estratégico. Após uma revisão conceitual dos artefatos existentes no mercado, foi concluído que o Canvas é o mais adequado ao contexto da pesquisa.

O Canvas, também conhecido como *Business Model Canvas*⁷, é uma ferramenta visual de planejamento estratégico que auxilia na organização de ideias sobre um negócio, permitindo o desenvolvimento e esboço de modelos de negócios novos ou existentes de forma simples e fácil de ser consultada e atualizada. Diferentemente de um plano de negócios tradicional, que é um documento complexo e frequentemente acaba sendo arquivado em uma gaveta, o Canvas é uma ferramenta muito mais visual e intuitiva (FERNANDES; FERREIRA; CALVOSA, 2021).

A palavra *canvas* significa tela, e a ferramenta é, de fato, um quadro em que é possível descrever a essência do negócio e todos os pontos necessários para o seu funcionamento. Ele é composto por nove blocos, que representam os principais elementos do modelo de negócios, como segmento de clientes, proposta de valor, canais de distribuição, relacionamento com clientes, fontes de receita, recursos-chave, atividades-chave, parcerias principais e estrutura de custos.

Através da Figura 20, podemos visualizar um exemplo de como as ideias são representadas nos blocos que formam a conceitualização do negócio. Isso inclui a forma

⁷ <https://blog.eadplataforma.com/empreendedorismo-digital/modelo-canvas>

como a empresa irá operar e gerar valor para o mercado, definindo seus principais fluxos e processos. O Canvas possibilita uma análise e visualização do modelo de atuação no mercado de maneira clara e concisa.

Figura 20. Exemplo de Canvas para um negócio



Fonte: Canvas: Como estruturar seu modelo de negócios⁸

Como mencionado, o Canvas foi originalmente desenvolvido para uso em negócios e empresas, sejam elas novas ou estabelecidas. Seus principais objetivos são auxiliar no lançamento de novos produtos, modificar modelos de negócios, atingir diferentes públicos e mercados, e fornecer uma visão geral da situação atual da empresa. Esses objetivos ajudam os gestores a criar produtos que entreguem mais valor aos clientes e tornem a empresa mais competitiva.

Para adequar o Canvas ao contexto da pesquisa, foi criado um modelo de quatro áreas para utilizá-lo no planejamento da disciplina. Essas áreas são: COMO?, O QUÊ?, PARA QUÊ? e PARA QUEM?.

Na área COMO?, buscamos entender como alcançaríamos os resultados desejados na disciplina e identificamos as plataformas, canais e recursos que seriam utilizados na sala de aula com os alunos. Nessa seção, foram delineadas as estratégias de ensino e as metodologias ativas que seriam empregadas para promover a aprendizagem efetiva dos estudantes.

⁸ <https://www.sebraepr.com.br/canvas-como-estruturar-seu-modelo-de-negocios/>

A área O QUÊ? nos ajudou a identificar as ações específicas que seriam tomadas na disciplina em sala de aula. Nesse bloco, foram listados os conteúdos programáticos, os projetos práticos, as atividades de laboratório e as avaliações que comporiam o desenvolvimento do conteúdo da disciplina.

Já em PARA QUÊ?, definimos os objetivos que deveriam ser alcançados pela disciplina e mapeamos as atividades de acompanhamento e avaliação. Nessa parte do Canvas, foram estabelecidos os resultados de aprendizagem esperados e os critérios de avaliação para mensurar o progresso dos alunos.

Finalmente, em PARA QUEM?, identificamos o público-alvo da disciplina e os componentes curriculares que deveriam ser utilizados. Nesse bloco, foram destacados os pré-requisitos dos estudantes para cursarem a disciplina, bem como a integração da matéria com outras disciplinas do curso de informática.

A Figura 21 ilustra o resultado do Canvas elaborado para o planejamento da disciplina Linguagem de Programação 1. Essa visualização do Canvas permitiu uma representação gráfica e organizada de todas as informações relevantes para o planejamento da disciplina, facilitando a comunicação e compreensão entre os envolvidos no processo de ensino.

Essa abordagem estratégica possibilitou uma visão integrada e abrangente do planejamento da disciplina, garantindo que todas as áreas essenciais para o seu desenvolvimento fossem devidamente consideradas e alinhadas com os objetivos educacionais propostos.

Figura 21. Planejamento estratégico para a disciplina LP1

COMO?	O QUÊ?	PARA QUEM?
Plataformas, canais e recursos: Google Classroom; Hora do Código; Lightbot; Scratch; Calango; Língua JAVA;	Descrição das ações: Aprender fazendo; Programação com blocos; Aprendizagem através de jogos; Atividades lúdicas e autorais; Utilização de recursos multimídias; Aprendizagem baseada em projetos; Disponibilização de materiais de apoio; Aprendizagem colaborativa;	Unidade(s): ESCOLA TÉCNICA ESTADUAL MARIA MERCEDES MENDES TEIXEIRA
Período: Fevereiro a Julho de 2022.		Público-alvo: Estudantes ingressantes no 1º módulo do curso técnico subsequente de Informática.
PARA QUÊ?	Objetivos: Desenvolver a habilidade de programar: desenvolver as habilidades do Pensamento Computacional (decomposição, reconhecimento de padrões, abstração e algoritmo); Conhecer as estruturas das linguagens de programação: se expressar em diferentes línguas (blocos, fluxograma, pseudocódigo e língua JAVA). Atividades de Acompanhamento: Participação durante as aulas; Exercícios e atividades autorais; Dinâmicas individuais e colaborativas; Projetos;	Componentes curriculares: Pensamento Computacional (com o apoio de jogos e blocos); Linguagem de Programação (com o apoio de jogos, blocos e pseudocódigo); Língua JAVA;

Fonte: Elaborada pelo autor

3.3 Etapa de Prototipação

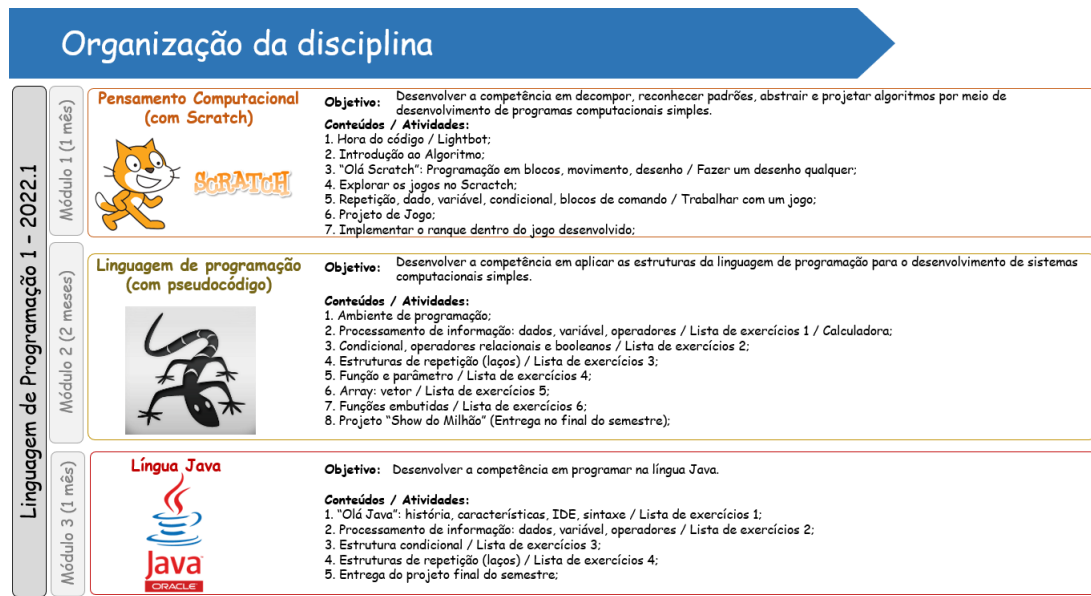
Após a fase de ideação finalizada, iniciamos a etapa de prototipagem. Nessa fase, propusemos uma nova estrutura para a disciplina, com base nos resultados obtidos anteriormente. O objetivo era proporcionar uma sequência lógica no aprendizado dos alunos, assim, a disciplina foi dividida em três módulos: Pensamento Computacional, Linguagem de Programação e Língua Java, conforme pode ser visualizado na Tabela 3. Este novo planejamento foi apresentado aos alunos no início do semestre, durante a aula de boas-vindas e discussão de expectativas. O objetivo era fornecer uma visão clara do caminho a ser percorrido na disciplina. Um formato visual foi utilizado para apresentar o planejamento, conforme ilustrado na Figura 22.

Tabela 3. Nova organização da disciplina LP1

MÓDULO	OBJETIVO	CONTEÚDO / ATIVIDADES
Pensamento Computacional	Desenvolver a competência em decompor, reconhecer padrões, abstrair e projetar algoritmos por meio de desenvolvimento de programas computacionais simples	1. Hora do código/Lightbot; 2. Introdução ao algoritmo; 3. “Olá Scratch”: Programação em blocos, movimento, desenho / Fazer um desenho qualquer; 4. Explorar os jogos no Scratch; 5. Repetição, dado, variável, condicional, blocos de comando; 6. Projeto de jogo; 7. Implementar o ranque dentro do jogo desenvolvido.
Linguagem de Programação	Desenvolver a competência em aplicar as estruturas de linguagem de programação para o desenvolvimento de sistemas computacionais simples	1. Ambiente de programação; 2. Processamento de informação: dados, variável, operadores / Lista de exercícios 1 / Calculadora; 3. Condicional, operadores relacionais e booleanos / Lista de exercícios 2; 4. Estruturas de repetição (laços) / Lista de exercícios 3; 5. Função e parâmetro / Lista de exercícios 4; 6. Array: vetor / Lista de exercícios 5; 7. Funções embutidas / Lista de exercícios 6; 8. Projeto Show do milhão (Entrega no final do semestre).
Língua JAVA	Desenvolver a competência em programar na língua JAVA	1. “Olá Java”: história, características, IDE, sintaxe / Lista de exercícios 1; 2. Processamento de informação: dados, variável, operadores / Lista de exercícios 2; 3. Estrutura condicional / Lista de exercícios 3; 4. Estruturas de repetição (laços) / Lista de exercícios 4; 5. Entrega do projeto final do semestre.

Fonte: elaborado pelo autor

Figura 22. Planejamento da disciplina LP1 apresentada aos alunos



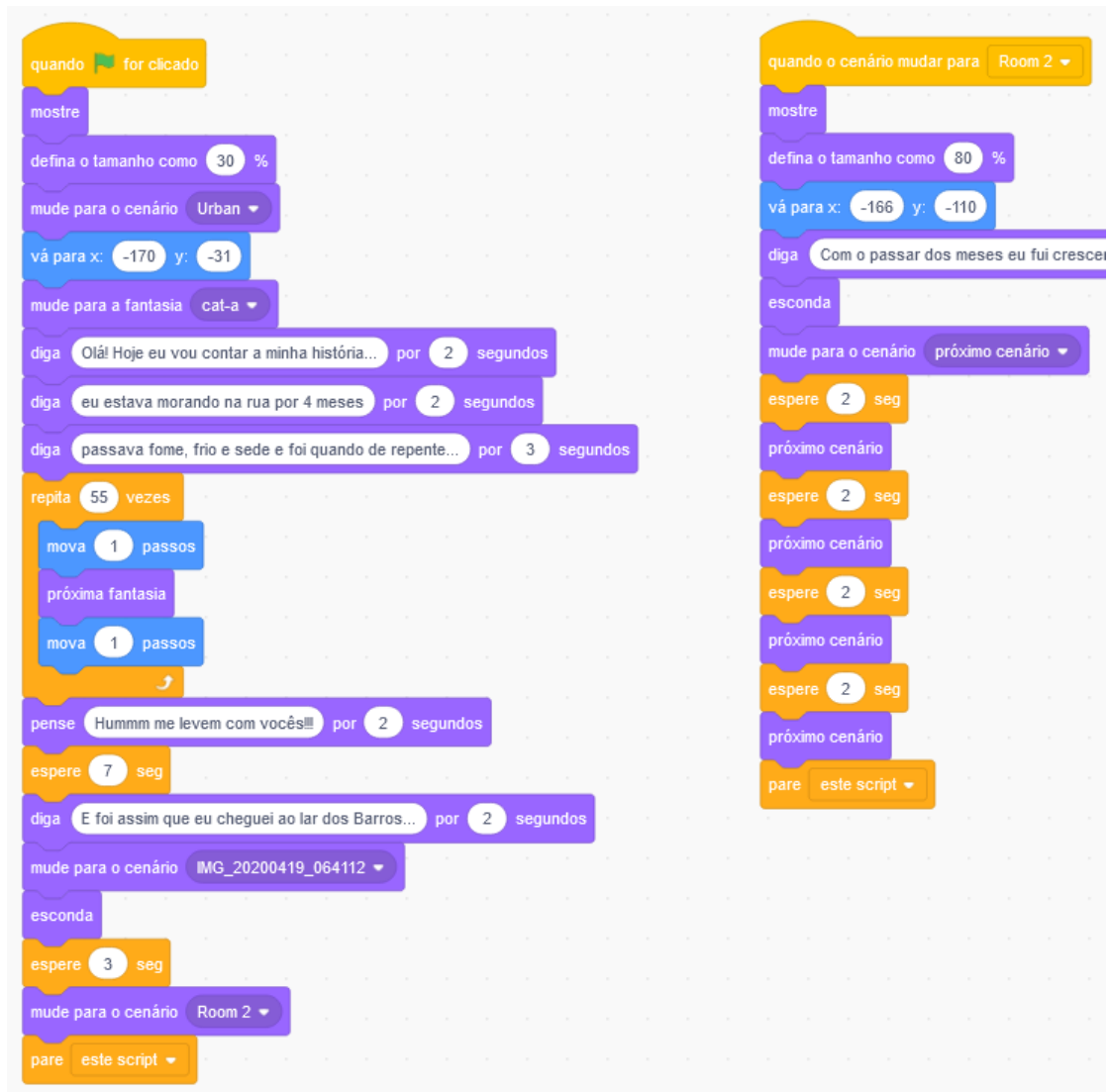
Fonte: Elaborada pelo autor

Durante o primeiro módulo, chamado Pensamento Computacional, com duração de um mês, o objetivo principal é desenvolver e/ou fortalecer as habilidades e conceitos do Pensamento Computacional. Para atingir esse objetivo, a principal ferramenta utilizada é o Scratch⁹, uma plataforma de programação visual baseada em blocos desenvolvida pelo grupo Lifelong Kindergarten no Media Lab da universidade americana MIT. O Scratch foi criado com o propósito de ensinar lógica de programação para crianças e adolescentes entre 8 e 16 anos de idade.

Essa plataforma utiliza uma língua de programação baseada em blocos, na qual cada bloco representa um comando específico. Ao encaixar os blocos em sequência, é possível criar programas de maneira intuitiva, como montar um quebra-cabeça. A Figura 23 ilustra um exemplo de códigos gerados no Scratch. Com o Scratch, os alunos têm a capacidade de criar animações, jogos e histórias interativas, permitindo a expressão de suas paixões e interesses pessoais por meio da programação em blocos.

⁹ <https://scratch.mit.edu/>

Figura 23. Exemplo de código gerado no Scratch



Fonte: Elaborada pelo autor

A grande vantagem do Scratch é que ele está disponível tanto *online*, permitindo o acesso de qualquer dispositivo conectado à internet, quanto *offline*, por meio de *download* no site oficial e instalação em computadores ou *tablets*. Isso possibilita que os alunos utilizem a ferramenta de programação em diversos ambientes, incluindo a sala de aula e em casa, o que facilita a prática e o aprimoramento de suas habilidades de programação.

O Scratch é uma ferramenta amplamente utilizada por pais que desejam ensinar programação aos seus filhos, bem como por professores do ensino fundamental e médio para introduzir a lógica de programação e o ensino de tecnologia. Além disso, o Scratch oferece diversas funcionalidades que são igualmente úteis para professores do ensino superior no ensino de programação.

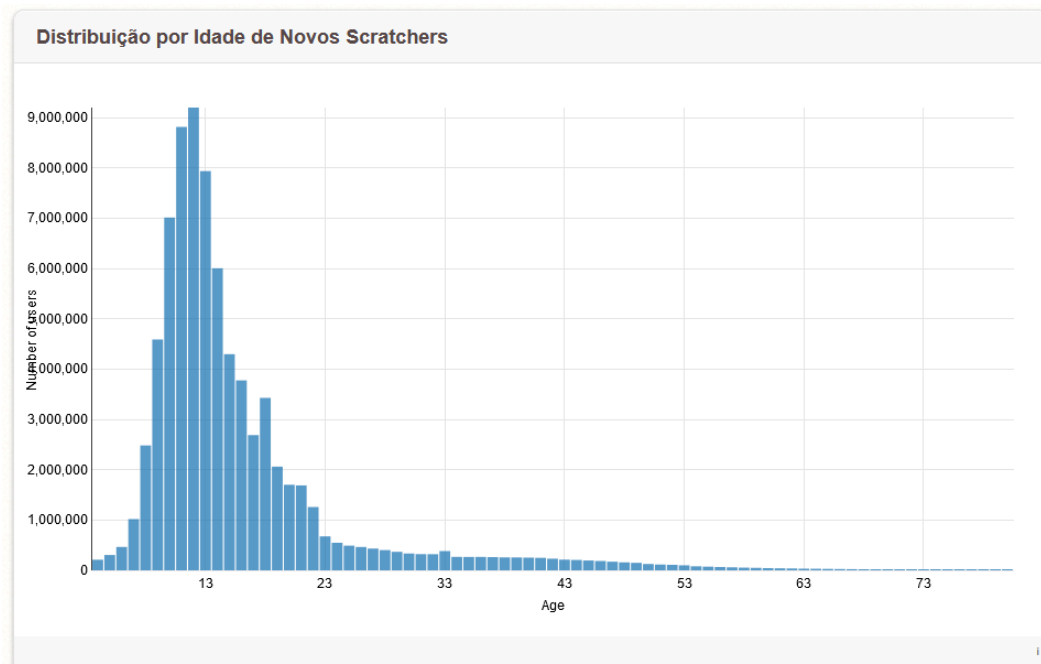
O Scratch é uma plataforma popular não apenas para uso educacional, mas também para pessoas que buscam aprender, se divertir e passar o tempo. Além disso, a plataforma

oferece recriações de jogos famosos, proporcionando uma experiência de entretenimento interativa e envolvente para seus usuários.

O site do Scratch¹⁰ fornece diversas estatísticas sobre a comunidade de usuários, incluindo a distribuição de idade dos participantes, conforme pode ser observado na Figura 24. Ao analisar o gráfico, é possível notar que a maioria dos usuários na plataforma tem entre 8 e 16 anos. Essa faixa etária é resultado do intenso uso do Scratch para o ensino de programação no ensino fundamental em todo o mundo.

No entanto, além do público mais jovem, também é notável a presença de um número significativo de usuários entre 16 e 22 anos. Esse segmento pode ser atribuído ao crescente número de professores universitários que adotam o Scratch como parte do processo de ensino de programação para estudantes do nível superior.

Figura 24. Distribuição dos usuários Scratch por idade



Fonte: Scratch Statistics - Imagine, Program, Share¹¹

Além do Scratch, o módulo conta com o apoio de outras duas plataformas de programação com blocos para o aprendizado de programação: A Hora do Código¹² e o Lightbot¹³.

A Hora do Código é um movimento global criado pela ONG Code.org, que tem como objetivo desmistificar a programação e mostrar aos alunos e professores como dar os primeiros passos nesse universo. O movimento alcança milhões de pessoas em todo o mundo, envolvendo-as em eventos de programação sem a exigência de experiência prévia

¹⁰ <https://scratch.mit.edu/statistics/>

¹¹ <https://scratch.mit.edu/statistics/>

¹² <https://hourofcode.com/br>

¹³ <https://www.lightbot.lu/>

em programação. Ele disponibiliza tutoriais em cerca de 45 idiomas para suas atividades, tornando-se acessível para um público diversificado. No Brasil, o Programaê!¹⁴ é o principal apoiador dessa iniciativa e busca mobilizar o maior número de participantes possível para experimentar, por pelo menos uma hora, o desafio chamado Desafio Hora do Código, que já ocorreu em 2014 e 2015.

Já a plataforma Lightbot, inspirada no conceito original da plataforma Lightbot.com¹⁵, está disponível *online* e pode ser acessada por qualquer dispositivo conectado à internet. Ela oferece aos usuários um jogo no qual é preciso dizer ao robô como acender todas as peças de luz do tabuleiro por meio de instruções, apresentadas no formato de blocos que o robô pode executar. A interface da plataforma é intuitiva, permitindo que os estudantes criem sequências de comandos facilmente, mesmo que o idioma seja inglês.

O módulo não se limita apenas às plataformas mencionadas, pois também inclui uma dinâmica chamada Estacionamento Algorítmico¹⁶, que tem como objetivo reforçar o Pensamento Computacional por meio do conceito de computação desplugada. A computação desplugada, também conhecida como “computação sem computador”, é um método de ensino que aborda conceitos de ciência da computação sem a necessidade de um computador ou dispositivo eletrônico. Esse método utiliza atividades práticas, jogos e outros recursos para ensinar habilidades de resolução de problemas, pensamento lógico, algoritmos e outras noções básicas de programação.

O objetivo principal é proporcionar aos alunos uma compreensão mais ampla da tecnologia e prepará-los para o mundo digital, mesmo em locais com recursos limitados ou sem acesso à tecnologia. A computação desplugada é uma abordagem versátil, podendo ser utilizada em qualquer faixa etária, desde a educação infantil até o ensino superior. Ela se mostra como uma alternativa interessante para complementar ou substituir o ensino tradicional de informática e tecnologia (ZANETTI; BORGES; RICARTE, 2016).

A dinâmica Estacionamento Algorítmico é um jogo de tabuleiro composto por vinte desafios, sendo dez de nível iniciante e dez de nível intermediário. Nesse jogo, os alunos devem utilizar uma sequência de comandos de movimentação para retirar o veículo X, que é representado pela cor vermelha, do estacionamento. Cada veículo pode ser movido um de cada vez, e os veículos estacionados na horizontal só podem ser deslocados para a direita ou esquerda, enquanto os veículos na vertical só podem ser movidos para cima ou para baixo.

Essa atividade proporciona uma forma divertida e prática de fortalecer o Pensamento Computacional por meio do conceito de computação desplugada. Ela ensina conceitos de ciência da computação sem a necessidade de um computador ou dispositivo

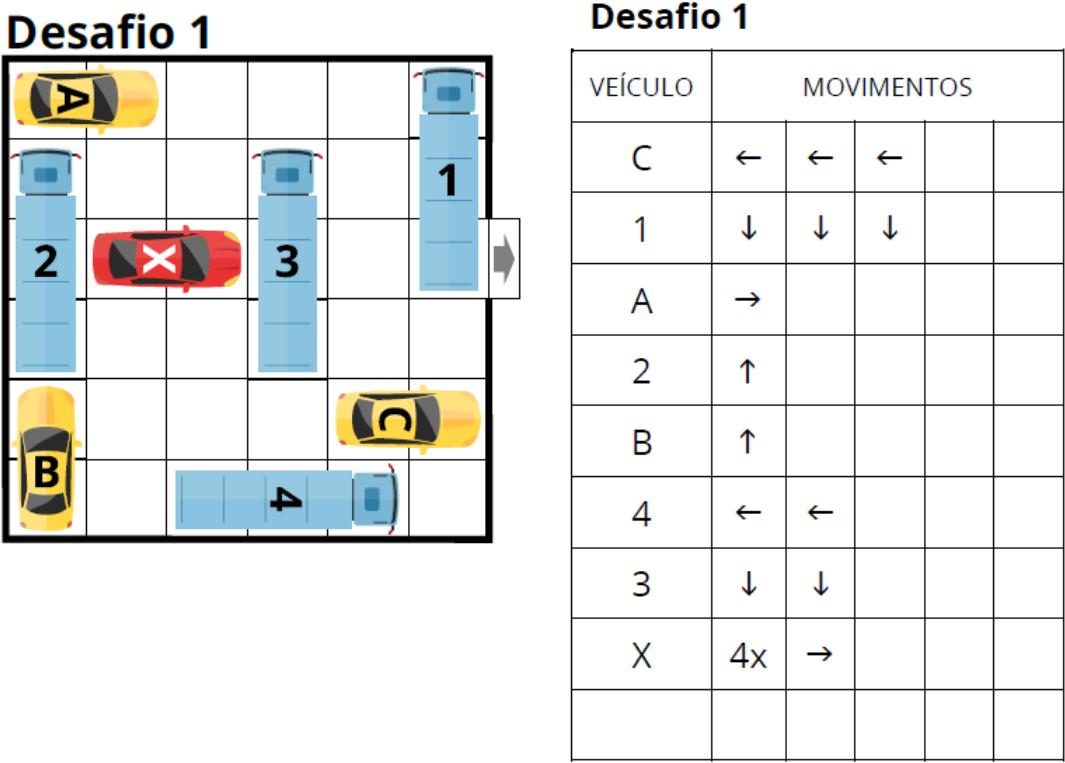
¹⁴ <https://programae.org.br/>

¹⁵ <https://lightbot.com/>

¹⁶ https://www.computacional.com.br/atividades/Estacionamento__Algoritmico.zip

eletrônico, permitindo que os alunos aprimorem suas habilidades de resolução de problemas, pensamento lógico e criação de algoritmos de maneira tangível e concreta.

Figura 25. Primeiro desafio da atividade Estacionamento Algorítmico



Fonte: Elaborada pelo autor

O primeiro desafio e seu respectivo gabarito do nível iniciante são demonstrados na Figura 25. Através dessa representação, os alunos replicam o desafio no tabuleiro do jogo. Após prepararem o tabuleiro, os estudantes identificam os comandos necessários para movimentar os outros veículos e liberar o caminho para o veículo X, registrando a sequência de comandos no gabarito através das setas indicativas. Ao completar o desafio, os alunos desenvolvem um algoritmo que resolve o problema apresentado, e a cada novo desafio, a complexidade da solução aumenta.

Todos os desafios, tabuleiro e gabaritos utilizados na dinâmica estão identificados no Apêndice C.

O segundo módulo, denominado Linguagem de Programação, tem duração de dois meses e tem como objetivo desenvolver a competência dos alunos em aplicar as estruturas da linguagem de programação para o desenvolvimento de sistemas computacionais simples. O conteúdo previsto para esse módulo inclui tópicos fundamentais, tais como dados, variáveis, operadores, estruturas de sequência, estruturas de condição, estruturas de repetição, funções, parâmetros e *arrays*.

Para apoiar o aprendizado, foi selecionado o interpretador de pseudocódigo Calango, uma ferramenta gratuita desenvolvida em Java. O Calango é um interpretador educacional

de pseudocódigo em língua portuguesa, projetado para auxiliar na construção, depuração e manutenção de algoritmos. Essa plataforma é multiplataforma, o que permite aos alunos compreenderem de forma mais concreta como um programa seria executado. De acordo com Silva, Santos e Rissoli (2020), o Calango tem o propósito de apoiar a aprendizagem inicial em programação de computadores, buscando atingir seus objetivos educacionais. Algumas características identificadas no Calango incluem:

- Maior facilidade de aprendizagem de programação para estudantes que são nativos da língua portuguesa;
- Uso de representação mais simples em algoritmos que pode solucionar problemas propostos usando a língua que o estudante domina;
- Execução do algoritmo pelo computador com orientações na língua do estudante;
- Depuração do pseudocódigo em tempo real, sendo executado passo a passo e apresentando ao estudante a realidade momentânea de cada valor armazenado na memória, semelhante ao teste de mesa em programação;
- Autonomia para estudantes novatos em programação, possibilitando um processo de aprendizagem contínua e significativa que considera o ritmo individual de cada aluno;
- Assistência na depuração do pseudocódigo, verificando possíveis erros ou depurando o algoritmo para aprimorá-lo.

O pseudocódigo utilizado pelo Calango foi inspirado nas características estruturais da língua C, uma vez que essa língua é amplamente reconhecida e utilizada como base para a introdução ao ensino de programação (EZENWOYE, 2018). Além dos operadores e comentários semelhantes aos disponíveis na língua C, o Calango também utiliza uma notação mais instrutiva para a marcação de blocos de processamento.

Por fim, o terceiro módulo, chamado de Língua Java, tem duração de um mês e tem como objetivo desenvolver as habilidades básicas dos estudantes para programar na língua Java, com base nos conceitos apresentados no módulo de Linguagem de Programação. Esse módulo tem como propósito preparar os estudantes para a continuidade do curso, que adota o Java como língua de programação padrão em sua estrutura curricular. Os alunos são introduzidos à sintaxe do Java e aprendem a escrever códigos que resolvam os problemas apresentados.

Embora o Java seja uma língua orientada a objetos, o professor pode “traduzir” o pseudocódigo gerado no Calango para um código escrito em Java, facilitando o aprendizado dos alunos. Essa abordagem de transição entre o pseudocódigo e a língua Java é

especialmente útil para os estudantes, pois lhes permite associar os conceitos aprendidos no módulo anterior com a prática da programação real na língua Java.

A (re)organização da disciplina em três módulos busca proporcionar uma sequência lógica de aprendizado de programação aos estudantes, oferecendo uma abordagem progressiva e gradual para o desenvolvimento de suas habilidades. O primeiro módulo introduz o aluno ao universo da programação por meio da programação em blocos com o Scratch, permitindo que eles criem jogos, histórias e animações de forma interativa e divertida.

Em seguida, o segundo módulo apresenta o pseudocódigo no Calango, onde os alunos podem traduzir os códigos criados no Scratch para uma língua mais próxima da programação convencional. Essa transição ajuda a consolidar os conceitos de programação e a compreender melhor as estruturas lógicas envolvidas na criação de algoritmos.

Por fim, no terceiro módulo, os códigos do Calango são traduzidos para a língua Java, proporcionando aos estudantes a experiência de codificar em uma língua de programação utilizada no desenvolvimento de sistemas computacionais. Nesse estágio, os alunos têm a oportunidade de experimentar a criação de programas de computador funcionais, consolidando ainda mais seus conhecimentos e habilidades em programação.

Essa progressão pedagógica, que vai desde a programação em blocos até a codificação em Java, permite que os estudantes desenvolvam suas competências de forma contínua e estruturada, construindo uma base sólida para sua futura atuação na área de tecnologia da informação.

No contexto de avaliação dos estudantes, a nova organização da disciplina prevê uma avaliação para cada um dos módulos. No primeiro módulo, os estudantes serão avaliados pela elaboração de um projeto de jogo autoral no Scratch. Nessa avaliação, eles terão a oportunidade de demonstrar suas habilidades de programação em blocos, criando um jogo original e explorando as funcionalidades do Scratch de forma criativa.

Na segunda avaliação, os alunos deverão construir um jogo autoral no Calango, utilizando o pseudocódigo aprendido durante o módulo de Linguagem de Programação. Nessa etapa, eles devem aplicar os conceitos aprendidos e traduzir suas ideias do Scratch para o Calango, consolidando os conhecimentos adquiridos anteriormente.

Por fim, na terceira avaliação, os estudantes deverão criar uma versão do mesmo jogo desenvolvida em Java. Essa avaliação é uma oportunidade para os alunos aplicarem os conceitos aprendidos durante o módulo de Língua Java, transformando seus códigos do Calango em um programa funcional em uma língua de programação.

Com a reorganização da disciplina, é possível observar como os princípios orientadores resultaram em novas práticas docentes ou adaptações das já existentes. A aprendizagem colaborativa promoveu aulas menos expositivas e mais participativas, onde os alunos são incentivados a debater o conteúdo em sala de aula, estimulando a troca de conhecimentos

e a construção coletiva do aprendizado.

A programação em pares resultou em um novo formato de sala de aula, em que os alunos são encorajados a trabalhar em duplas na elaboração de códigos, seja em listas de exercícios ou em projetos menores. Essa abordagem promove o trabalho em equipe, a comunicação e a resolução colaborativa de problemas, além de proporcionar aos alunos uma oportunidade de aprender com seus colegas.

A aprendizagem baseada em projetos e autoria afetou o formato de avaliação, substituindo testes e provas por projetos autorais em todos os módulos da disciplina. Os alunos têm a liberdade de criar um projeto baseado em suas experiências e vivências, o que torna a avaliação mais significativa e relevante para eles. Isso também estimula a criatividade e o interesse dos alunos, pois eles têm a oportunidade de aplicar os conhecimentos adquiridos em projetos práticos e relevantes.

A ludicidade é incorporada nas atividades da disciplina, tanto com o uso do computador quanto sem ele. O uso de jogos digitais para apoiar o ensino de programação levou à adoção de plataformas que usam programação em blocos como parte do aprendizado dos alunos. Isso permite que os alunos experimentem resultados visíveis e imediatos antes de serem inseridos em um contexto abstrato, tornando o aprendizado mais concreto e motivador.

O letramento e o design levaram o professor a revisar completamente seu material de apoio, simplificando a documentação disponibilizada para os estudantes e mesclando as listas de exercícios entre problemas matemáticos e problemas do cotidiano dos alunos. Essa abordagem torna o conteúdo mais acessível e relevante para os alunos, relacionando-o com suas experiências diárias e tornando-o mais compreensível e aplicável.

Por fim, o pensamento criativo permeia grande parte da disciplina, incentivando os alunos a fortalecer sua criatividade e resultar em projetos mais interessantes. Isso estimula a imaginação dos alunos, encorajando-os a encontrar soluções originais para os problemas apresentados e a desenvolver habilidades de inovação e pensamento “fora da caixa”.

Com todas essas mudanças e incorporações, a disciplina de Linguagem de Programação 1 se tornou mais dinâmica, envolvente e relevante para os alunos, proporcionando uma experiência de aprendizado mais significativa e preparando-os para o mundo da programação de forma mais abrangente e eficaz.

Depois de todo o processo percorrido na fase de prototipagem, que envolveu a elaboração de uma estratégia de planejamento e organização da disciplina, o resultado foi um novo plano didático para a disciplina. O plano está disponível no Apêndice B, onde pode ser acessado e utilizado como referência para futuras implementações da disciplina Linguagem de Programação 1. O novo plano representa o produto final do processo de reestruturação e incorpora todas as inovações e melhorias desenvolvidas ao longo da

pesquisa, visando proporcionar uma experiência de aprendizado mais eficaz e significativa para os estudantes.

3.4 Etapa de Testes

Após a conclusão da fase de prototipação e a elaboração do plano didático da disciplina, a fase de testes foi iniciada. Esta etapa tem como objetivo testar a solução proposta com os usuários finais, ou seja, os alunos. Diferentes dispositivos serão utilizados para avaliar a eficácia da nova abordagem.

O primeiro dispositivo é a observação da turma durante as aulas, incluindo o comportamento dos estudantes e as dificuldades encontradas ao longo do semestre. O Diário de Pesquisa do Professor será utilizado para registrar todos os marcos e acontecimentos observados ao longo do período letivo.

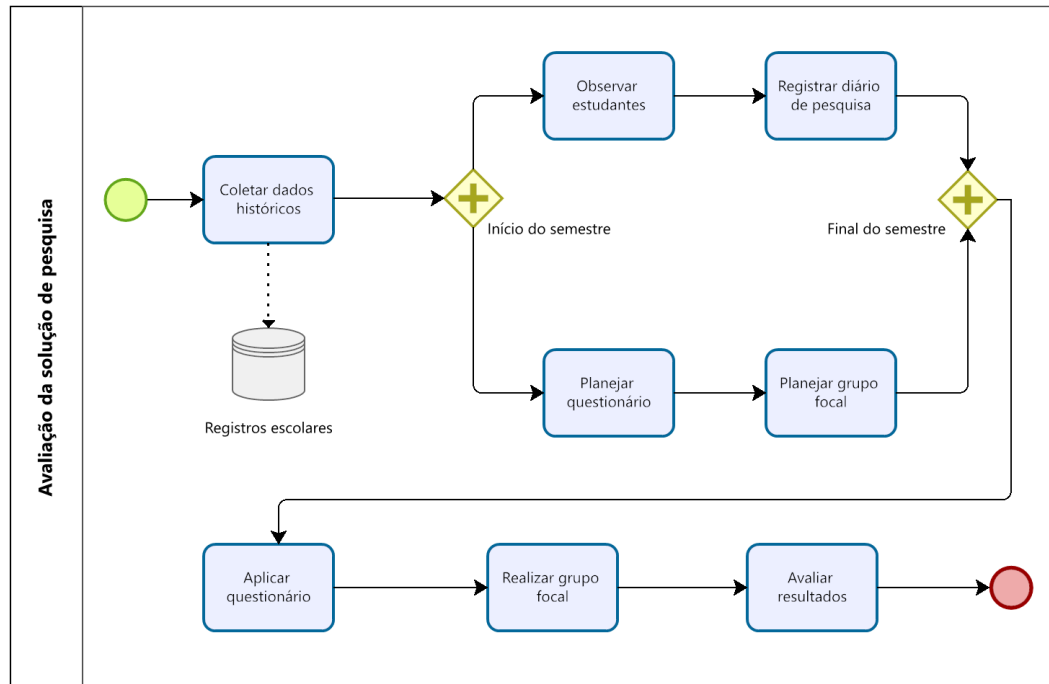
Além disso, será aplicado um questionário de opinião aos estudantes que cursarem a disciplina, com o intuito de coletar suas percepções e experiências em relação à nova organização da disciplina e aos métodos de ensino utilizados. Também serão coletados *feedbacks* dos alunos por meio de um grupo focal, uma técnica que permite aprofundar e obter informações valiosas sobre a experiência dos alunos.

Com a combinação desses dispositivos de avaliação, espera-se obter dados abrangentes e relevantes sobre o impacto da nova abordagem no aprendizado dos alunos e na sua percepção sobre a disciplina. Os resultados desses testes serão analisados e utilizados para aprimorar ainda mais o plano didático, tornando-o cada vez mais efetivo e alinhado às necessidades dos estudantes.

É importante destacar que a avaliação da solução será composta por insumos fornecidos tanto pela observação do pesquisador quanto pela opinião e vivência dos alunos durante o semestre letivo. Dessa forma, os resultados deverão expressar a visão tanto do pesquisador quanto dos alunos em relação à nova abordagem proposta para a disciplina LP1.

A Figura 26 ilustra graficamente como será realizada a avaliação da solução proposta. Inicialmente, foram coletados dados históricos de evasão e reprovação na disciplina de LP1 na unidade escolar nos anos de 2018 e 2019, excluindo o período da pandemia da covid-19, em que foi aplicada a modalidade de ensino remoto emergencial. Durante o semestre letivo, os dispositivos de observação e o diário de pesquisa serão aplicados continuamente, até o encerramento do semestre. Ao final do semestre, será realizado um questionário *online* e um grupo focal com os estudantes, gerando os insumos necessários para a avaliação da solução proposta.

Figura 26. Avaliação da solução proposta na pesquisa



Fonte: Elaborada pelo autor

4 QUADRO

EPISTÊMICO-

METODOLÓGICO

Nesta seção, nosso objetivo é apresentar a metodologia adotada na pesquisa, denominada Pesquisa-Formação, os dispositivos utilizados para a produção de dados, que resultaram em informações tanto quantitativas quanto qualitativas e descrever o cotidiano da pesquisa. Foram utilizados os seguintes dispositivos: Diário de Pesquisa, Questionário *online*, Diário de Classe e Grupo Focal.

4.1 Método Pesquisa-Formação

Esta pesquisa foi guiada pelo método denominado Pesquisa-Formação, já bastante consolidado em pesquisas científicas no campo Educacional. Toda Pesquisa-Formação parte dos dilemas docentes, ou seja, das inquietações e problemáticas advindas da prática e da docência do professor-pesquisador. Apesar de ser um método consolidado, Santos (2019) propõe uma atualização para o contexto da docência na cibercultura, que se fundamenta e caracteriza em função de algumas especificidades:

1. A cibercultura permitiu o surgimento de novos arranjos espaço-temporais e, conseqüentemente, novas práticas educativas. Pesquisar sem a imersão nas práticas da cibercultura não é possível, já que ela faz parte do contexto atual.
2. Pesquisar na cibercultura não se resume apenas ao uso de *softwares* para coletar e organizar dados, mas sim a atuar como praticante cultural e produzir dados em rede.
3. Não há Pesquisa-Formação desvinculada do contexto da docência.

A Pesquisa-Formação na cibercultura pressupõe que o docente-pesquisador participe da formação dos estudantes, onde estes estudam, interagem, produzem e criam conhecimentos em rede, e, ao mesmo tempo, transformam as práticas do pesquisador durante o processo.

No campo Educacional, a investigação da própria prática docente é uma abordagem para produzir conhecimentos científicos sobre aspectos didático-pedagógicos amplamente reconhecida e conhecida. Paulo Freire (2014) afirmava que ensinar é ser epistemologicamente curioso e isso implica participar na construção do conhecimento do objeto. Assim, um professor pesquisador pode ser considerado aquele que aprende enquanto ensina e ensina enquanto aprende. Ser curioso, como afirmava Freire, está relacionado à habilidade de aprender.

Desta forma, o pesquisador não é apenas o sujeito que constata o que ocorre, mas também deve participar ativamente das ocorrências. Assim, no contexto da prática da Pesquisa-Formação como método de produção de conhecimento, reflete-se sobre os problemas vividos pelo pesquisador em sua ação como docente, possibilitando mudanças nas práticas e nos sujeitos em formação (docente e estudantes). Conforme afirmou António Nóvoa, “a pessoa é, simultaneamente, objeto e sujeito da formação” (NÓVOA, 2004, p. 15).

A Pesquisa-Formação proporciona experiências onde são adotados ambientes e dispositivos de pesquisa que permitem o registro e a expressão de narrativas e imagens.

O dispositivo no método de Pesquisa-Formação é um conjunto de elementos que têm como objetivo facilitar a realização da pesquisa e a formação dos participantes. Esses elementos englobam técnicas, instrumentos, materiais didáticos, atividades e estratégias de ensino, entre outros. O dispositivo é uma ferramenta crucial para realizar a pesquisa e a formação, promovendo a troca de experiências, a construção conjunta do conhecimento e a transformação das práticas educacionais. São exemplos de dispositivos na Pesquisa-Formação uma atividade ou situação de aprendizagem elaborada pelo professor para investigar algo; uma ambiência computacional para que estudantes e docente possam vivenciar o uso de uma tecnologia inovadora e juntos possam produzir conhecimentos sobre ela; todo um curso, desenhado de tal maneira que possa ser o disparador de narrativas, imagens e sons que serão os rastros para a investigação desejada, entre diversos outros dispositivos materiais ou intelectuais que possibilitam disparar a produção de dados que irão possibilitar o pesquisador/a-professor/a investigar em parceria com as/os estudantes o objeto de investigação.

Durante o primeiro semestre de 2022, o novo plano da disciplina LP1, criado com base nos fundamentos e ideias estudados, foi utilizado como dispositivo nesta pesquisa. O dispositivo foi aplicado no cotidiano dos estudantes e do pesquisador na disciplina Linguagem de Programação 1, do curso técnico em Informática da Escola Técnica Estadual Maria Mercedes Mendes Teixeira. O objetivo da pesquisa não é apenas contribuir com a formação técnica dos envolvidos, mas também contribuir para a produção de conhecimentos sobre a arte de ensinar-aprender programação, estabelecendo interfaces com as áreas da Educação, Comunicação e Computação. O novo plano da disciplina LP1 está disponível no Apêndice B.

A Pesquisa-Formação foi escolhida como abordagem epistêmico-metodológica orientadora desta pesquisa, uma vez que integra pesquisa e formação de maneira colaborativa, possibilitando a produção de conhecimento de forma compartilhada entre pesquisadores e participantes envolvidos no processo. Portanto, o método Pesquisa-Formação é uma opção relevante para aqueles que desejam desenvolver práticas educativas mais participativas, engajadoras e críticas.

Esta pesquisa surge como uma tentativa de enfrentar e resolver os desafios e

conflitos encontrados pelo pesquisador em sua prática como professor. O método de Pesquisa-Formação é recomendado para estudos conduzidos por professores e está alinhado com os objetivos desta pesquisa. Os principais desafios enfrentados pelo pesquisador incluem altos índices de reprovação e evasão, dificuldades de aprendizagem dos alunos em programação, falta de motivação dos estudantes na disciplina e práticas descontextualizadas da realidade e dos interesses dos estudantes. Esses dilemas justificam a escolha do método de Pesquisa-Formação nesta investigação. Desde o início da minha trajetória como professor, o objetivo tem sido fornecer aos alunos conhecimentos e instrumentos que possam apoiá-los em suas carreiras profissionais, considerando que a disciplina faz parte de um curso técnico. No entanto, os altos índices de reprovação e evasão na disciplina de Linguagem de Programação 1 (LP1) podem ser atribuídos à complexidade exigida ao entrar no universo da programação, que requer o desenvolvimento de pensamento, língua e diversas habilidades diferentes das que estamos acostumados. Portanto, é fundamental repensar as nossas práticas docentes para lidar com os dilemas enfrentados pelo pesquisador.

4.2 Dispositivos de Pesquisa

Os pesquisadores são incentivados a expressar suas itinerâncias formativas, possibilitando em muitos casos o compartilhamento com outros envolvidos no processo. É comum encontrar como dispositivos para registro do pesquisador o diário de bordo ou itinerância, os memoriais de pesquisa e prática profissional, conversas, entrevistas abertas, entre outros.

4.2.1 Diário de Pesquisa

Existem diversas nomenclaturas que podem ser utilizadas para se referir a registros escritos durante uma pesquisa, tais como diário de campo, diário de bordo, jornal de pesquisa, diário *online*, diário de itinerâncias, diário de pesquisa, entre outros. Essas diferentes terminologias são apenas algumas das opções disponíveis para indicar esse tipo de registro. Um **diário de pesquisa** é um dispositivo que os pesquisadores utilizam para registrar suas reflexões, observações e sentimentos durante a realização da pesquisa e produção de dados. É útil para registrar perspectivas sobre os resultados percebidos e pode ser utilizada como fonte de informação. O diário de pesquisa também ajuda o pesquisador a manter um registro sistemático de suas experiências e aprendizados durante a pesquisa. É possível fazer o diário em formato digital ou físico e é importante atualizá-lo regularmente ao longo da pesquisa.

Desse modo, consiste na escrita contínua de fatos, encontros, leituras, reflexões, estudos, entre outros, organizados em torno das experiências vividas em ‘espacostempos’ diversos, seja no trabalho, na conjugalidade, na relação com o Outro,

ou consigo mesmo, propiciando: (a) a formação para a pesquisa, mediante diálogo com as questões que emergem no campo; (b) a formação para a escrita, a partir do exercício contínuo do registro das narrativas e, (c) principalmente, a formação autoral no contexto da pesquisa. Nessa perspectiva, carrega uma dimensão pedagógica, pois consiste, a um só tempo, num dispositivo e num processo, dado que “formar não é instruir... É primeiro refletir, é pensar uma experiência vivida”(AMARAL; SANTOS; SANTOS, 2022, p. 8)

O diário de pesquisa é um dispositivo importante para o pesquisador, pois possibilita a organização, interpretação e reflexão sobre seus dados de forma mais sistemática e abrangente. Alguns benefícios da utilização do diário de pesquisa incluem:

1. Registro sistemático: permite que o pesquisador mantenha um registro organizado e sistemático de suas observações, reflexões e percepções durante a pesquisa.
2. Fonte de informação: fornece uma fonte valiosa de informações que podem ser usadas para analisar e interpretar os dados.
3. Reforço da memória: ajuda o pesquisador a relembrar detalhes importantes que podem ter sido esquecidos ao longo da pesquisa.
4. Melhoria do planejamento: possibilita que o pesquisador planeje e ajuste a estratégia de produção dos dados com base em suas reflexões registradas.
5. Auxílio na análise: o diário de pesquisa pode fornecer informações valiosas para a interpretação dos dados, permitindo que o pesquisador identifique padrões e tendências.

4.2.2 Questionário *online*

Um questionário é um instrumento amplamente utilizado em pesquisas científicas para coletar dados de forma padronizada e quantitativa. Consiste em um conjunto de perguntas, questões ou itens pré-definidos, que têm o objetivo de determinar atributos ou características relacionadas a pessoas, organizações ou processos. Atualmente, é cada vez mais comum que os pesquisadores optem por utilizar tecnologias que permitem a elaboração de **questionários online**. Plataformas como o Google Forms, Microsoft Forms e LimeSurvey possibilitam a criação de questionários que podem ser respondidos eletronicamente pelos participantes, facilitando a coleta de dados.

A validade dos resultados de uma pesquisa depende, em última análise, da qualidade das medições realizadas pelos pesquisadores. Se o questionário foi mal elaborado, os resultados da pesquisa podem ser considerados inválidos ou não confiáveis. É essencial

que o questionário seja cuidadosamente desenvolvido, com perguntas claras, relevantes e adequadas aos objetivos da pesquisa, a fim de garantir a confiabilidade e a validade dos dados coletados. Um questionário bem elaborado aumenta a probabilidade de obter resultados precisos e significativos, contribuindo para a qualidade da pesquisa como um todo.

O processo de elaboração de um questionário é uma etapa crucial que requer atenção e não deve ser encarado como um processo único de inspiração. É necessário dedicar tempo e esforço durante a pesquisa, incluindo momentos de revisão e aprimoramento. Para auxiliar nesse processo, Coelho, Souza e Albuquerque (2020) sugerem a utilização de um *checklist* com processos básicos para a construção de um questionário. Seguir essas diretrizes pode contribuir para a qualidade e eficácia do instrumento de coleta de dados, aumentando a confiabilidade e validade dos resultados obtidos.

1. Confirme os objetivos da pesquisa e as informações necessárias a serem coletadas antes de planejar o questionário.
2. Determine os requisitos de dados (métodos de coleta) para alcançar cada objetivo da pesquisa.
3. Na introdução do questionário, devem-se incluir uma descrição geral do estudo e um termo de consentimento livre e esclarecido.
4. Em seguida, devem ser expressas com clareza as instruções de resposta ao questionário.
5. As questões e/ou itens devem seguir uma ordem lógica e encadeada, por exemplo: (i) questões de qualificação ou triagem, (ii) questões introdutórias, (iii) questões principais, (iv) questões psicográficas, (v) questões demográficas e (vi) questões de identificação.
6. Escolha o texto e a melhor estrutura para cada questão. Verifique o conteúdo individual de cada questão.
7. Use uma sequência temática ou seções, por exemplo: (i) questões sobre satisfação, (ii) questões atitudinais, (iii) questões socio-bio-demográficas. Inicie o questionário com as questões ou grupo de questões mais simples, avançando gradualmente em direção às questões ou grupo de questões mais complexas.
8. Determine o formato e layout final do questionário. Evite elaborar questões com diferentes formatos de mensuração na mesma seção do questionário.
9. Finalize o questionário com um agradecimento.
10. Realize um pré-teste.

Outro aspecto de grande importância ao qual o pesquisador precisa estar atento diz respeito aos critérios para a redação das perguntas ou itens do questionário. Coelho, Souza e Albuquerque (2020, p. 14-15) apresentam 10 critérios que contribuem para tornar o questionário robusto. São eles: unicidade, objetividade, simplicidade, clareza, relevância, precisão, variedade, modalidade, tipicidade e credibilidade. Seguir esses critérios durante a elaboração das perguntas ou itens do questionário auxilia na obtenção de dados confiáveis, consistentes e relevantes, garantindo a qualidade e a validade da pesquisa.

A elaboração de um questionário deve levar em consideração o tipo de resultado que se pretende alcançar. Se o objetivo do pesquisador é registrar dados quantitativos, como medidas de tamanho, extensão, volume ou quantidade, as perguntas devem ser formuladas de forma fechada, oferecendo opções de resposta predefinidas. Por outro lado, se o resultado desejado é registrar informações qualitativas, como subjetividades, sensações, reações e julgamentos individuais, as perguntas devem ser formuladas de forma aberta, permitindo que os participantes expressem suas opiniões e experiências em suas próprias palavras. É comum que um questionário apresente uma combinação de questões dos dois tipos, visando atingir uma variedade de dados e perspectivas.

Com o questionário definido, o pesquisador precisa determinar a abordagem de análise dos dados. Se a pergunta for do tipo fechada, a análise será conduzida de forma quantitativa, utilizando técnicas estatísticas como Frequência, Média, Desvio Padrão, Correlação, Regressão, Análise Fatorial, entre outras. Por outro lado, se a pergunta for do tipo aberta, a análise será realizada de forma qualitativa, utilizando métodos como a análise de conteúdo (por exemplo, o modelo de Bardin) ou a análise de conteúdo informatizada utilizando *softwares* como o Iramuteq (com base no modelo de Reinert), por exemplo. A escolha da abordagem de análise dependerá da natureza dos dados produzidos e dos objetivos da pesquisa.

Durante a realização de uma pesquisa acadêmica, o pesquisador pode empregar uma variedade de instrumentos metodológicos. Um desses instrumentos é a pesquisa documental, que envolve a investigação ou análise de documentos relevantes para o estudo em questão. A utilização de documentos em uma pesquisa pode proporcionar uma ampla gama de informações, permitindo uma compreensão mais aprofundada do objeto de estudo em sua contextualização histórica e sociocultural. Isso enriquece o conhecimento e contribui para uma análise mais abrangente e fundamentada. (SÁ-SILVA et al., 2009)

A pesquisa documental e a pesquisa bibliográfica possuem semelhanças, mas apresentam diferenças significativas. Enquanto a pesquisa bibliográfica se concentra em fontes científicas e materiais já publicados, como livros, artigos e periódicos, a pesquisa documental abrange materiais não publicados, como memorandos, diários, relatórios e outros documentos não editados. Portanto, a principal distinção entre esses tipos de pesquisa reside na natureza das fontes utilizadas.

No campo da Educação, existem várias fontes documentais que podem ser utilizadas em pesquisas, como o diário de classe, o plano de ensino e o Projeto Político-Pedagógico (PPP), entre outros. Esses documentos fornecem informações importantes sobre a prática pedagógica, a organização curricular e as políticas educacionais adotadas em determinado contexto.

No estudo de Cechinel et al. (2016), é apresentada uma sugestão de processo documental que começa com a avaliação preliminar de cada documento, realizando uma análise crítica por meio da consideração dos seguintes elementos: contexto, autores, interesses, confiabilidade, natureza do texto e conceitos-chave. Essa abordagem visa examinar e criticar cada documento, levando em conta esses aspectos, a fim de compreender sua relevância e confiabilidade para a pesquisa em questão.

No âmbito desta pesquisa, a pesquisa documental utilizou os seguintes dispositivos: o diário de classe, os projetos submetidos pelos estudantes e o histórico de notas da disciplina no curso. Esses documentos foram analisados e considerados como fontes de informações relevantes para a investigação em questão.

4.2.3 Diário de classe

O **diário de classe** é um registro utilizado pelos professores para acompanhar o progresso e o desempenho dos alunos em uma disciplina ou curso específico. Nele, são registradas informações como presença dos alunos nas aulas, notas, tarefas e trabalhos realizados, além de observações sobre comportamento e desempenho em sala de aula. O diário de classe é uma ferramenta essencial para avaliar e monitorar o desempenho dos alunos, auxiliando os professores no planejamento de suas aulas e atividades de acordo com as necessidades e interesses dos alunos. Também é uma fonte útil de informação para os pais ou responsáveis, permitindo que acompanhem o progresso acadêmico de seus filhos.

4.2.4 Projeto de Avaliação

Um **projeto de avaliação** entregue pelos estudantes refere-se a um trabalho desenvolvido por eles como parte de uma atividade de avaliação em um curso ou disciplina. Geralmente, os estudantes recebem uma tarefa ou problema para resolver, e eles precisam aplicar seus conhecimentos, habilidades e criatividade para desenvolver um projeto que aborde a questão proposta. Através do projeto de avaliação, os estudantes têm a oportunidade de demonstrar seu entendimento do conteúdo estudado, aplicar conceitos teóricos na prática, desenvolver habilidades de pesquisa e análise, exercitar a capacidade de solucionar problemas e promover a autonomia e o pensamento crítico. Para os pesquisadores, os projetos de avaliação entregues pelos estudantes podem fornecer percepções valiosas sobre o processo de aprendizagem, permitindo observar a aplicação dos conhecimentos na prática,

identificar pontos fortes e fracos dos alunos, e avaliar a eficácia das estratégias de ensino utilizadas.

4.2.5 Histórico de notas

O **histórico de notas** de uma disciplina é um registro que contém informações sobre as notas ou avaliações obtidas pelos estudantes ao longo do tempo em uma determinada disciplina ou curso. O histórico de notas geralmente é mantido pela instituição educacional e serve como um registro formal do desempenho acadêmico dos estudantes ao longo do período em que a disciplina é ministrada. O histórico de notas também pode ser utilizado como base para a análise de dados acadêmicos, a identificação de tendências ou padrões de desempenho dos alunos e a elaboração de relatórios estatísticos sobre os resultados obtidos na disciplina ao longo do tempo.

4.2.6 Grupo Focal

O **Grupo Focal** é uma técnica de coleta de dados utilizada comumente em pesquisas qualitativas, especialmente na área de Informática na Educação. Uma das principais características dessa técnica é a possibilidade dos participantes dialogarem sobre o(s) assunto(s) da pesquisa, com o objetivo de estabelecer um consenso. Entretanto, é importante ressaltar que nem sempre é possível chegar a um consenso.

“Toda a vez que se estabelece um diálogo, como ocorre no Grupo Focal, mesmo que não enunciada há uma expectativa de que se estabeleçam o acordo, a conformidade de opiniões, ideias, sentimentos ou impressões, ou seja, o consenso, sem a aplicação de procedimentos coercitivos.” (OLIVEIRA; VASCONCELLOS, 2021, p. 4)

Após conceituarmos diálogo e consenso, é importante aprofundarmos no que é o Grupo Focal. Pode-se dizer que o Grupo Focal é uma técnica originada de trabalhos em grupos pela Psicologia Social, com o objetivo de coletar material de maneira interativa ou colaborativa, debatendo sobre um determinado assunto, a partir de um roteiro pré-estabelecido. O material coletado deve envolver conceitos, sentimentos, atitudes, crenças, experiências e reações dos participantes.

Para alguns autores, pode ser considerado uma espécie de entrevista em grupo, porém, o Grupo Focal se caracteriza pela utilização de temas ou estímulos apresentados pelo moderador, que pode ser o próprio pesquisador. Durante a sessão, além do moderador, podem participar observadores e relatores, responsáveis pelo registro das informações. Com a conclusão da sessão, o conteúdo da discussão é transcrito e analisado de acordo com os objetivos da pesquisa.

Os participantes selecionados precisam ter características comuns, como por exemplo, serem estudantes da mesma instituição ou do mesmo nível de ensino, mas também precisam ser suficientemente diferentes para favorecer o debate. O moderador não precisa atuar apenas como ouvinte do grupo, pois é responsável por propor encaminhamentos a partir do tema e interferir, quando necessário, para auxiliar as trocas e manter o foco da discussão de acordo com os objetivos da pesquisa.

Uma das principais vantagens da adoção do Grupo Focal é permitir a coleta de respostas mais amplas e completas em relação ao tema discutido. Além disso, possibilita a compreensão das ideias compartilhadas pelos participantes e como podem se influenciar mutuamente, em vez de focar apenas em opiniões pessoais.

A utilização do Grupo Focal pode ocorrer em diferentes etapas da pesquisa, de acordo com o objetivo estabelecido pelo pesquisador. Entre as vantagens, podemos destacar sua utilização como busca do aperfeiçoamento da compreensão, com o detalhamento de dados oriundos de outras técnicas. De acordo com Oliveira e Vasconcellos (2021), as vantagens da utilização do Grupo Focal podem ser resumidas em:

1. O Grupo focal permite aos participantes, através da troca de opiniões e vivências, tomarem consciência das crenças e atitudes, pensamentos e aprendizagens que estão presentes em seus comportamentos e nos dos outros.
2. Por comparar diferentes visões sobre o mesmo tema, serve para investigar comportamentos complexos e motivações.
3. A dinâmica estabelecida no grupo pode ser um fator sinérgico no fornecimento de informações.
4. Ajuda o pesquisador a conhecer a linguagem que a população usa para descrever as experiências, os valores, os estilos de pensamento, além do processo de comunicação que estabelecem.
5. É produtivo, pois um número pequeno de grupos pode gerar um grande número de ideias sobre as categorias pretendidas pela pesquisa.

Com o objetivo de compreender o conceito, o uso, os dados produzidos, os benefícios e as limitações de cada um dos dispositivos apresentados, foi possível elaborar um quadro comparativo entre eles e analisar como podem impactar o contexto da pesquisa. As informações detalhadas estão apresentadas na Tabela 4.

Tabela 4. Quadro comparativo entre os dispositivos adotados na pesquisa

DISPOSITIVO	TIPO DE DADO	PONTOS FORTES	PONTOS FRACOS
Diário de pesquisa	Qualitativo	Registro detalhado das experiências, observações e reflexões do pesquisador; fornece dados em tempo real; permite ao pesquisador expressar seus sentimentos, pensamentos e percepções pessoais relacionados ao contexto da pesquisa.	É uma ferramenta subjetiva, os dados coletados podem ser influenciados pelos vieses e interpretações do pesquisado; o pesquisador pode selecionar apenas as informações consideradas relevantes ou interessantes; não segue uma estrutura ou formato padronizado.
Formulário <i>online</i>	Quantitativo	Permitem a coleta rápida e eficiente de dados de um grande número de participantes; são estruturados com perguntas específicas e opções de resposta predefinidas, o que facilita a padronização das respostas dos participantes; garantem o anonimato e a confidencialidade dos participantes, incentivando-os a responder com mais sinceridade.	Os participantes podem fornecer respostas superficiais, apenas para concluir o questionário; os participantes podem enfrentar dificuldades em entender ou interpretar corretamente as perguntas do questionário, levando a respostas imprecisas ou confusas; não permitem uma interação direta entre o pesquisador e o participante, o que limita a capacidade de esclarecer dúvidas.
Diário de classe	Quantitativo	Permite um registro detalhado das atividades, interações e eventos ocorridos em sala de aula; permite o registro imediato das informações logo após as aulas; pode reunir uma variedade de informações, como anotações do professor, registros de presença, evidências de aprendizado dos alunos, anotações de comportamento e até mesmo reflexões pessoais do professor.	Pode ser influenciado pela subjetividade do professor; o professor pode selecionar apenas as informações que considera relevantes ou interessantes para registrar no diário de classe, deixando de fora outros aspectos ou eventos menos significativos; não têm uma estrutura ou formato padronizado.
Grupo focal	Qualitativo	Permite a interação direta entre os participantes, promovendo a troca de ideias, percepções e experiências; oferece a oportunidade de explorar temas complexos de forma mais abrangente e aprofundada; é uma fonte rica de dados qualitativos, permitindo capturar narrativas, percepções, atitudes e crenças dos participantes de maneira mais abrangente.	As dinâmicas de grupo podem influenciar as respostas e opiniões dos participantes; geralmente envolvem um número limitado de participantes; requer tempo e recursos consideráveis.

Fonte: elaborado pelo autor

4.3 Cotidiano pesquisado

Em uma pesquisa de docência, o cotidiano pesquisado refere-se às práticas, experiências, desafios e contextos vivenciados no dia a dia do professor em sua atuação profissional. Isso inclui as atividades de planejamento e preparação de aulas, as interações com os alunos, a aplicação de estratégias de ensino, a avaliação do progresso dos estudantes, o uso de recursos e tecnologias educacionais, entre outros aspectos relacionados à prática docente.

Na pesquisa, as experiências foram vivenciadas em sala de aula na Escola Técnica

Estadual Maria Mercedes Mendes Teixeira durante o primeiro semestre letivo de 2022. Nesse período, atuei como professor responsável pela disciplina de Linguagem de Programação 1, lecionando para a turma do primeiro módulo do curso técnico de informática, composta por vinte e cinco alunos. O semestre teve uma duração total de 144 aulas, realizadas em dois encontros semanais. O primeiro encontro consistia em seis tempos de aula, enquanto o segundo encontro era composto por dois tempos de aula. Durante o semestre, os alunos puderam experimentar uma abordagem didática inovadora e a adoção de novas práticas docentes.

As questões de pesquisa foram abordadas por meio da utilização de uma variedade de dispositivos, tanto em termos de formato quanto de tipo de dados produzidos. Esses dispositivos proporcionaram a obtenção de dados quantitativos e qualitativos, que, juntamente com as observações, percepções e sentimentos do pesquisador, contribuíram para atingir os objetivos da pesquisa. Foram empregados o diário de pesquisa, um questionário *online*, um diário de classe e um grupo focal.

No diário de pesquisa, as informações mais relevantes daquele encontro foram registradas imediatamente após cada aula lecionada. O diário foi mantido no período de fevereiro a julho de 2022 e desempenhou um papel fundamental como instrumento de apoio e fonte de informações ao longo da pesquisa.

O questionário *online* foi desenvolvido utilizando a plataforma LimeSurvey e foi aplicado aos alunos na última aula do semestre, imediatamente após a apresentação do projeto final da disciplina. O objetivo do questionário era produzir dados de identificação do perfil dos participantes e compreensão de como os recursos e atividades propostos durante as aulas contribuíram para a aprendizagem de programação.

O diário de classe foi utilizado para registrar informações sobre a situação escolar de cada aluno ao longo do semestre, incluindo a frequência às aulas. O preenchimento do diário ocorria ao final de cada aula, fornecendo informações em tempo real tanto para o professor quanto para a secretaria escolar. Esses registros permitiram acompanhar de perto o desempenho e a presença dos alunos ao longo do semestre.

Os projetos de avaliação entregues pelos alunos ao longo do semestre desempenharam um papel fundamental no acompanhamento do progresso individual de cada aluno no campo da programação. Além disso, esses projetos também foram utilizados como critério para avaliar o desempenho de cada aluno, contribuindo para a atribuição das respectivas notas.

O histórico de notas desempenhou um papel fundamental na pesquisa, permitindo ao pesquisador investigar o desempenho dos estudantes em semestres anteriores na disciplina LP1. Por meio dessa análise, foi possível compreender e registrar informações como a taxa de evasão e abandono, bem como os percentuais de aprovação e reprovação. Ao considerar

o histórico da disciplina e o cenário proporcionado pela pesquisa, o pesquisador foi capaz de responder às questões identificadas de forma mais embasada.

O grupo focal foi realizado logo após a aplicação do questionário, com o objetivo de aprofundar as informações coletadas e promover o diálogo entre os alunos. A sessão do grupo focal foi registrada em áudio e vídeo pelo pesquisador, permitindo a análise posterior dos dados registrados. Esse método proporcionou uma oportunidade para uma discussão mais aprofundada e uma compreensão mais abrangente das percepções e experiências dos participantes.

Após a conclusão do registro de dados por meio dos diversos dispositivos de pesquisa durante o primeiro semestre de 2022, o pesquisador pôde avançar para a próxima etapa da investigação no segundo semestre do mesmo ano. Nessa fase, o foco foi direcionado para a análise e interpretação dos dados registrados, visando compreender os resultados alcançados. Essa análise permitiu ao pesquisador examinar de forma mais aprofundada os dados e extrair percepções relevantes para o desenvolvimento da pesquisa.

5 RESULTADOS

Nesta seção, apresentaremos os resultados obtidos a partir da utilização dos dispositivos de avaliação empregados. Tais dispositivos incluem: o Diário de Pesquisa, elaborado durante o primeiro semestre de 2022; o questionário *online*, que foi desenvolvido e aplicado aos estudantes para coletar informações relacionadas à avaliação da solução proposta; o Diário de Classe, um documento que registra informações sobre a disciplina LP1; e o Grupo Focal, um instrumento utilizado com os estudantes para compreender, avaliar e aprofundar a abordagem proposta, bem como as práticas docentes adotadas na disciplina.

5.1 Diário de Pesquisa

O semestre letivo teve início em 14/02/2022, com a retomada das aulas presenciais após dois anos de aulas remotas devido à pandemia da covid-19. Na primeira semana, houve uma baixa procura pelos estudantes, resultando em uma turma esvaziada no primeiro módulo da disciplina Linguagem de Programação 1. Inicialmente, apenas três alunos estavam matriculados na turma noturna, comparecendo intercaladamente nas três primeiras aulas. Devido à regra da escola, que não permite que um professor leccione quando apenas um aluno está presente, esses três encontros foram direcionados para uma conversa sobre a disciplina, as expectativas dos alunos e seus conhecimentos prévios de programação, buscando entender o perfil de cada um dos estudantes matriculados.

Durante essas conversas, fiquei preocupado porque os três alunos não pareciam interessados em permanecer na turma e consideravam mudar para outro turno ou até mesmo de curso, já que as turmas da manhã e da tarde estavam mais cheias. Compartilhei minha preocupação com a secretaria da escola e fui informado de que, se os três alunos desistissem, a turma seria cancelada e eu seria alocado para lecionar outra disciplina. Essa possibilidade me deixou ansioso e preocupado, pois o cancelamento da turma tornaria inviável a aplicação da solução proposta na minha pesquisa.

Na mesma conversa com a secretaria da escola, fui informado de que o período que determinaria o cancelamento da turma aconteceria logo após o feriado de Carnaval, no início de março. Durante esse período, havia a expectativa da chegada de novos alunos para compor a turma, já que, pelo calendário da FAETEC, o mês de março estava reservado para a disponibilização das vagas ociosas e remanescentes do concurso, preenchidas através de sorteio para o público em geral. Logo, a permanência dos alunos já matriculados e o ingresso de novos estudantes pelo sorteio manteriam a turma com um maior número de alunos.

Após a retomada das aulas após o período de carnaval, os três alunos anteriores permaneceram matriculados e um novo aluno ingressou no curso, totalizando quatro alunos matriculados. Com a turma confirmada para o restante do semestre e com a presença total dos alunos, as aulas puderam ser iniciadas de acordo com o planejamento elaborado.

O primeiro passo foi realizar um mapeamento junto aos alunos para identificar seu perfil em relação ao conhecimento prévio de programação. Foi constatado que os alunos da turma não tinham nenhuma experiência prévia em programação de computadores. O início do aprendizado de programação ocorreu com a introdução do conceito de algoritmos, utilizando as plataformas Hora do Código e LightBot, que permitiram apresentar aos alunos os conceitos iniciais de programação de uma forma lúdica. Os estudantes se mostraram totalmente interessados e engajados para resolver os desafios propostos por cada nível de dificuldade das plataformas. A estratégia utilizada em sala de aula foi resolver todos os níveis disponíveis em uma das atividades, progredindo nível a nível até o último, e sugerir que eles resolvessem, no mínimo, uma segunda atividade em casa. Depois de superarem os desafios em ambas as plataformas, os estudantes apresentaram um *feedback* positivo sobre a facilidade de uso das plataformas, com um consenso de que o Lightbot promoveu desafios mais estimulantes do que a Hora do Código.

Durante o decorrer da disciplina, diversos recursos tecnológicos foram utilizados para auxiliar na aprendizagem dos alunos. Entre eles, recursos audiovisuais como videoaulas, video tutoriais e *podcasts* foram empregados e conseguiram atrair ainda mais a atenção dos estudantes durante a exibição em sala de aula.

No primeiro módulo da disciplina, chamado “Pensamento Computacional”, o Scratch foi adicionado como ferramenta de apoio. Os alunos avaliaram a ferramenta como de fácil utilização e a possibilidade de desenvolver jogos e animações despertou o interesse de todos. Durante as aulas, foram criados projetos de exemplo e estabelecidos desafios para que os alunos pudessem praticar e se desenvolver. Ao final de cada aula de Scratch, foi sugerido que os alunos trabalhassem em seus projetos de avaliação, melhorando-os com novos recursos e/ou funcionalidades. Os alunos se motivaram e mostraram-se engajados com todas as possibilidades oferecidas pelo Scratch, e isso se refletiu no aumento da troca de mensagens no grupo de WhatsApp da turma.

No contexto da avaliação da disciplina, foi acordado que os alunos deveriam desenvolver um projeto de jogo autoral no Scratch em vez de realizar uma prova tradicional para o primeiro bimestre. Essa escolha aumentou a motivação dos alunos em aprender programação e melhorar seus jogos, o que foi evidente no interesse em trazer suas dúvidas para a aula. Essa abordagem de avaliação foi considerada bem-sucedida para manter o interesse dos alunos na disciplina e aprimorar os conceitos de programação.

Observando a motivação dos alunos, foi realizada a dinâmica denominada Estacionamento Algorítmico. A dinâmica foi realizada em duplas para fortalecer a programação

em pares, um método em que duas pessoas trabalham juntas em uma única solução. Esperava-se que os alunos pudessem trabalhar em equipe para solucionar os problemas propostos na dinâmica. Os desafios e tabuleiros utilizados na dinâmica estão ilustrados no Apêndice C. A Figura 27 ilustra uma dupla durante a dinâmica.

Figura 27. Alunos durante a dinâmica do Estacionamento Algorítmico



Fonte: registrada pelo autor

A dinâmica realizada com os alunos foi um sucesso, mantendo-os engajados e motivados em resolver os desafios propostos nos níveis iniciante e intermediário. Ao final da aula, foi realizada uma conversa para que os alunos pudessem fornecer *feedback* sobre todo o processo da dinâmica. Eles afirmaram que a atividade foi interessante e desafiadora, contribuindo para aprimorar as habilidades de lógica e algoritmo. Um estudante destacou

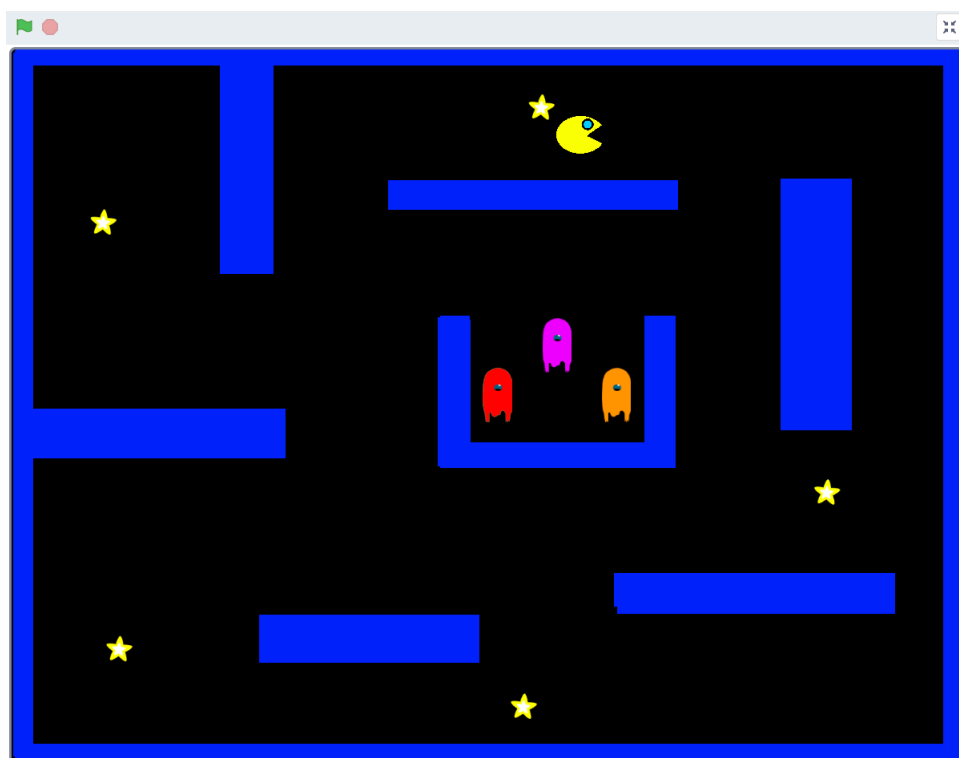
que agora estava começando a entender programação e que a abordagem utilizada com novas práticas poderia contribuir significativamente para o aprendizado de programação.

“Agora sim estou começando a entender programação. Em 2019, eu tranquei porque não entendia nada, a professora chegava na sala e passava uns códigos sem explicar nada e a turma toda ficava sem entender o que era para fazer”

Além disso, durante o *feedback*, os alunos mencionaram a possibilidade de construir um projeto de jogo no Scratch com base na dinâmica do Estacionamento Algorítmico, o que se encaixa perfeitamente com a abordagem de atividades autorais. Embora haja complexidades na construção do projeto, essa possibilidade despertou grande interesse nos alunos.

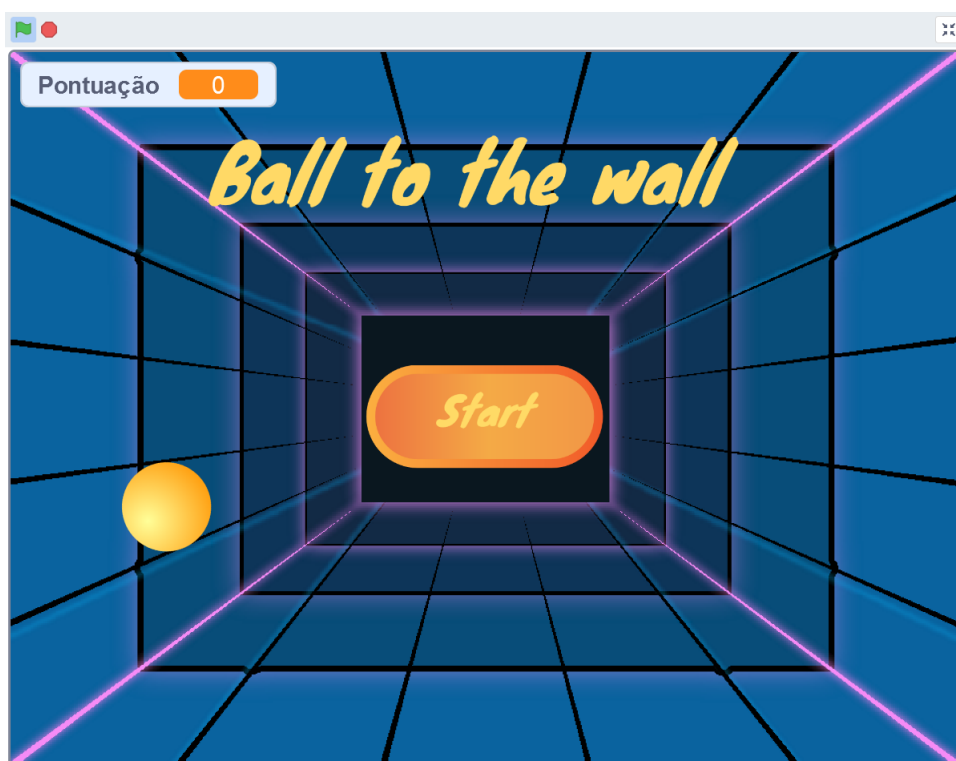
A finalização do módulo Pensamento Computacional foi marcada pela entrega e apresentação dos projetos de jogos autorais dos alunos no Scratch, que correspondeu à avaliação do primeiro bimestre. Dos quatro alunos que iniciaram a disciplina, um comunicou que não permaneceria no curso devido à mudança de horário no emprego, o que o tornou incompatível com o horário da escola. Os outros três estudantes apresentaram seus projetos para a turma. Fiquei surpreso positivamente com os projetos elaborados, dos quais dois eram jogos com menos recursos, porém, funcionais e que podem ser evoluídos. O primeiro jogo simula o tradicional Pacman, conforme ilustrado na Figura 28. Já o segundo se assemelha ao estilo de jogo de pinball, conforme ilustrado na Figura 29. Por fim, o terceiro projeto, conforme ilustrado na Figura 30, superou minhas expectativas, pois o estudante criou um jogo do tipo perseguição em que um avião ataca com tiros os inimigos, registra a pontuação do jogador e, ao atingir um total de pontos, o jogador é direcionado para a fase do "chefe". Além disso, o estudante implementou uma tela de início e de fim de jogo (*game over*). Foi interessante perceber como esses estudantes conseguiram construir seus jogos no Scratch sem nenhum conhecimento prévio de programação, apenas participando das aulas e atividades propostas na disciplina LP1, após 1 mês do início das aulas.

Figura 28. Captura de tela do jogo no formato PACMAN



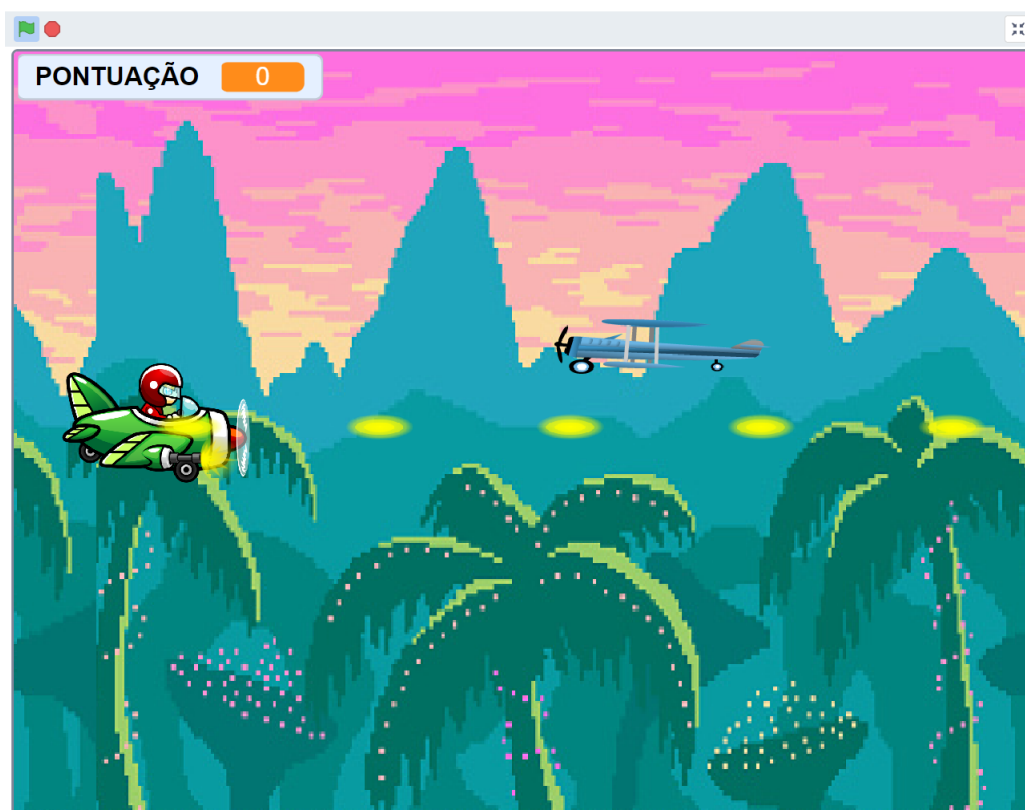
Fonte: registrada pelo autor

Figura 29. Captura de tela do jogo no formato PIMBALL



Fonte: registrada pelo autor

Figura 30. Captura de tela do jogo no formato TIRO



Fonte: registrada pelo autor

Após a conclusão do módulo de Pensamento Computacional, iniciamos o módulo de Linguagem de Programação, com foco nos conceitos básicos de programação e com o apoio do Calango. Optamos pelo Calango em vez do Portugol devido à sua maior simplicidade na codificação, o que poderia simplificar o aprendizado. No início, apresentamos o ambiente de desenvolvimento e a estrutura básica do Calango. No entanto, percebi que os alunos encontraram dificuldades em resolver problemas simples, então incluí a prática de resolver exercícios em conjunto com os alunos, buscando orientá-los e esclarecer as dúvidas apresentadas.

Ao final da elaboração de exercícios que abordavam os conceitos de entrada e saída de dados, operações matemáticas e de comparação, foi possível perceber a evolução dos alunos, que passaram a escrever códigos de forma autônoma. É relevante destacar que, mesmo com as dificuldades iniciais, a turma manteve-se interessada, participativa e engajada em adquirir os conhecimentos necessários que estavam sendo apresentados. É importante ressaltar que parte das dificuldades apresentadas pelos alunos era esperada no momento de transição da programação com blocos para a escrita de códigos, mesmo em uma pseudolíngua como a do Calango.

Deve-se considerar um ponto que pode ter impactado o aprendizado dos alunos durante a pesquisa. A disciplina de Lógica para Informática, lecionada por um novo pro-

fessor, utilizou o Netbeans e o Java como ferramentas de apoio em seu conteúdo. Esse fato precisou ser observado de perto, uma vez que os alunos codificavam os exercícios propostos pelo professor sem ter conhecimento inicial de programação em Java. A aprendizagem simultânea do Calango e do Java pode ter gerado dificuldades de compreensão por parte dos alunos, visto que eles escreviam os comandos em Java sem entender completamente o que estavam fazendo, apenas para que o exercício “funcionasse”.

No início do mês de abril, foi confirmado que novos estudantes se juntariam à turma, que até então era composta por quatro alunos. Além disso, alunos dos turnos da manhã e tarde poderiam migrar para a turma da noite, preenchendo vagas ociosas do concurso realizado. Isso apresentou um novo desafio para a pesquisa, já que grupos distintos de alunos que ingressaram em períodos diferentes na escola foram formados dentro da mesma turma. Adicionalmente, outro desafio que poderia surgir seria lecionar para alunos que migraram do turno da manhã ou tarde, já que os professores estavam ensinando programação com o auxílio da ferramenta Portugol.

Na semana seguinte ao meu retorno à sala de aula, fui surpreendido com a chegada de dez novos alunos que ingressaram no curso. Ao consultar o diário da disciplina, constatei que mais cinco alunos estavam matriculados, mas não estavam presentes naquele momento. Não posso negar que, naquele momento, experimentei uma mistura de sentimentos que foi da frustração à euforia. A frustração veio pelo fato de que todo o semestre estava planejado de acordo com a quantidade de alunos que eu tinha, e quase na metade do semestre a turma quadruplicou de tamanho. Por outro lado, fiquei entusiasmado por ter mais alunos na turma e, por consequência, ter uma amostra mais representativa para a pesquisa.

Contudo, em minha opinião, os alunos foram os mais prejudicados no processo de aprendizagem, já que ingressaram na metade da disciplina e precisaram se adaptar aos conteúdos já lecionados. É importante ressaltar que essa prática é comum na FAETEC, que libera vagas ociosas para o público em geral através de sorteio e permite o ingresso de alunos após o início das aulas. No entanto, no primeiro semestre de 2022, devido ao cenário da pandemia e à retomada das aulas presenciais, a chegada dos alunos foi ainda mais tardia.

Foi necessário (re)planejar minha atuação a partir daquele momento. A solução que encontrei para minimizar o impacto nos novos estudantes foi não realizar qualquer tipo de avaliação no primeiro bimestre e replicar a nota alcançada no segundo bimestre. Com relação ao conteúdo da disciplina, os alunos que estavam cursando desde o início receberam a atribuição de resolver uma lista de exercícios propostos para desenvolver no Calango, enquanto os novos alunos receberam o conteúdo de aproximadamente seis semanas em apenas duas. Essa estratégia buscou nivelar o conhecimento da turma para que o semestre pudesse decorrer sem a necessidade de dividir os alunos em grupos. O módulo de Scratch consistiu apenas em uma demonstração rápida e apresentação dos jogos elaborados pelos

outros alunos, como uma oficina da plataforma, possibilitando uma visão geral da mesma.

É interessante notar que o perfil dos alunos recém-chegados era bastante diverso. O grupo era composto por pessoas entre 18 e 55 anos, alguns deles já graduados em outras áreas e trabalhando em carreiras como Pedagogia e Segurança do Trabalho. Um aluno estava cursando Biomedicina e os demais buscavam oportunidades no mercado de trabalho. Pelo que pude perceber inicialmente, a maioria dos alunos estava interessada e deve concluir o curso, independentemente das dificuldades que surgissem. No entanto, alguns ingressaram no curso com a expectativa de adquirir conhecimentos básicos de informática, como usar um computador, um pacote de escritório e navegar na internet, e poderiam desistir a qualquer momento. Na primeira semana, os alunos ficaram assustados ao terem contato com programação, lógica, modelagem de dados, *web* e bancos de dados. Os professores sempre buscaram manter a motivação dos alunos, já que, além das disciplinas do curso serem difíceis, eles estavam em desvantagem em relação ao tempo de ingresso, participando apenas da metade do período letivo. Esse cenário poderia levar os alunos a desistirem do curso, não apenas na unidade em que leciono, mas em outras unidades da rede.

Como ponto positivo, destaco a motivação evidenciada pelos alunos em adquirir os conhecimentos apresentados, tanto através de sua participação e estudos realizados em sala de aula quanto em atividades extracurriculares. Levando em consideração que cerca de oito semanas de aulas foram realizadas em apenas três, o comprometimento de cada aluno foi fundamental para nivelar a turma no conteúdo da disciplina.

No entanto, a estratégia de apresentar diretamente os conceitos de linguagem de programação e a utilização do Calango para os novos alunos não apresentou os resultados esperados. Minha conclusão é que ocorreu o mesmo que no período pré-pandemia, quando os alunos apresentavam enormes dificuldades no aprendizado. Então, revisei o planejamento e realizei com os novos alunos a atividade Estacionamento Algorítmico, que já havia sido executada anteriormente com outros alunos. Após a conclusão da atividade, solicitei *feedback* aos alunos, e foi unânime que a dinâmica realizada foi interessante, proporcionando ganhos na parte lógica e simplificando o conceito de instruções sequenciais com algoritmos. Um dos alunos me parabenizou pela realização da atividade e disse que “*a didática utilizada na atividade foi perfeita*”.

Lecionar para duas turmas simultaneamente foi um desafio constante que exigiu adaptação do planejamento inicial para buscar o nivelamento do aprendizado, manter a motivação dos alunos e trabalhar com conteúdos distintos para grupos diferentes dentro da mesma sala de aula.

Ao avaliar o cenário de ingresso de alunos em diferentes momentos do semestre, fica evidente a dificuldade do professor em ajustar o planejamento da disciplina para nivelar o conhecimento da turma, considerando que todos os alunos ingressaram sem qualquer conhecimento prévio sobre programação. Entretanto, um ponto positivo deste

cenário desafiador é a oportunidade de avaliar os impactos da abordagem proposta na pesquisa em comparação com aqueles que ingressam diretamente no módulo de linguagem de programação, sem a oportunidade de navegar pelo primeiro módulo proposto. Esses resultados serão mais evidentes no segundo bimestre, pois contarão com a avaliação de toda a turma.

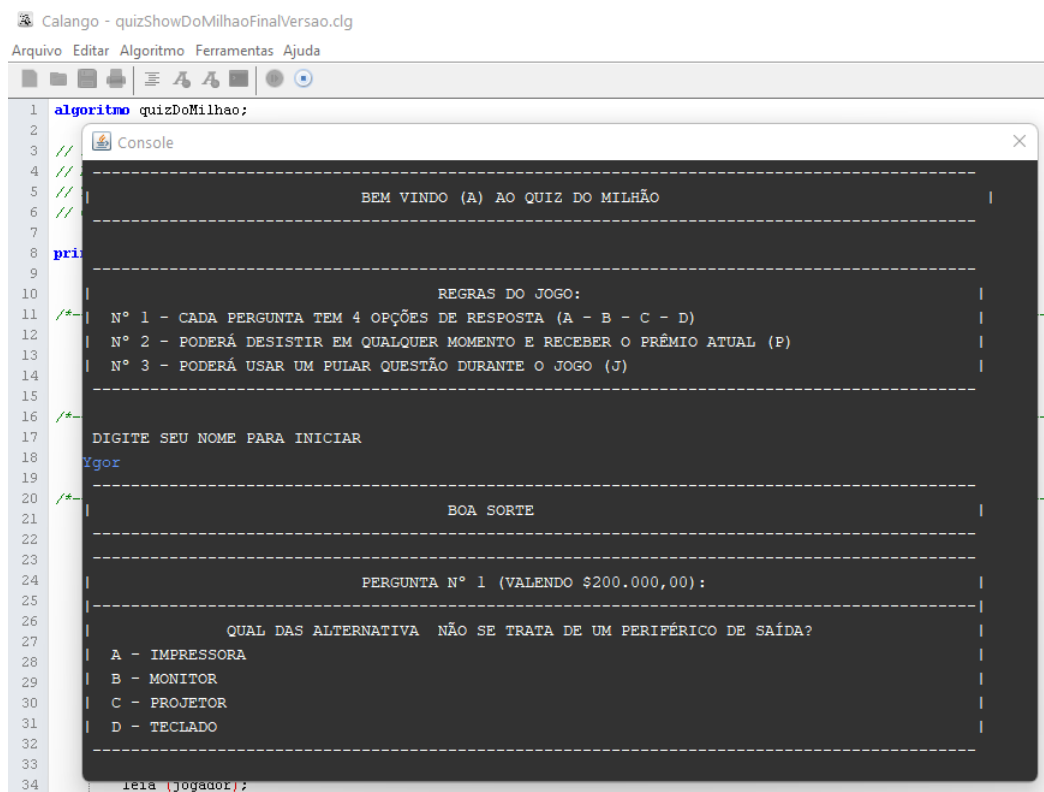
No segundo bimestre, seguindo os princípios da pesquisa, utilizou-se como instrumento de avaliação a elaboração de jogos no Calango e no Java. Os alunos tiveram três opções para escolher, incluindo o jogo “Show do Milhão”, o “Jogo da Forca” e o “Jogo da Velha”. Trabalharam em duplas para criar seus jogos, com cerca de trinta e cinco dias para completar o projeto. Devido ao fato de serem alunos iniciantes em programação, a complexidade dos jogos foi limitada, com o objetivo de garantir que aprendessem conceitos como variáveis, entrada e saída de dados, estruturas de controle e repetição, além de procedimentos e funções. A escolha de jogos populares como tema para o projeto também foi positiva, pois despertou o interesse dos alunos e tornou o processo de criação mais envolvente e divertido. Os jogos “Show do Milhão”, o “Jogo da Forca” e o “Jogo da Velha” são conhecidos e apreciados, o que motivou os estudantes a se esforçarem na construção de suas versões.

O projeto foi dividido em duas etapas, com os alunos desenvolvendo o mesmo jogo no Calango e, em seguida, em Java. A avaliação dos alunos baseou-se na entrega dos códigos do jogo, na apresentação para a turma e no relato da experiência de trabalhar em duplas no desenvolvimento do projeto. O modelo de avaliação, que não previa a aplicação de provas teóricas, deixou os alunos motivados e engajados com a atividade.

Com o encerramento do módulo de linguagem de programação, o terceiro e último módulo da disciplina teve início. Intitulado “Língua Java”, este módulo apresentou aos alunos o conceito da língua, o ambiente de desenvolvimento e a sintaxe do Java. A metodologia adotada para este módulo foi a “tradução” dos códigos elaborados no Scratch e no Calango. Os problemas já resolvidos no Calango foram usados como base para chegar às soluções do mesmo problema no Java, através da tradução do código. Por exemplo, um exercício que propunha o cálculo do fatorial, já desenvolvido no Calango, foi utilizado, e as estruturas de condição e repetição foram adaptadas para a sintaxe do Java.

O terceiro módulo da disciplina foi concluído na data prevista para a apresentação dos jogos desenvolvidos pelos alunos. Além da avaliação dos alunos, foram realizadas a avaliação da solução através da aplicação do questionário de pesquisa e a realização do grupo focal, que serão detalhados nas próximas seções. A Figura 31 ilustra o funcionamento do jogo similar ao “Show do Milhão” desenvolvido por uma das duplas no Calango.

Figura 31. Captura de tela do jogo desenvolvido no Calango



Fonte: registrada pelo autor

Um ponto importante que não pode ser deixado de ser registrado é o desafio encontrado para a reorganização dos conteúdos da disciplina seguindo a nova abordagem proposta. Essa etapa gerou muitos desafios e exigiu uma quantidade significativa de trabalho, que precisa ser considerada como parte da pesquisa. Preparar materiais para aulas, exercícios, atividades e dinâmicas durante o semestre foi altamente desafiador e me exigiu dedicação e planejamento para alcançar tudo o que foi planejado. O preparo de todo o material precisou estar alinhado com minhas atividades profissionais, além de minhas demais atividades acadêmicas e docentes.

5.2 Questionário *online*

A coleta de dados ocorreu por meio de um formulário *online* de pesquisa, disponível no Apêndice E, que foi disponibilizado para os alunos no dia 27 de junho de 2022, logo após a apresentação dos projetos de jogos. Para motivar a participação do maior número de estudantes, expliquei a importância da pesquisa e como ela pode contribuir para a comunidade científica. Aqueles que aceitaram participar da pesquisa foram convidados a assinar o Termo de Consentimento Livre e Esclarecido, disponível no Apêndice D.

O dispositivo de coleta de dados foi elaborado por meio da plataforma LimeSurvey, escolhida pelo pesquisador devido à sua experiência prévia no uso da ferramenta e aos

recursos que a plataforma oferece. O questionário foi composto por duas seções: a primeira apresentou uma breve introdução à pesquisa, indicando seus objetivos, e o termo de consentimento para participação. A segunda seção foi dividida em questões abertas e questões fechadas. A primeira parte das perguntas teve como objetivo coletar informações que permitissem a construção do perfil dos participantes, incluindo informações como gênero, idade, equipamentos utilizados durante a disciplina, locais disponíveis para estudo e conhecimento prévio de programação antes do início do curso. Já a segunda parte do questionário buscou entender como os recursos e atividades propostas oportunizaram a aprendizagem de programação. Os participantes puderam avaliar, em uma escala de cinco pontos, a importância de cada item listado e fornecer *feedback* por meio de questões abertas.

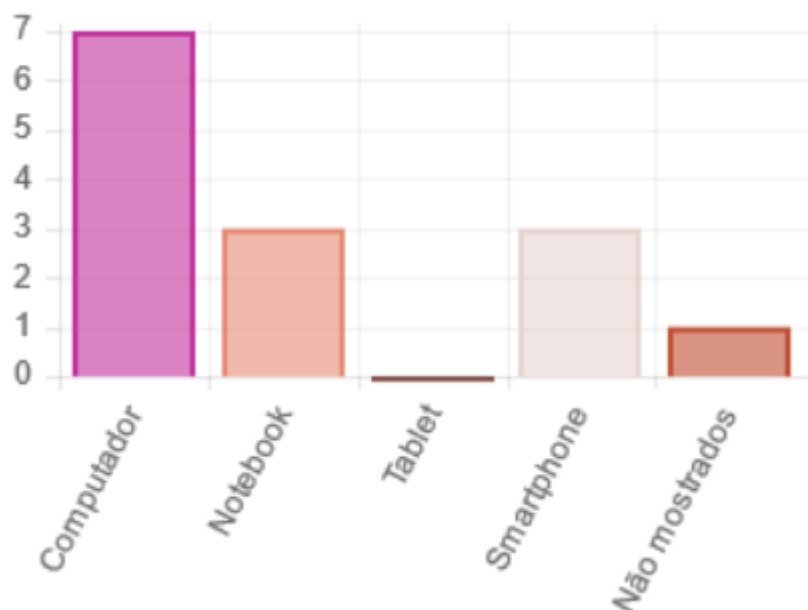
O questionário foi aplicado na sala de aula logo após a apresentação dos projetos de jogos em Calango e Java, com o objetivo de garantir a participação do maior número possível de respondentes. Foram obtidas vinte e uma respostas dentro do prazo estabelecido para a coleta de dados. No entanto, foram desconsiderados quatorze registros incompletos, restando apenas sete registros preenchidos de forma completa.

Os dados coletados foram analisados quantitativa e qualitativamente. Os dados quantitativos foram extraídos das questões fechadas e de múltipla escolha e foram analisados utilizando a técnica de estatística descritiva (SANTOS, 2007). A análise qualitativa foi realizada a partir das respostas às questões abertas do questionário, seguindo os passos recomendados por Creswell (2014), incluindo a preparação e organização dos dados e a identificação dos principais achados.

Os dados coletados na primeira metade do questionário permitiram a caracterização dos estudantes respondentes. Dos sete participantes, seis eram do gênero masculino (85,71%) e um do gênero feminino (14,29%). A idade dos participantes foi distribuída da seguinte forma: um participante com 18 anos, um participante com 19 anos, dois participantes com 20 anos, um participante com 30 anos, um participante com 31 anos e um participante com 37 anos.

As questões que visavam entender quais dispositivos e locais foram utilizados para estudar a disciplina permitiam aos respondentes selecionar mais de uma opção de resposta. Com relação aos dispositivos utilizados, todos os estudantes afirmaram utilizar o computador, enquanto três selecionaram a opção *Notebook*, três a opção *Smartphones* e um a opção Não Mostrado, como ilustrado na Figura 32. Em relação ao local de estudo para a disciplina, cinco estudantes selecionaram a opção Casa, um selecionou a opção Trabalho, três selecionaram a opção Laboratório da Escola e um selecionou a opção Não Mostrado.

Figura 32. Dispositivos utilizados pelos estudantes durante a disciplina



Fonte: registrada pelo autor

A questão sobre o conhecimento prévio dos estudantes antes de iniciar o curso técnico mostrou que dois estudantes nunca tinham ouvido falar de programação, enquanto três estudantes já haviam escutado sobre programação e outros dois estudantes já conheciam programação através de vídeos no YouTube.

A Tabela 5 apresenta um resumo dos dados coletados sobre a importância de cada atividade ou recurso para os estudantes. Os estudantes só poderiam selecionar uma opção, sendo 0 (zero) se a atividade/recurso foi irrelevante para o processo de aprendizagem, 4 (quatro) se a atividade/recurso foi muito importante, ou um valor intermediário para expressar sua avaliação. A opção X (Não se aplica) estava disponível para os estudantes que não participaram da atividade ou não utilizaram o recurso em questão. Analisando a tabela, é possível afirmar que a maioria das respostas se concentraram nas opções quatro e três, que possuem um alto grau de importância. Conclui-se que as estratégias adotadas na abordagem proposta foram eficazes para a aprendizagem em programação pelos alunos.

É relevante enfatizar que ao longo do semestre, houve a disponibilização de um curso gratuito de Java, cortesia da Oracle para os alunos. Isso foi possível graças a uma parceria estabelecida entre a Oracle e a FAETEC, que concede tanto a professores quanto a alunos a oportunidade de participar de programas de treinamento em Java, abrangendo desde o nível introdutório até o avançado, através de uma plataforma de ensino a distância (EAD), sem nenhum custo. Esse curso, que inicialmente não estava planejado, se mostrou mais um recurso adotado para enriquecer o processo de ensino e aprendizagem dos estudantes.

Tabela 5. Sumário das respostas coletadas na importância de cada atividade/recurso

	X	0	1	2	3	4
Hora do Código	1	0	1	0	2	3
Lightbot	2	0	1	0	1	3
Scratch	1	0	0	0	2	4
Dinâmica "Estacionamento algorítmico"	2	0	0	0	1	4
Recursos audio visuais	0	0	0	1	1	5
Pseudo código (Calango)	0	0	0	0	2	5
Projeto de jogo em Scratch	1	0	0	0	1	5
Projeto de jogo em Calango	1	0	0	0	1	5
Projeto de jogo em Java	0	0	0	0	2	5
Listas de Exercícios	1	0	0	1	1	4
Tradução de pseudo código para Java	0	0	0	0	1	6
Atividades em pares	0	0	0	1	0	6
Material disponibilizado (Apresentações em PDF)	0	0	0	0	0	7
Curso de Java na plataforma Oracle Academy	1	0	0	1	0	5

Fonte: elaborado pelo autor

Avaliando as atividades de apoio à programação com blocos, a **Hora do Código** recebeu um total de cinco avaliações com maior grau de importância, representando 71,4% das respostas. Apenas um estudante considerou a atividade pouco importante e outro estudante não participou da atividade.

Já o **Lightbot** recebeu quatro avaliações indicando que a atividade foi importante para o processo de aprendizagem (57,1%), uma resposta indicando baixa importância, enquanto dois estudantes não participaram da atividade.

O Scratch e a dinâmica do **Estacionamento Algorítmico** receberam um total de 100% de respostas indicando muita importância para o processo de aprendizagem na disciplina, quando desconsiderados os estudantes que não participaram da atividade. O **Scratch** recebeu seis respostas com maior importância (85,7% do total) e apenas um aluno não participou da atividade. Já o **Estacionamento Algorítmico** recebeu cinco respostas com maior grau de importância (71,4% do total), enquanto dois estudantes não participaram da atividade.

Considerando a avaliação dos projetos de jogos digitais realizados na turma, o desenvolvimento de um **Projeto de Jogo em Scratch** registrou seis respostas, indicando que a atividade foi muito importante para a aprendizagem (85,7%). Apenas um dos estudantes não participou da atividade. O mesmo ocorreu no **Projeto de jogo em Calango**, onde os registros de resposta foram idênticos. Já no **Projeto de jogo em Java**, 100% das respostas indicaram que a elaboração do projeto como ferramenta de avaliação foi importante para a aprendizagem.

Considerando as **Atividades em pares** utilizadas para a construção dos projetos de jogos digitais, seis estudantes indicaram que foram muito importantes para a aprendizagem (85,7%), enquanto um aluno indicou que não foi tão importante.

É possível afirmar que a substituição das tradicionais provas por projetos de jogos digitais como ferramenta de avaliação foi importante e contribuiu significativamente para o processo de aprendizagem dos estudantes.

O ensino de linguagem e língua de programação utilizado foi considerado muito importante pelos estudantes. A utilização do **pseudocódigo Calango** foi avaliada como importante por 100% dos alunos. Já a estratégia de lecionar a língua Java por meio da **tradução do pseudocódigo para Java** também registrou um grau de importância de 100% para os alunos. É possível afirmar que a estratégia de utilizar inicialmente a programação com blocos, em seguida o pseudocódigo na língua nativa dos estudantes e, por último, iniciar uma língua de programação, com a tradução dos códigos escritos em Calango para Java, contribuiu significativamente e ajudou os alunos a compreenderem programação de computadores.

No contexto dos recursos oferecidos aos alunos durante a disciplina, a utilização de **recursos audiovisuais** durante as aulas, como vídeos disponíveis na plataforma Youtube e projeções em datashow, foi considerada importante por 85,7% dos estudantes (seis), enquanto um aluno considerou não ter muita importância. As **listas de exercícios** utilizadas durante as aulas foram consideradas importantes por 71,4% dos estudantes, embora um estudante tenha considerado não ser tão importante e outro estudante não tenha realizado as listas. O **material em PDF** disponibilizado registrou um total de 100% dos estudantes indicando que era muito importante para o processo de aprendizagem. O **curso de Java na plataforma Oracle Academy** registrou um total de cinco estudantes identificando-o como um recurso muito importante (71,4%), enquanto um estudante o identificou como um recurso sem importância e outro estudante não participou do curso.

É possível afirmar que os recursos disponibilizados para os alunos durante a disciplina foram importantes como artefatos de apoio para o processo de aprendizagem dos estudantes.

Após o preenchimento da questão que avaliou a importância dos recursos utilizados na disciplina, os alunos foram convidados a responder três questões abertas. Nessas questões, os alunos poderiam explicar por que algumas atividades/recursos foram mais ou menos importantes para seu aprendizado e sugerir alterações para o próximo semestre, incluindo ou excluindo atividades ou recursos. O preenchimento dessas questões não era obrigatório, assim como as demais questões. A primeira questão solicitava que os alunos explicassem por que algumas atividades foram mais importantes no aprendizado de programação. Abaixo estão as respostas coletadas:

1. “Eu só tinha visto programação em vídeo no YouTube e nem era tutorial era pessoas programando e eu fascinado com aquilo. Linguagem de programação me mostrou que não é tão difícil quanto aparenta O calango facilitou muito na utilização de Java

Toda a estrutura de ensino uma coisa ligando a outra ajudou bastante.”

2. “Através dos recursos áudio e visual pode assimilar melhor o conteúdo e com os PDFs tive a oportunidade de organizar e rever os conceitos.”
3. “Pois possibilitou uma apresentação mais amigável ao processo de desenvolvimento.”
4. “Pela facilidade no entendimento.”
5. “Porque através dessas atividades consegui compreender de forma fácil a matéria de programação.”
6. “As atividades com maior importância foram as que mais influenciaram, de maneira prática, o aprendizado da programação. Sendo essenciais para o entendimento da disciplina, de forma que, sem elas, a compreensão do conteúdo seria quase inalcançável, sem um entendimento prévio de programação.”

Ao avaliarmos as respostas dos alunos, podemos observar que, de forma geral, todos mencionaram que as atividades/recursos proporcionaram maior facilidade no aprendizado de programação. Essa estratégia de encadeamento do aprendizado de programação em conjunto com os recursos de apoio disponíveis na disciplina possibilitou um melhor aprendizado por parte dos alunos.

A próxima questão solicitava que os alunos explicassem por que algumas atividades/recursos foram menos importantes para o aprendizado de programação. Abaixo estão as respostas registradas:

1. “Pelo fato de eu não poder aplicar.”
2. “Todas as atividades em que participei tiveram um mesmo grau perceptível de importância.”
3. “Sem essas ferramentas a didática seria a mesma.”

Ao analisarmos as respostas sobre as atividades/recursos que tiveram menos importância no aprendizado, notamos que o número menor de respostas pode estar relacionado à falta de clareza por parte dos alunos sobre a qual atividade/recurso estavam se referindo. É possível perceber que alguns alunos não participaram ou não realizaram determinadas atividades, conforme pode ser observado na Tabela 5, o que pode ter influenciado na sua percepção sobre a importância das mesmas no processo de aprendizagem.

A última pergunta do questionário pedia que os alunos indicassem possíveis alterações na disciplina para o próximo semestre, incluindo ou excluindo temas, atividades ou recursos. As seguintes respostas foram identificadas:

1. “Que as pessoas possam ter um laboratório para estudos pois nem todo mundo tem computador, notebook etc. para realizar as atividades que são postas nas aulas. Mais trabalho em sala sendo corrigido no mesmo dia para tirar dúvida dos alunos seria muito bom.”
2. “Nada, seu jeito de ensinar é muito bom como está.”
3. “Acredito que todos os recursos foram bem elaborados e atenderam a maioria da turma. Porém gostaria de ressaltar a importância de viabilizar condições para que alunos tenham acesso ao laboratório da unidade para que assim possam praticar e desenvolver seus conhecimentos Visto que nem todos dispõem de acesso a dispositivos adequados.”
4. “Aumentar o número de projetos nas plataformas de desenvolvimento, facilitar o uso dos laboratórios da Unidade Escolar.”
5. “Nenhuma, pois o modo que foi efetuado na disciplina foi excelente.”
6. “Aulas mais práticas, de programação, de forma livre, em dias fora do horário de aula.”

Ao analisarmos as respostas dos alunos a essa questão, é possível observar que a principal demanda dos alunos está relacionada à infraestrutura e recursos disponíveis na unidade escolar como um todo, e não especificamente à disciplina em questão. A principal sugestão ou demanda dos alunos é a disponibilização de um laboratório de informática aberto, onde eles possam estudar e praticar programação em horários alternativos aos de aula. Entendemos que essa demanda possa ser intensificada pelo fato de alguns alunos não terem acesso a um espaço com computadores para estudo, o que pode limitar seu aprendizado, já que nem todos têm computadores disponíveis em casa.

No que diz respeito à disciplina em si, é relevante destacar que não foram apresentadas sugestões de alteração na ementa da disciplina pelos alunos. Esse fato nos permite afirmar que o planejamento e a organização da disciplina foram bem-sucedidos em alcançar os objetivos esperados no processo de aprendizagem de programação de computadores pelos estudantes.

5.3 Diário de Classe

Na escola em que o pesquisador atua, o diário de classe é disponibilizado pela secretaria em uma planilha eletrônica compartilhada, permitindo que todas as partes interessadas tenham acesso em tempo real. A planilha é organizada em guias mensais, nas quais os professores registram a frequência dos alunos e o conteúdo ministrado em cada

dia de aula. Um exemplo é ilustrado na Figura 33, que mostra a guia referente ao mês de maio de 2022. Além disso, há uma guia final denominada “consolidado”, que contém todas as informações relevantes do semestre letivo, como a frequência dos alunos, as notas atribuídas e a situação final de cada aluno na disciplina. A Figura 34 exibe o consolidado da disciplina LP1 ao final do primeiro semestre de 2022.

Figura 33. Captura da tela do diário no mês de maio de 2022

CURSO: TÉCNICO EM INFORMÁTICA										TURMA: 311.122										DISCIPLINA: LINGUAGEM DE PROGRAMAÇÃO I										Lançamento da Matéria Lecionada									
DISCIPLINA: LINGUAGEM DE PROGRAMAÇÃO I										TURMA: 311122																													
NUM.	Anotação de Frequência às Aulas Dadas																														Total Faltas	Total Presença	Nº	Dias	Resumo de Maio				
	Maio																																						
01	2	2	2	2	2	5	5	12	16	16	16	16	16	16	19	19	23	23	23	23	30	30	30	30	30	30	28	0	01	2	ESTRUTURA DE REPETIÇÃO: ESTUTURA PARA, ENQUANTO E FAÇA.. ENQUANTO; EXERCICIOS								
02	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	0	28	02	5	PROCEDIMENTOS; FUNÇÕES; EXERCICIOS							
03																											28	0	03	9	CONSELHO DE CLASSE (COC)								
04	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	0	28	04	12	FUNÇÕES CALANGO							
05	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	0	28	05	15	LISTA DE EXERCICIOS 2 E 3							
06						F	F																			26	2	06	19	WORKSHOP REALIZADO EM 14/05: DESENVOLVIMENTO DE SITCS COM CMS									
07								F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	22	6	07	23	VETOR; MATRIZ; INTRODUÇÃO AO JAVA; LISTA DE EXERCICIOS 2 E 3								
08	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	0	28	08	26	PROJETO DA DISCIPLINA; TESTE DE MESA NO CALANGO							
09	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	0	28	09	30	ESTRUTURA CONDICIONAL IF E SWITCH; EXERCICIOS							
10															F	F						F	F			24	4	10											
11	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	0	28	11									
12	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	0	28	12									
13							F	F									F	F	F	F	F	F	F	F	F	F	16	12	13										
14	F	F	F	F	F	F	F	F									F	F	F	F	F	F	F	F	F	F	18	10	14										
15	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	0	28	15										
16																										28	0	16											
17	F	F	F	F	F	F	F	F													F	F	F	F	F	F	16	12	17										
18						F	F	F	F							F	F	F	F	F	F	F	F	F	F	16	12	18											
19	F	F	F	F	F	F															F	F				20	8	19											
20	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	0	28	20										
21	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	0	28	21										
22	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	0	28	22										
23																										10	18	23											
24	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	0	28	24										
25																										28	0	25											
26	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	0	28	26										
27																										9	1	27											
28																										0	0	28											

Fonte: registrada pelo autor

Figura 34. Captura da tela do diário na planilha de consolidado

io de Cla	CURSO: TÉCNICO EM INFORMÁTICA												
	TURMA: 311.122												
	DISCIPLINA: LINGUAGEM DE PROGRAMAÇÃO I												
	TOTAL DE AULAS DADAS: 144			CARGA HORÁRIA:									
AVALIAÇÕES													
NUM.	Total Presença s	Total Faltas	% Pres.	TESTE 1º BIM	PROVA 1º BIM	MÉDIA 1º BIM	TESTE 2º BIM	PROVA 2º BIM	MÉDIA 2º BIM	MÉDIA FINAL	Prova Final	Nota Final	Resultado
01	116	28	81%	10,0	10,0	10,0	9,0	9,0	9,0	9,5		9,5	APROVADO
02	50	94	35%	NA	NA	#DIV/0!	NA	NA	#DIV/0!	#DIV/0!		#DIV/0!	REP. FALTAS
03	130	14	90%	6,5	6,5	6,5	9,0	9,0	9,0	8,0		8,0	APROVADO
04	0	144	0%	NA	NA	#DIV/0!	NA	NA	#DIV/0!	#DIV/0!		#DIV/0!	TRANCADO
05	0	144	0%	NA	NA	#DIV/0!	NA	NA	#DIV/0!	#DIV/0!		#DIV/0!	TRANCADO
06	130	14	90%	6,0	6,0	6,0	8,0	8,0	8,0	7,0		7,0	APROVADO
07	72	16	50%	9,0	9,0	9,0	9,0	9,0	9,0	9,0		9,0	APROVADO
08	14	74	10%	NA	NA	#DIV/0!	NA	NA	#DIV/0!	#DIV/0!		#DIV/0!	DESSISTENTE
09	0	88	0%	NA	NA	#DIV/0!	NA	NA	#DIV/0!	#DIV/0!		#DIV/0!	DESSISTENTE
10	76	12	53%	NA	NA	#DIV/0!	NA	NA	#DIV/0!	#DIV/0!		#DIV/0!	REPROVADO
11	0	88	0%	NA	NA	#DIV/0!	NA	NA	#DIV/0!	#DIV/0!		#DIV/0!	TRANCADO
12	6	82	4%	NA	NA	#DIV/0!	NA	NA	#DIV/0!	#DIV/0!		#DIV/0!	DESSISTENTE
13	30	58	21%	NA	NA	#DIV/0!	NA	NA	#DIV/0!	#DIV/0!		#DIV/0!	DESSISTENTE
14	64	24	44%	9,0	9,0	9,0	9,0	9,0	9,0	9,0		9,0	APROVADO
15	6	82	4%	NA	NA	#DIV/0!	NA	NA	#DIV/0!	#DIV/0!		#DIV/0!	DESSISTENTE
16	88	0	61%	8,5	8,5	8,5	9,0	9,0	9,0	9,0		9,0	APROVADO
17	30	58	21%	NA	NA	#DIV/0!	NA	NA	#DIV/0!	#DIV/0!		#DIV/0!	REP. FALTAS
18	58	30	40%	9,5	9,5	9,5	9,5	9,5	9,5	9,5		9,5	APROVADO
19	68	20	47%	9,0	9,0	9,0	9,0	9,0	9,0	9,0		9,0	APROVADO
20	6	82	4%	NA	NA	#DIV/0!	NA	NA	#DIV/0!	#DIV/0!		#DIV/0!	DESSISTENTE
21	0	82	0%	NA	NA	#DIV/0!	NA	NA	#DIV/0!	#DIV/0!		#DIV/0!	DESSISTENTE
22	8	74	6%	NA	NA	#DIV/0!	NA	NA	#DIV/0!	#DIV/0!		#DIV/0!	DESSISTENTE
23	34	48	24%	9,0	9,0	9,0	9,0	9,0	9,0	9,0		9,0	APROVADO
24	0	82	0%	NA	NA	#DIV/0!	NA	NA	#DIV/0!	#DIV/0!		#DIV/0!	DESSISTENTE
25	68	0	47%	9,5	9,5	9,5	9,5	9,5	9,5	9,5		9,5	APROVADO
26	0	68	0%	NA	NA	#DIV/0!	NA	NA	#DIV/0!	#DIV/0!		#DIV/0!	TRANCADO
27	13	37	9%	NA	NA	#DIV/0!	NA	NA	#DIV/0!	#DIV/0!		#DIV/0!	TRANCADO

LEVANTAMENTO ESTATÍSTICO			
CURSO:		TÉCNICO EM INFORMÁTICA	
TURMA:		LINGUAGEM DE PROGRAMAÇÃO I	
TRANCADOS:	5	APROVADOS:	10
DESISTENTES:	7	REPROVADOS:	1
TRANSFERIDOS:	0	REP. FALTAS:	2
EVADIDOS:	0		

Fonte: registrada pelo autor

A turma de LP1 contou com um total de vinte e cinco alunos matriculados ao final do semestre. Ao analisar a coluna de frequência na planilha consolidada, é notável que a maioria dos alunos não atingiu o mínimo de 75% de presença exigido pela escola. Essa situação pode ser justificada pelo fato de que muitos alunos se juntaram à turma em momentos diferentes durante o semestre, e a planilha de frequência considera o semestre

como um todo. Portanto, para esses alunos, a frequência foi avaliada manualmente, levando em conta o período efetivamente cursado e o total de faltas.

Ao final do semestre, foi atribuída uma situação para cada aluno, de acordo com seu desempenho na disciplina: trancado, desistente, transferido, evadido, aprovado, reprovado ou reprovado por falta.

A análise estatística da situação da turma pode ser observada na Tabela 6. A distribuição por situação se apresentou da seguinte forma: cinco alunos como trancados; sete alunos como desistentes; dez alunos como aprovados; um aluno como reprovado; e dois alunos reprovados por falta.

A situação “Trancado” é atribuída ao aluno que, até a data limite estabelecida pela escola, solicita o trancamento de sua matrícula no semestre corrente apresentando uma justificativa. Já a situação “Desistente” é atribuída ao aluno que deixa de frequentar a escola e a equipe de orientação pedagógica não consegue nenhum contato com o estudante. Esses alunos têm sua matrícula cancelada automaticamente sem qualquer possibilidade de renovação.

A situação “Aprovado” é atribuída ao aluno que alcança o mínimo de 75% de frequência e média final mínima de seis. A situação de “Reprovado” é atribuída ao aluno que não alcança a nota mínima exigida pela escola, seja na média dos bimestres ou na prova final. Por fim, a situação “Reprovado por falta” é atribuída ao aluno que não alcança o mínimo de frequência estabelecida pela escola.

Essas informações são importantes para a compreensão do desempenho da turma e para o planejamento de ações pedagógicas que possam contribuir com o sucesso dos estudantes.

Tabela 6. Análise estatística da situação da turma

SITUAÇÃO	QUANTIDADE	PERCENTUAL
Trancado	5	20%
Desistentes	7	28%
Aprovados	10	40%
Reprovados	1	4%
Reprovados por falta	2	8%

Fonte: elaborado pelo autor

Ao avaliar o desempenho da turma, é possível perceber que 48% dos estudantes interromperam ou abandonaram a disciplina. Esse número está próximo das taxas de evasão mundial e nacional. Analisando individualmente, observamos que 20% dos alunos optaram por trancar a matrícula e 28% desistiram do curso.

Para entender as razões por trás da saída desses alunos, entrei em contato com eles para ouvir suas justificativas. Alguns optaram por não continuar no curso por receio de não

conseguir acompanhar o conteúdo de todas as disciplinas, uma vez que ingressaram no curso na metade do semestre. Outros precisaram interromper o curso devido à incompatibilidade de horários entre trabalho e escola e, devido à sua situação econômica, a prioridade no momento era manter o emprego.

Dos 52% de alunos que finalizaram a disciplina, 40% foram aprovados e 12% reprovados. Considerando-se somente o número de alunos que concluíram a disciplina, o percentual de aprovados foi de 76,9%, enquanto o percentual de reprovados foi de 23,1%.

No entanto, a análise superficial dos números e percentuais da turma não permite concluir se a abordagem proposta pela pesquisa foi eficaz. Para avaliar adequadamente os resultados obtidos, é necessário considerar os demais dados coletados e apresentados nesta seção. Apesar de ter havido um número expressivo de aprovações, é preciso verificar se houve melhora significativa no desempenho da turma.

5.4 Grupo Focal

Neste estudo, foi realizado um Grupo Focal logo após a aplicação do questionário *online*, com o objetivo de aprofundar as informações coletadas e promover o diálogo entre os participantes. A participação dos estudantes no Grupo Focal foi voluntária, e o pesquisador esclareceu como a atividade seria conduzida, assumindo o papel de mediador. Como não havia observadores ou relatores presentes, a sessão foi gravada em vídeo para posterior análise e transcrição. Os estudantes foram informados da gravação e convidados a assinar um Termo de Autorização de Imagem e Voz, disponível no Apêndice F.

Todos os alunos aceitaram participar do Grupo Focal, que foi realizado com eles sentados em uma roda, conforme ilustrado na Figura 35. O pesquisador mediador permaneceu de pé para conduzir a sessão e operar os equipamentos de apoio, que consistiram em dois aparelhos celulares: um para captar o áudio dos estudantes e outro para gravar o vídeo da dinâmica. Os arquivos digitais gerados foram posteriormente transcritos. Durante a sessão, o pesquisador solicitou que os estudantes falassem um de cada vez, a fim de evitar a sobreposição de vozes e facilitar a transcrição, mas esclareceu que cada estudante poderia responder à colocação de outro participante, pois o objetivo era debater o assunto proposto de forma colaborativa.

Figura 35. Grupo Focal em ação



Fonte: registrada pelo autor

Durante a dinâmica do Grupo Focal, os estudantes tiveram a oportunidade de discutir os assuntos propostos pelo mediador. Cada tópico foi apresentado individualmente aos participantes por meio de uma apresentação desenvolvida especificamente para esse fim, conforme ilustrado no Apêndice G. O objetivo do pesquisador ao conduzir o Grupo Focal foi obter informações mais detalhadas sobre os assuntos considerados mais relevantes para a pesquisa. Esses temas já haviam sido abordados anteriormente por meio de um formulário de pesquisa, no qual os participantes atribuíram valores em uma escala, gerando dados quantitativos para a análise. Durante a sessão, os temas foram explorados em profundidade por meio das seguintes questões:

1. Quais são os pontos fortes e fracos da disciplina Linguagem de Programação 1?
2. O que vocês acharam da Programação com blocos (Hora do Código, Lightbot e Scratch)? Ela deve ser mantida ou retirada na próxima turma da disciplina de programação do nosso curso?
3. O que vocês acharam da dinâmica “Estacionamento Algorítmico”? Ela deve ser mantida ou retirada na próxima turma da disciplina de programação do nosso curso?
4. O que vocês acharam da utilização de pseudo código (Calango)? Ela deve ser mantida ou retirada na próxima turma da disciplina de programação do nosso curso?
5. O que vocês acharam da utilização de Java? Ela deve ser mantida ou retirada na próxima turma da disciplina de programação do nosso curso?

6. O que vocês acharam na aprendizagem baseada em projetos (jogos digitais)? Ela deve ser mantida ou retirada na próxima turma da disciplina de programação do nosso curso?
7. O que vocês acharam da avaliação da disciplina? O que vocês sugerem de modificação para a próxima turma da disciplina de programação do nosso curso?

Para transcrever os registros, o pesquisador utilizou um processo semelhante ao realizado por Azevedo (2022) em seu projeto de graduação na UNIRIO. Inicialmente, o *software* Wondershare Filmora foi utilizado para melhorar a qualidade do áudio e eliminar ruídos nas gravações. Em seguida, os áudios foram separados em novos arquivos, um para cada participante, para facilitar a transcrição.

A próxima etapa consistiu na instalação do *software* Voicemeeter¹, que permite ao usuário gerenciar qualquer fonte de áudio de ou para qualquer dispositivo ou aplicativo de áudio. Isso possibilitou ao microfone captar o conteúdo da gravação reproduzido pelos alto-falantes do computador. Com o *software* configurado conforme as orientações do fabricante e o computador capaz de captar o próprio áudio, foi utilizada uma ferramenta *online* gratuita, o Dictation.io², para transcrever a fala em texto.

É importante ressaltar que o processo automatizado não garante uma transcrição perfeita das falas. A ferramenta tem dificuldades em transcrever gírias, termos em outros idiomas ou mesmo certas palavras em português. Por isso, uma revisão manual da transcrição foi realizada para corrigir erros, ajustar o texto e garantir que o resultado se aproxime ao máximo do que foi dito pelos participantes. A transcrição completa pode ser encontrada no Apêndice H.

Com a transcrição finalizada, foi possível aprofundar cada uma das questões apresentadas pelo mediador no Grupo Focal. A primeira pergunta, que tratava dos **pontos fortes e fracos da disciplina LP1**, revelou como pontos positivos a utilização de projetos como instrumento de avaliação, a disponibilização de materiais de apoio elaborados pelo professor e a utilização de pseudocódigo como parte do processo de aprendizagem. Esses pontos são evidenciados nas falas dos estudantes, como pode ser destacado a seguir:

“Outro ponto que eu achei bom foi o fato de ser projeto e não ser prova porque prova a gente vai tipo só tentar gravar ali ou memorizar, esse projeto do show do milhão é claro que a gente aprendeu bem mais do que ficar só estudando ali, eu aprendi bem mais tudo, o if, o else, tudo e a gente praticou mais também.”

“...pela metodologia que o senhor aplica é fácil entender a matéria já que é dividida tudo certinho em slide ou se a gente tem uma dúvida em um ponto específico, a gente vai naquele ponto e já está lá tudo bonitinho para a gente, tem exemplos, tem tudo...”

¹ <https://vb-audio.com/Voicemeeter/>

² <https://dictation.io/speech>

“... o Calango nos trouxe a lógica e claro que a didática do Calango nos preparou para poder enfrentar o Java...”

É possível perceber que o uso de material didático próprio, preparado especificamente para a disciplina e organizado de acordo com o cronograma das aulas, pode ser considerado uma prática positiva. Além disso, ficou evidente entre os alunos que a utilização de projetos como instrumento de avaliação, em substituição às tradicionais provas, é uma abordagem pedagógica eficaz e bem aceita pelos estudantes. Esses aspectos foram apontados como pontos fortes da disciplina LP1, contribuindo significativamente para o aprendizado dos alunos.

Observou-se que a pedagogia por projetos estimulou uma maior dedicação por parte dos estudantes, promovendo o estudo ao longo do tempo, em vez de limitar-se apenas aos dias que antecedem as provas, como geralmente ocorre. A adoção do pseudocódigo Calango também foi apontada como um ponto positivo, já que proporcionou uma aprendizagem mais simples e permitiu que os alunos compreendessem os conceitos de programação de forma mais intuitiva, facilitando o aprendizado da língua Java, na qual foi necessário apenas entender a sintaxe. Outro aspecto favorável identificado foi o fato de o pseudocódigo ser totalmente compreensível no idioma nativo dos estudantes, o que auxiliou no entendimento e na assimilação dos conceitos de programação.

Os pontos negativos destacados pelos alunos apontam para a necessidade de ampliar o tempo destinado à prática em sala de aula, possibilitando um maior comprometimento com a aprendizagem. Além disso, os estudantes sugeriram a elaboração de um projeto inicial que seja aprimorado ao longo da disciplina, o que lhes permitiria trabalhar em um projeto contínuo e progredir gradualmente. Também foi mencionada a dificuldade enfrentada por alguns alunos na utilização do Calango em seus próprios computadores, o que impediu que eles pudessem praticar adequadamente, como pode ser observado nas seguintes falas:

“Se tiver tempo para fazer mais exercício em sala você tem um comprometimento maior”.

“Então um projeto inicial e a gente ir aprimorando ele até chegar ao final com ele todo otimizado também seria bom.”

“Essa questão também professor, eu tive muita dificuldade de baixar o Calango e executar, pedi ajuda e muitos me ajudaram mas mesmo assim eu não consegui, então eu fiquei assim sem poder praticar por causa dessa dificuldade que eu tive enquanto o equipamento né...”.

Esses pontos levantados pelos alunos são importantes para aprimorar as práticas docentes. Disponibilizar mais tempo em sala de aula para a prática pelos alunos pode ser fundamental para o processo de aprendizagem e também pode beneficiar aqueles que não

possuem computador ou tiveram dificuldades para instalar o ambiente Java e o Calango em seus próprios dispositivos. Iniciar um projeto e aprimorá-lo conforme novos conceitos de programação são apresentados pode ser uma alternativa para substituir a aplicação de listas de exercícios, já que cada exercício é isolado dos demais e das outras listas.

A segunda pergunta da dinâmica abordou a Programação com Blocos e sua importância na disciplina. Durante a discussão, foi consenso entre os participantes que a utilização dessa tecnologia contribuiu para o aprendizado e aumentou o interesse e a dedicação dos alunos à disciplina. Os desafios e a interface gráfica proporcionaram um aprendizado mais prazeroso e, ao mesmo tempo, mais desafiador. Esse consenso pode ser verificado nas seguintes falas dos alunos:

“mesmo que eu tenha chegado depois, o fato deles terem apresentado o trabalho me interessou mais a querer aprender, a querer ver, pensar e tentar.”

“Ele que me deu interesse, porque assim que eu cheguei no curso já tinha o projeto dos caras, de montar um jogo e eu já estava caraca o bagulho *inaudível, seguido por muitas gargalhadas* os moleques estão montando um jogo me ajudou a querer e aprender a fazer.”

Essa reflexão aponta para a importância de utilizar instrumentos pedagógicos que estimulem a criatividade e o interesse dos alunos pelo conteúdo. A Programação com Blocos pode ser uma excelente alternativa para alcançar esse objetivo, já que possibilita a criação de projetos mais divertidos e interessantes, além de ser uma forma mais acessível e fácil de aprendizado, especialmente para os alunos iniciantes.

A pergunta seguinte abordou a dinâmica do Estacionamento Algorítmico e sua continuidade nas próximas turmas. Embora os alunos tenham concordado com a importância da dinâmica, não houve muita discussão durante a sessão, pois eles entenderam que deveria ser realizada no início do semestre. Foi destacado que a dinâmica ajudou no entendimento da lógica e na compreensão de que um programa de computador é formado por uma sequência de comandos que são executados.

Durante a sessão, o mediador questionou sobre a possibilidade de levar a dinâmica para o meio digital, utilizando a plataforma Scratch. Os alunos acharam essa ideia interessante e desafiadora, pois permitiria aprimorar os conhecimentos no Scratch com um jogo já conhecido pelos alunos.

A discussão sobre **a utilização do Calango e sua manutenção nas próximas turmas** gerou um diálogo interessante, com momentos de divergência no grupo. O Calango, de acordo com os estudantes, facilitou a aprendizagem e os preparou adequadamente para a transição para a língua Java, permitindo a tradução da sintaxe do pseudocódigo para o código em Java. A interface amigável do Calango também foi apontada como um ponto positivo, proporcionando uma experiência mais acessível e menos intimidadora

em comparação com as IDEs mais complexas geralmente usadas no desenvolvimento de códigos em Java.

Um aspecto crucial destacado pelos alunos foi o uso do pseudocódigo como etapa anterior ao início da língua Java. Eles mencionaram que essa abordagem foi fundamental para evitar desistências na disciplina e no curso, pois o módulo de Calango forneceu uma base sólida para compreender os conceitos de programação. Além disso, os estudantes que ingressaram no curso no meio do semestre também puderam acompanhar o conteúdo e adquirir conhecimentos em programação, mesmo sem ter participado do módulo de programação com blocos. As falas dos alunos destacadas reforçam essas afirmações:

“Eu tinha medo de código e o Calango diminuiu isso.”

“...se tivesse começado com o Java eu não estaria aqui não...”

“É uma estrutura de base para você aprender o Java.”

A pergunta sobre a **utilização do Java e sua continuidade nas próximas turmas** revelou um consenso entre os estudantes, indicando que a abordagem de ensino de Java logo após o Calango deve ser mantida. Eles destacaram que o aprendizado de Java se tornou mais simples e compreensível, uma vez que os conceitos de linguagem de programação foram introduzidos no Calango e, posteriormente, foi necessário apenas conhecer a sintaxe específica do Java e realizar a tradução das estruturas aprendidas no Calango para o Java. As falas dos alunos ilustram bem essa perspectiva:

“Tem que manter no primeiro para a pessoa chegar no segundo já sabendo o que ela vai encontrar porque se a pessoa se acostumar com o Calango e depois der de cara com o Java, vai ver aquela outra realidade... então não, tem que ter aquilo em mente, o Calango é inicial para eu aprender os comandos, o que é função, vetores, variáveis... então, pegou isso, leva para o Java...”

“Mas o que realmente ajudou mesmo foi quando o senhor chegou na introdução do Java que o senhor não mostrou só o começo do Java, comparou no PDF o Calango com o Java, mostrando a diferença.”

“... o senhor começou do zero, variável e variável aqui Calango e no Java, escreva e System.out.println, foi exatamente assim...”

Essas declarações enfatizam a importância de dar continuidade à metodologia de introduzir o Java logo após o Calango, pois os estudantes demonstraram que essa transição gradual foi significativa para o entendimento e assimilação do conteúdo. Ao apresentar os conceitos de programação em uma pseudolíngua como o Calango, os alunos se sentiram mais preparados e confiantes para enfrentar a língua Java, que é considerada mais complexa. A comparação entre o Calango e o Java, mostrando a relação entre as estruturas de ambas, também se mostrou útil para a compreensão dos estudantes.

O diálogo sobre a **aprendizagem baseada em projetos (jogos digitais) e sua continuidade nas próximas turmas** gerou um consenso positivo entre os estudantes. Eles expressaram de forma unânime a opinião favorável ao uso de projetos como método de avaliação e destacaram as vantagens em relação às provas tradicionais. Os alunos acreditam que a avaliação por prova é um método ultrapassado que não reflete o real conhecimento do aluno, enquanto a elaboração de projetos promoveu uma abordagem mais prática da programação, estimulando o aprendizado de forma mais efetiva.

Eles perceberam que os projetos proporcionaram uma prática contínua da programação, possibilitando o enfrentamento de desafios e a realização de pesquisas. Ao substituir as provas escritas por projetos, o aprendizado tornou-se mais significativo e prazeroso, uma vez que não era mais necessário apenas decorar informações para serem reproduzidas em uma prova, mas sim aplicar os conhecimentos na construção de um projeto real.

Além disso, a abordagem de projetos permitiu aos alunos cometerem erros e aprenderem com eles, sem o estresse e a pressão de terem que terminar uma prova em pouco tempo. Essas ideias podem ser percebidas nas seguintes falas:

“Eu acho que só deve ser feita dessa maneira por todos os professores...vamos abolir a prova no papel.”

“É porque você... é a prática, tudo é a prática e ali você está praticando, você está fazendo. Não é no papel onde eu vou ter que decorar não, ali você está praticando.”

“E o melhor disso tudo que eu acho é que não avalia a gente pelo conhecimento que a gente tem naquele momento, avalia a gente pelo processo de ter chegado lá...”

“Outro ponto disso professor, é que quando a gente vai fazer o projeto e a gente tem prazo e a gente pode falar com as outras pessoas, você se sente confortável em errar. Numa prova a gente fica muito nervoso para acertar o máximo de coisas possíveis e a gente não se dá o trabalho de ver o que errou e tentar consertar, a gente quer acabar com aquele momento de sofrimento o mais rápido possível. O projeto a gente queria ter até mais prazo para implementar mais coisas.”

Para finalizar, o grupo discutiu sobre **a avaliação da disciplina e possíveis melhorias para as próximas turmas**. Embora seja uma novidade para todos, o grupo concordou que a avaliação colaborativa, que tem papel fundamental em um ambiente de aprendizagem colaborativa (UGULINO et al., 2009), deve ser mantida como parte do processo de avaliação. Os estudantes se mostraram engajados durante as apresentações, avaliando os demais com base em critérios pré-definidos e absorvendo as críticas construtivas dos colegas. Essa abordagem proporcionou uma troca de experiências significativa entre os alunos, além de incentivar a reflexão sobre seus próprios projetos e aprendizados.

As falas dos estudantes evidenciam a aprovação da avaliação colaborativa, como pode ser visto em:

“Cara sinceramente, eu acho que não precisa mudar nada. Foi uma apresentação descontraída, ninguém ficou nervoso, todo mundo brincou com o outro, teve acidente no meio do percurso.”

“Para mim foi algo novo, eu nunca tinha visto esse negócio.”

“A gente viu uma forma de, ao mesmo tempo de estar avaliando, se auto avaliar né... Vejo uma coisa interessante vou usar na minha avaliação também, vou usar na minha apresentação então assim é uma forma que foi feita para avaliar e se auto avaliar... tipo eu não fiz tão bem quanto ele mas não é uma crítica, é construtivo né, então foi bem interessante.”

“Eu acho que também funciona muito bem para a gente meio que ficar com o pé no chão porque quando a gente faz e funciona, está tudo bem, isso que importa. Mas quando a gente compara com o das outras pessoas a gente fala "pô essa pessoa fez uma coisa que eu nem sequer tentei fazer". Talvez o meu esforço tenha sido menor, de certa forma, que o daquela pessoa. Então se tem que avaliar por nota, e tem um topo que é 10, então na minha visão, eu não poderia ter uma nota igual a daquela pessoa que eu sei que se esforçou mais do que eu. Então acho que funciona muito bem para entender onde a gente está faltando e saber que, mesmo que funcionou, não é porque funcionou que está 100% bom.”

Com base nos resultados apresentados, provenientes da coleta de dados coletados por meio de diferentes dispositivos de pesquisa, encerra-se esta seção, pavimentando o caminho para a realização das discussões na próxima parte.

6 DISCUSSÕES

O objetivo desta seção é apresentar as reflexões sobre os resultados alcançados na pesquisa por meio dos dados quantitativos e qualitativos coletados, sejam eles positivos ou negativos, e como estes podem responder às questões propostas. Além disso, é fundamental incluir a discussão de como o método de Pesquisa-Formação impactou o processo de ensino e aprendizagem, provocando alterações nas práticas docentes.

Antes de responder às questões de pesquisa que nortearam o trabalho, é fundamental avaliar os resultados alcançados a partir dos princípios identificados para a elaboração da proposta de solução. Ao avaliarmos o tema do trabalho, “Ensino de Programação no Ensino Médio por meio de uma abordagem lúdica, autoral e colaborativa”, percebemos que alguns dos princípios são apresentados ao leitor. No entanto, não devemos entender que esses princípios possuem maior relevância ou importância se comparados aos demais. É importante enfatizar que todos os princípios que compõem a solução proposta possuem seu grau de relevância para o trabalho.

Durante todas as seções, é frequentemente mencionada uma “nova abordagem didática” para se referir à solução proposta para a pesquisa. Conforme mencionado, é comum que as disciplinas de programação sejam planejadas para ensinar ao aluno os conceitos de linguagem de programação em uma língua já definida, como C ou Java, por exemplo, com o objetivo de solucionar problemas. Essa forma de organização deriva dos matemáticos, que acreditam que o pensamento abstrato é melhor e que, se pensarmos abstratamente, conseguiremos aplicar e resolver qualquer problema proposto. No entanto, a educação afirma o contrário, orientando a iniciar pelo mais concreto para alcançar o abstrato.

Essa afirmação baseia-se nos estudos do biólogo Jean Piaget, que afirma que a “inteligência de uma criança se modifica junto ao crescimento do indivíduo, partindo de uma inteligência prática, que ajuda o ser humano a se adaptar ao meio, para chegar à inteligência propriamente dita”. É nesse último estágio que a criança já consegue elaborar hipóteses, solucionar situações-problema e praticar um raciocínio lógico.

O desenvolvimento cognitivo inicia-se desde o nascimento e se divide em quatro estágios principais: sensório-motor, simbólico, concreto e formal. O estágio sensório-motor (0 a 2 anos) é caracterizado pelo aprendizado através da experimentação e interação com o mundo ao redor, desenvolvendo a percepção do próprio corpo e dos objetos. Durante essa fase, a criança concentra-se apenas no que pode perceber imediatamente pelos sentidos, e se um objeto não está à vista, para a criança, ele não existe. Outra característica desse estágio do desenvolvimento cognitivo é a transição de uma percepção egocêntrica do mundo para

um maior interesse nas outras pessoas. O bebê adquire progressivamente uma consciência sobre o mundo externo e sobre como os outros percebem esse mundo.

Estágio pré-operacional ou simbólico (2 a 7 anos): Nesse estágio, as crianças começam a dominar a linguagem e os símbolos de comunicação, permitindo que elas imitem, representem, imaginem e classifiquem. Por exemplo, a palavra “carro” já pode gerar uma imagem mental de um carro, mesmo que não haja nenhum à frente delas. Esse estágio é marcado pelo desenvolvimento da comunicação verbal, e as crianças frequentemente falam tudo o que lhes passa pela mente, sem considerar o que as outras pessoas dizem. Conforme a criança envelhece, ela passa a prestar mais atenção no que é dito pelos outros, tornando a conversa mais coerente.

Estágio operatório-concreto (7 a 11/12 anos): Nesse estágio, o egocentrismo diminui e a criança passa a ter maior capacidade de se colocar no lugar do outro e entender conceitos morais de certo e errado, geralmente por volta dos sete anos. As crianças apresentam uma maior capacidade de pensar em soluções mentais para problemas reais, e as informações que receberam até agora começam a se conectar num raciocínio mais lógico, considerando vários aspectos simultaneamente. Conceitos como números e relações são mais facilmente compreendidos, e a linguagem se torna mais socializada. A criança também adquire maior autonomia em relação aos adultos, estabelecendo seus próprios valores morais.

Estágio operatório-formal (a partir de 12 anos): No último estágio do desenvolvimento cognitivo, as crianças são capazes de realizar operações mentais que envolvem abstrações e símbolos que não necessariamente têm formas concretas, ou seja, têm a capacidade do raciocínio abstrato. Elas também são capazes de se colocar no lugar dos outros e imaginar a perspectiva de outras pessoas sobre determinadas situações. A característica mais importante dessa fase é o desenvolvimento do pensamento hipotético-dedutivo. As crianças aprimoram suas habilidades de formular hipóteses para explicar e resolver problemas e começam a entender teorias, doutrinas e conceitos, sendo capazes de fazer leituras críticas do mundo ao redor. Esse processo reforça a vontade de independência e autonomia, permitindo que elas assumam suas opiniões, personalidade e posição no mundo!

Retomando o contexto do ensino de programação, defendemos que o aprendizado deve seguir a linha de pensamento de Piaget, iniciando pelo mais concreto até chegar ao abstrato. Propomos que o aluno aprenda primeiro a linguagem de programação com seus comandos, dados e operações, por meio de atividades como a computação desplugada ou jogos digitais, para assim conseguir desenvolver os elementos da linguagem de programação, como repetição, funções, entre outros. Com esses conceitos bem estabelecidos, os estudantes estarão preparados para resolver problemas complexos.

Com base na ilustração da Figura 36, podemos entender que tradicionalmente as abordagens de ensino de programação utilizam a abordagem *top-down*, ou seja, partem

do mais abstrato para o concreto. Em contraste, nossa pesquisa propõe exatamente o oposto, o que estamos denominando de **Abordagem Construtivista de Programação**, iniciando pelo mais concreto para o abstrato.

Figura 36. Categorias do Domínio Cognitivo



Fonte: Taxonomy of educational objectives: The classification of educational goals: Cognitive Domain, por Bloom et al., 1956, Michigan: Longmans. Adaptado pelo autor.

A Abordagem Construtivista de Programação não é uma cartilha rígida a ser seguida pelo docente em seu planejamento e execução da disciplina, mas sim um formato baseado em princípios que visam alcançar resultados mais satisfatórios junto aos estudantes. Os princípios que orientamos os docentes a seguir são ludicidade, autoria e colaboração. Juntos, esses princípios proporcionam um aprendizado de programação mais simples e prazeroso para o aluno, tornando-o mais engajado e comprometido com a disciplina.

Ao invés de impor uma sequência de passos fixos, a Abordagem Construtivista de Programação oferece flexibilidade ao professor para adaptar suas estratégias de ensino de acordo com as características e necessidades de sua turma. Dessa forma, o docente pode explorar diferentes atividades lúdicas, que despertem o interesse e a criatividade dos alunos, incentivando-os a se tornarem autores de suas próprias soluções.

A colaboração é outro pilar importante dessa abordagem, pois promove a interação entre os estudantes, possibilitando a troca de ideias, o trabalho em equipe e a construção coletiva do conhecimento. A colaboração também estimula a resolução conjunta de desafios, proporcionando um ambiente de aprendizagem cooperativo e enriquecedor.

Para fundamentar a colaboração, recorreremos às teorias de Vigotsky (2007), que defendia que o aprendizado de um indivíduo não pode ser separado do contexto histórico, social e cultural no qual ele está imerso. Para adquirir conhecimento, desenvolver sua compreensão e construir sua própria identidade, o ser humano necessita interagir com

outros indivíduos de sua espécie, com o ambiente e também com a cultura circundante.

As ideias de Vygotsky englobam quatro conceitos fundamentais: interação, mediação, internalização e Zona de Desenvolvimento Proximal (ZDP). O teórico argumentava que, para aprimorar o processo de aprendizagem, não bastava que o indivíduo apenas agisse sobre o ambiente; ele necessitava, acima de tudo, interagir com o ambiente. Vygotsky sustentava que todo indivíduo adquire conhecimento por meio de relações interpessoais, de interações com o ambiente, o que justifica o termo “interativo” nesse contexto. Ele argumenta que, mesmo que um indivíduo possua o potencial biológico para se desenvolver, esse desenvolvimento não ocorrerá plenamente se ele não interagir com os outros.

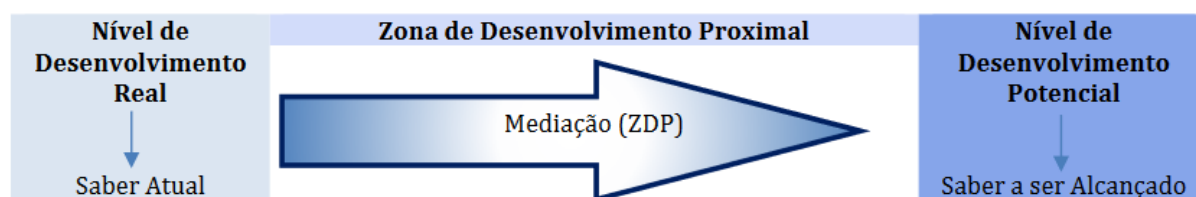
Outro conceito defendido por Vygotsky é a internalização, que ocorre quando o processo de aprendizado se completa. Nesse momento, ao refletir sobre o nome e o significado de um objeto e internalizá-los, o indivíduo consegue abstrair o conceito e torná-lo universal, por meio da mediação da linguagem e da interação com os outros. Dessa forma, ele adquire conhecimentos, papéis sociais e valores.

Segundo Vygotsky, as zonas de desenvolvimento podem ser divididas em três categorias: o nível de desenvolvimento real, que se refere às etapas já alcançadas pelo indivíduo e que lhe permitem solucionar problemas de forma independente; o nível de desenvolvimento potencial, ou seja, a capacidade do indivíduo de desempenhar tarefas com mediação; e a zona de desenvolvimento proximal, que representa a distância entre os níveis de desenvolvimento real e potencial, ou seja, o percurso necessário até atingir o amadurecimento e a consolidação das funções superiores.

A Zona de Desenvolvimento Proximal (ZDP) é o espaço que se situa entre aquilo que o indivíduo já conhece e o potencial de conhecimento que pode adquirir, desde que seja orientado por um mediador. O termo “proximal” implica em algo próximo ou íntimo, sendo o educador ou outro indivíduo mais experiente aquele que identifica o potencial a ser alcançado e estimula o indivíduo a superar-se e a apropriar-se dos conhecimentos que pode adquirir. Nesse contexto, conforme Vygotsky, o educador desempenha o papel de mediador entre o indivíduo e o mundo, atuando como um guia na descoberta da ZDP do aluno, auxiliando-o na interação com os outros e consigo mesmo para alcançar seu verdadeiro potencial.

A Figura 37, elaborada por Santos et al. (2021), ilustra as zonas de desenvolvimento, proporcionando uma melhor compreensão e localização da mediação na ZDP. A mediação que ocorre na ZDP é a ferramenta que impulsiona o indivíduo a maximizar sua aprendizagem e expandir suas percepções. Essa etapa é fundamental para seu progresso e a ampliação de seus conhecimentos.

Figura 37. Zona de Desenvolvimento Proximal



Fonte: Santos et al. (2021, p. 9)

A ludicidade, por sua vez, é um elemento-chave para tornar o ensino de programação mais atrativo e prazeroso. Ao utilizar jogos digitais, atividades interativas e desafios envolventes, os estudantes se sentem mais motivados a aprender e a desenvolver suas habilidades em programação.

Dessa forma, ao adotar a Abordagem Construtivista de Programação, os docentes têm a oportunidade de criar um ambiente de aprendizagem estimulante, onde os alunos se tornam protagonistas de sua própria jornada de aprendizado. Essa abordagem também contribui para que os estudantes desenvolvam habilidades essenciais, como pensamento crítico, resolução de problemas e trabalho em equipe, que são fundamentais para sua formação acadêmica e profissional.

Para seguir a Abordagem Construtivista de Programação em nossa organização da disciplina LP1, iniciamos o aprendizado com a utilização da programação com blocos, apoiados pelas tecnologias Scratch, Hora do Código e Lightbot. Em seguida, avançamos para os conceitos de linguagem de programação com o uso da pseudo-língua Calango. Por fim, introduzimos a língua Java para o estudante.

O princípio da ludicidade pode ser observado com a aplicação de dinâmicas de jogos por meio da computação desplugada e projetos de animações, jogos e histórias durante as aulas. Os estudantes são incentivados a aprender de forma lúdica e divertida.

Já a autoria é percebida no momento em que os estudantes são orientados a estabelecer temas dos projetos de acordo com suas preferências e vivências, sem a imposição da realidade do docente. Eles têm a oportunidade de se tornarem autores de suas próprias criações, desenvolvendo soluções personalizadas e explorando sua criatividade.

A colaboração é uma parte fundamental do ambiente de aprendizado. Os alunos trabalham em atividades de sala e avaliações em duplas e/ou grupos, onde todos conseguem participar e apoiar os colegas, resultando em uma aula interativa e promovendo o trabalho em equipe. A colaboração também estimula o compartilhamento de ideias, a resolução conjunta de desafios e o aprendizado mútuo entre os estudantes.

Após compreendermos a abordagem e os princípios norteadores, nosso objetivo

é responder às questões de pesquisa por meio das informações coletadas nas diferentes fontes apresentadas na seção anterior. As discussões e reflexões serão apresentadas em relação a cada uma das questões.

1. Uma nova abordagem didática no ensino de programação pode proporcionar aos estudantes um processo de aprendizagem mais simples, interessante e motivador?

É possível afirmar que a adoção da abordagem Abordagem Construtivista de Programação na disciplina de Linguagem de Programação 1 conseguiu promover aos estudantes um aprendizado mais simples, interessante e motivador. Essa afirmação é respaldada por observações do pesquisador em sala de aula, bem como pelas respostas fornecidas pelos estudantes em questionários e durante as sessões do grupo focal.

Durante a sessão de grupo focal, os estudantes afirmaram que a abordagem utilizada na disciplina promoveu uma aprendizagem mais simples, especialmente quando o conteúdo foi organizado e segmentado, e artefatos de apoio foram utilizadas durante as aulas. Em uma das aulas, um aluno afirmou “Agora sim estou começando a entender programação. Em 2019 eu tranquei porque não entendia nada, a professora chegava na sala e passava uns códigos sem explicar nada e a turma toda ficava sem entender o que era para fazer”. Segundo o aluno, a turma inteira ficava sem entender o que fazer. Essa observação evidencia que aprender programação a partir de uma língua abstrata pode gerar enormes dificuldades de aprendizagem para os estudantes, e que o ensino do mais concreto para o abstrato simplifica o aprendizado, especialmente para aqueles que não possuem nenhum contato anterior com programação.

Outras falas coletadas durante o grupo focal permitem afirmar que a aprendizagem foi interessante e motivadora para os estudantes. Por exemplo, um dos estudantes disse: “...ele que me deu interesse, porque assim que eu cheguei no curso já tinha o projeto dos caras, de montar um jogo e eu já estava caraca o bagulho inaudível, seguido por muitas gargalhadas os moleques estão montando um jogo me ajudou a querer e aprender a fazer...” e outro afirmou: “...mesmo que eu tenha chegado depois, o fato deles terem apresentado o trabalho me interessou mais a querer aprender, a querer ver, pensar e tentar...”. Essas observações demonstram que as estratégias utilizadas na disciplina foram bem-sucedidas em motivar e engajar os estudantes, proporcionando um ambiente mais dinâmico e estimulante para a aprendizagem.

2. A adoção da abordagem didática proposta pode tornar a aprendizagem prazerosa e engajante?

O aprendizado de programação pode ser entediante e monótono, especialmente quando são utilizados exemplos simples, como calcular a média de um aluno ou um fatorial. Essa abordagem pode desmotivar o aluno, pois ele não consegue perceber como

a programação pode ser interessante e relevante na prática. No entanto, ao incluir jogos digitais, animações, histórias e problemas que fazem parte do cotidiano do aluno, é possível despertar seu interesse e torná-lo mais engajado no processo de aprendizagem. Essa abordagem lúdica e contextualizada permite que o estudante veja a programação como uma ferramenta poderosa para criar coisas divertidas e úteis, tornando o aprendizado mais significativo e prazeroso.

A substituição da prova como instrumento de avaliação pela elaboração de projetos lúdicos e autorais também contribuiu para engajar os estudantes, pois foram desafiados a construir projetos com temas que consideravam interessantes dentro de suas experiências e vivências. O conjunto dessas estratégias permitiu afirmar que a Abordagem Construtivista de Programação promoveu uma aprendizagem prazerosa e engajante aos estudantes, como pode ser evidenciado em algumas falas durante o grupo focal:

“...outro ponto que eu achei bom foi o fato de ser projeto e não ser prova porque prova a gente vai tipo só tentar gravar ali ou memorizar, esse projeto do show do milhão é claro que a gente aprendeu bem mais do que ficar só estudando ali, eu aprendi bem mais tudo, o *if*, o *else*, tudo e a gente praticou mais também...”

“... o próprio site do Scratch já deixa ali delimitado, esse aqui você fala, esse aqui você movimento, assim, assim sucessivamente... só vai arrastando e é bem didático...”

“...outro ponto disso professor, é que quando a gente vai fazer o projeto e a gente tem prazo e a gente pode falar com as outras pessoas, você se sente confortável em errar. Numa prova a gente fica muito nervoso para acertar o máximo de coisas possíveis e a gente não se dá o trabalho de ver o que errou e tentar consertar, a gente quer acabar com aquele momento de sofrimento o mais rápido possível. O projeto a gente queria ter até mais prazo para implementar mais coisas...”

Essas declarações reforçam como a Abordagem Construtivista de Programação proporcionou aos alunos uma aprendizagem mais significativa, interativa e com maior possibilidade de exploração e aprimoramento, em comparação com as abordagens tradicionais baseadas em provas e exemplos abstratos. O engajamento e a motivação demonstrados pelos estudantes ao participarem das atividades de projeto são indícios importantes de que a metodologia aplicada trouxe benefícios reais ao processo de ensino e aprendizagem.

3. A adoção da abordagem didática proposta pode reduzir o percentual de evasão e reprovação?

Responder a esta questão não é uma tarefa simples. Afinal, não basta observar apenas os números sem avaliar o contexto. Ao olhar apenas para os dados apresentados na Tabela 6, notamos que o total de aprovação é de 40%, o que é um percentual baixo e que historicamente se repete. No entanto, se considerarmos as demais situações, chegamos a um total de 60% de alunos que foram aprovados ou estão em processo de recuperação. Avaliando

simplesmente esses números, seria possível afirmar que a Abordagem Construtivista de Programação não reduziu efetivamente a evasão e reprovação.

No entanto, se avaliarmos cada uma das situações que compõem o percentual de 60%, podemos entender melhor a situação. Ao adotarmos este caminho, percebemos que o percentual de reprovados é de 4%, os reprovados por falta são 8%, os desistentes são 28% e os trancados são 20%.

Para entender o contexto, é preciso avaliar os números de cada uma das situações. Ao investigarmos o percentual de 20% dos alunos que trancaram, representando cinco alunos da turma, é importante ressaltar o cenário escolar da unidade. Parte desses alunos ingressou próximo da avaliação do primeiro bimestre e deixou de participar de 50% das aulas. Esses alunos entenderam que seria interessante trancar sua matrícula para reativá-la no semestre seguinte, sem prejuízo de reprovação.

Ao avaliar o percentual de alunos desistentes, correspondente a 28% e um total de sete alunos, notamos que é o maior índice dentre todos os alunos não aprovados. Esses alunos iniciaram os estudos e, por motivos diversos, abandonaram o curso e a escola sem qualquer justificativa. Recebi informações sobre eles de forma oficiosa, por meio de outros alunos da turma, e é possível afirmar que fatores sociais contribuíram significativamente para o percentual de evasão. A FAETEC não oferece alimentação e transporte aos seus alunos, sendo responsabilidade do aluno custear seu deslocamento até a escola e sua alimentação. Além disso, por ser um curso noturno, a violência na cidade também contribuiu para esses números. A escola está localizada no bairro de Ricardo de Albuquerque, zona norte da cidade do Rio de Janeiro, em uma área que oferece riscos aos alunos e professores no momento do deslocamento até suas residências. Por fim, o aspecto econômico também contribuiu para a desistência, uma vez que parte dos alunos são adultos e responsáveis pelo sustento de suas famílias, o que pode dificultar conciliar o emprego com o estudo.

Quanto aos alunos reprovados por falta, um total de dois estudantes, correspondendo a um percentual de 8%, cursaram o semestre parcialmente ou em sua totalidade, mas não alcançaram o índice de presença em todas as disciplinas necessário para aprovação. Os alunos reprovados por falta têm a opção de retomar o curso no semestre seguinte, refazendo todas as disciplinas do primeiro módulo.

Para concluir, é importante destacar que um aluno da turma não obteve aprovação na disciplina, representando 4% do total. Embora tenha frequentado todas as aulas, não participou das atividades e projetos de avaliação, resultando em uma nota abaixo da média mínima para aprovação.

Nesse sentido, não é possível afirmar de forma simples se a Abordagem Construtivista de Programação foi efetiva na redução da evasão e reprovação. É necessário considerar

o contexto e possíveis variáveis externas que possam ter influenciado os resultados obtidos. Uma análise mais cuidadosa permite reconhecer que fatores sociais, por exemplo, podem ter afetado diretamente as estatísticas. Portanto, é válido refletir que a aplicação dessa abordagem em outras realidades sociais pode gerar resultados bem diferentes.

4. A adoção da abordagem didática proposta pode reduzir a curva de aprendizagem em programação?

A resposta para a questão de pesquisa em análise depende da capacidade do docente/pesquisador de observar e inferir. É necessário comparar o desempenho da turma atual com o desempenho das turmas anteriores na mesma disciplina.

No caso da Abordagem Construtivista de Programação, as diferenças em relação ao processo anterior são significativas, desde os conteúdos ministrados até a forma de avaliação. Por isso, não é possível fazer uma comparação direta entre os projetos desenvolvidos. É preciso avaliar o desempenho dos alunos no dia a dia, sua capacidade de raciocínio lógico e a solução final alcançada.

É importante destacar que os alunos tanto deste semestre quanto dos semestres anteriores começaram a disciplina com pouco ou nenhum conhecimento de programação. Nesse sentido, a resposta para a questão em análise está relacionada ao grau de evolução de cada estudante ao final do semestre.

Ao comparar o conhecimento inicial e final dos estudantes, incluindo aqueles que trancaram ou abandonaram a disciplina em um curto espaço de tempo, é possível afirmar que a abordagem utilizando programação com blocos proporcionou uma aprendizagem dos conceitos de programação mais rápida em comparação com a abordagem tradicional de ensino de programação. No primeiro mês da disciplina, os estudantes foram capazes de desenvolver animações e jogos na plataforma Scratch com pouca ou nenhuma ajuda do professor, utilizando conceitos de programação como condição, repetição, variáveis e operações. Ao final do semestre, os estudantes elaboraram projetos utilizando pseudocódigo e JAVA que empregavam, além dos conceitos mencionados anteriormente, funções, procedimentos e vetores.

Portanto, é possível afirmar que a Abordagem Construtivista de Programação promoveu uma aprendizagem mais rápida, preparando melhor os estudantes para os módulos seguintes e, em alguns casos, para sua busca por estágios. Os estudantes que experimentaram essa abordagem finalizaram a disciplina mais preparados para a continuidade do curso em comparação com os estudantes de semestres anteriores. Contudo, é importante considerar que os resultados podem variar de acordo com o perfil dos alunos, o contexto social e as condições específicas de cada turma. A Abordagem Construtivista de Programação pode se mostrar mais efetiva em alguns cenários, mas pode encontrar desafios em outros, portanto, é necessário um acompanhamento constante e avaliações

contínuas para ajustes e melhorias no processo de ensino e aprendizagem.

5. A adoção da abordagem didática proposta pode potencializar o comprometimento dos estudantes?

A Abordagem Construtivista de Programação é concebida com o intuito de promover um maior comprometimento dos estudantes, utilizando recursos lúdicos, jogos digitais e projetos autorais como forma de avaliação. A utilização de mídias digitais também é uma estratégia para aumentar o envolvimento dos estudantes no processo de aprendizagem.

O comprometimento dos estudantes pode ser percebido pelo docente/pesquisador no dia a dia, tanto nas atividades em sala de aula quanto na elaboração dos projetos e nas comunicações feitas pelo grupo de WhatsApp da turma. Além disso, os estudantes demonstraram buscar outras fontes de informação, como podcasts, vídeos no Youtube e cursos *online* com foco em programação.

Algumas falas registradas durante o grupo focal reforçaram as observações do professor/pesquisador, quando os estudantes mencionaram que empenharam mais esforço na disciplina para elaborar projetos de qualidade superior e superar suas dificuldades em programação.

Como mencionado na seção anterior, a qualidade dos projetos entregues pelos estudantes está diretamente relacionada ao esforço e comprometimento empregados na realização das tarefas. Nesse sentido, a elaboração de um jogo autoral como instrumento de avaliação, seja na plataforma de programação com blocos, em pseudocódigo ou em uma língua de programação, promoveu um maior comprometimento dos estudantes na realização da tarefa. Dessa forma, pode-se afirmar que a Abordagem Construtivista de Programação potencializou o comprometimento dos estudantes com a disciplina. É válido ressaltar que o interesse e a motivação dos estudantes também podem ser impulsionados pelo ambiente educacional e pelo incentivo à participação ativa, fatores que a Abordagem Construtivista de Programação busca incorporar em sua metodologia.

6.1 Recomendações práticas

A Abordagem Construtivista de Programação não foi concebida como um guia estritamente linear a ser seguido pelo docente. Ela foi idealizada para permitir ao educador incorporar os conceitos apresentados de maneira gradual, se necessário, no planejamento e implementação da disciplina que está ministrando. Nesse sentido, fornecemos algumas orientações visando simplificar sua aplicação.

1. Facilitar o processo de aprendizado ao começar pelo concreto e progredir para o abstrato. Essa abordagem possibilita que alunos iniciantes comecem com algo tangível, como compreender o conceito de repetição por meio dos passos de movimentação de

um personagem em um jogo. Associar o conceito de repetição a um resultado tangível imediatamente ajuda na assimilação desse mesmo conceito em uma linguagem de programação.

2. Adotar a utilização da programação com blocos, pois essa abordagem permite a criação de projetos de programação sem a necessidade de escrever linhas de código, mas sim com base no entendimento dos conceitos estruturais das linguagens de programação.
3. Seguir a trajetória de ensino começando pela programação com blocos, progredindo para os conceitos de linguagem de programação com o auxílio de uma pseudocódigo, como o Calango ou Portugol, e finalmente chegando a uma língua de programação completa, como o Java. Essa abordagem possibilita que o aluno se familiarize com o processo de tradução em cada fase, diminuindo as dificuldades no processo de aprendizado.
4. Eliminar ou diminuir o uso de exercícios e atividades exclusivamente baseados em problemas matemáticos. Incorporar situações do dia a dia do aluno ou relacionadas à área da computação, como criptografia ou criação de uma calculadora para churrascos, pode ampliar o interesse do estudante.
5. Integrar o uso de jogos, tanto digitais quanto convencionais, no processo de ensino e aprendizagem.
6. Incentivar a abordagem do aprendizado prático. Dado que a programação requer prática e repetição, é crucial motivar os alunos a praticar e aplicar ativamente o conhecimento adquirido nessa área.
7. Centralizar o aluno no cerne do processo de ensino e aprendizagem, substituindo o tradicional formato de aula expositiva, no qual o professor é o transmissor do conhecimento e o aluno é o receptor, por aulas focadas na discussão colaborativa.
8. Substituir a utilização de provas como método de avaliação por projetos que envolvam a autoria, ou seja, nos quais o aluno possa propor os temas de seus projetos com base em suas experiências.
9. Estimular a colaboração e o trabalho em equipe ao formar duplas ou grupos para a elaboração dos projetos, seja durante as aulas ou como parte da avaliação da disciplina.

A adoção das recomendações apresentadas nesta seção pode resultar em um processo de aprendizagem mais acessível para os estudantes novatos em programação, contribuindo para aumentar sua motivação e envolvimento com a disciplina. Contudo,

para alcançar esse resultado, o educador precisa reconsiderar suas abordagens atuais e ajustá-las para incorporar as novas práticas. Esse processo pode ser gradual e desafiador, mas é essencial que o docente esteja disposto a reexaminar suas estratégias de ensino na área de programação.

7 CONCLUSÃO

O processo de ensino e aprendizagem de programação de computadores é um grande desafio para professores e estudantes, sendo um dos maiores obstáculos em cursos de computação, informática e áreas afins. A falta de motivação dos estudantes pode ser apontada como um dos problemas enfrentados, o que pode levar à reprovação e evasão. Esta pesquisa busca oferecer uma solução que minimize esses problemas, proporcionando uma aprendizagem mais atrativa e prazerosa para estudantes iniciantes em programação.

Para realizar a pesquisa, utilizou-se o método Pesquisa-Formação, que parte das inquietações e problemáticas advindas da prática e da docência do professor-pesquisador. A Abordagem Construtivista de Programação foi aplicada em sala de aula, e as práticas docentes propostas foram utilizadas como método de avaliação da pesquisa. Foram coletados dados quantitativos por meio de questionários e diários de classe, e dados qualitativos a partir de um grupo focal com os estudantes e do diário de pesquisa. A interpretação dos dados quantitativos foi baseada na técnica de estatística descritiva, enquanto para os dados qualitativos foram seguidos os pressupostos indicados por Creswell, incluindo a preparação e organização e a identificação dos achados.

Após definirmos como conduzir a pesquisa, iniciamos a busca do referencial teórico para compreender como os conceitos relacionados à Computação, Comunicação e Educação se inter-relacionam. O referencial teórico proporcionou reflexões e discussões que nortearam a pesquisa. A partir do entendimento conceitual, foi possível planejar a solução para a pesquisa, composta pelos princípios norteadores, elaboração de um planejamento para a disciplina, nova organização da disciplina LP1 e avaliação da solução.

Realizamos a pesquisa no cotidiano dos estudantes e do pesquisador na disciplina Linguagem de Programação 1, do curso técnico em Informática da Escola Técnica Estadual Maria Mercedes Mendes Teixeira, durante o primeiro semestre de 2022. A avaliação da solução ocorreu no segundo semestre de 2022, com base nos dados já coletados por meio dos diferentes dispositivos, que serviram como insumo para discussão de cada uma das questões abordadas na pesquisa. Portanto, foi viável fornecer respostas para cada uma das questões de pesquisa da seguinte forma:

1. Uma nova abordagem didática no ensino de programação pode proporcionar aos estudantes um processo de aprendizagem mais simples, interessante e motivador?

A partir das observações do pesquisador em sala de aula, bem como pelas respostas fornecidas pelos estudantes em questionários e durante as sessões do grupo focal, é possível afirmar que a Aprendizagem Construtivista de Programação alcançou este

objetivo.

2. A adoção da abordagem didática proposta pode tornar a aprendizagem prazerosa e engajante?

A abordagem lúdica permite que o estudante veja a programação como uma ferramenta poderosa para criar coisas divertidas e úteis, tornando o aprendizado mais significativo e prazeroso.

A substituição da prova como instrumento de avaliação pela elaboração de projetos lúdicos e autorais contribuiu para engajar os estudantes, pois foram desafiados a construir projetos com temas que consideravam interessantes dentro de suas experiências e vivências.

3. A adoção da abordagem didática proposta pode reduzir o percentual de evasão e reprovação?

Não é possível afirmar de forma simples se a Abordagem Construtivista de Programação foi efetiva na redução da evasão e reprovação. É necessário considerar o contexto e possíveis variáveis externas. Uma análise mais cuidadosa permitiu reconhecer que fatores sociais afetaram diretamente os índices.

4. A adoção da abordagem didática proposta pode reduzir a curva de aprendizagem em programação?

A Aprendizagem Construtivista de Programação promoveu uma aprendizagem mais rápida, preparando melhor os estudantes para os módulos seguintes e, em alguns casos, para sua busca por estágios. Os estudantes que experimentaram essa abordagem finalizaram a disciplina mais preparados para a continuidade do curso em comparação com os estudantes de semestres anteriores.

5. A adoção da abordagem didática proposta pode potencializar o comprometimento dos estudantes?

A qualidade dos projetos entregues pelos estudantes está diretamente relacionada ao esforço e comprometimento empregados na realização das tarefas. O interesse e a motivação dos estudantes também podem ser impulsionados pelo ambiente educacional e pelo incentivo à participação ativa, fatores que a Abordagem Construtivista de Programação incorpora.

Concluimos que a Abordagem Construtivista de Programação e as práticas docentes sugeridas proporcionaram aos estudantes um processo de aprendizagem simples, interessante, motivador, prazeroso e engajante. Além disso, houve um aumento no comprometimento dos estudantes com a disciplina e redução na curva de aprendizagem em programação. No entanto, é importante ressaltar que a evasão e reprovação dos estudantes

na disciplina podem ser influenciadas por variáveis externas não controladas, tais como fatores sociais e econômicos. Portanto, é necessário avaliar o contexto em que a disciplina está inserida para uma conclusão mais precisa e considerar que a aplicação dessa abordagem em outras realidades sociais pode gerar resultados diferentes.

7.1 Contribuições da pesquisa

As principais contribuições desta pesquisa para a área de Computação e Educação foram:

1. Elaboração da Abordagem Construtivista de Programação, que defende o ensino de programação partindo do concreto ao abstrato.
2. Conceituação e definição de princípios norteadores para as práticas docentes.
3. Reflexões e discussões que fundamentaram a pesquisa e instigam a reflexão no leitor.
4. Aplicação do método Pesquisa-Formação, que proporcionou ao professor-pesquisador mudanças em suas práticas e aos sujeitos em formação (docentes e estudantes).
5. Geração de conhecimento sobre a importância da mudança das práticas e da abordagem no ensino de programação para a melhoria do processo de ensino e aprendizagem.
6. Recomendações práticas para adoção da Abordagem Construtivista de Programação.

7.2 Limitações e trabalhos futuros

Avaliamos apenas uma turma de ensino médio profissionalizante da rede pública no Rio de Janeiro nesta pesquisa. Embora o cenário tenha sido real, reconhecemos que os resultados obtidos podem ser limitados. Portanto, a utilização da Abordagem Construtivista de Programação e das práticas docentes em um período mais longo pode resultar em descobertas diferentes das nossas. Além disso, a aplicação com estudantes de ensino privado ou superior pode gerar resultados distintos.

Outra limitação que precisa ser considerada é que, por questões externas à pesquisa, precisamos adaptar o planejamento elaborado para atender aos dois grupos de estudantes que ingressaram na turma em períodos diferentes.

Como trabalhos futuros, pretendemos realizar estudos em diferentes contextos educacionais. Em primeiro lugar, planejamos trabalhar com novas turmas do professor-pesquisador nos próximos semestres, adotando as melhorias sugeridas nesta pesquisa. Em segundo lugar, vamos disseminar a solução entre os colegas da unidade e da rede, incentivando a reflexão sobre seus processos de ensino e aprendizagem em programação. Por

fim, vamos expandir a abordagem para docentes que atuam no ensino de programação em contextos diferentes, como no ensino fundamental, superior e em instituições privadas. Essas ações permitirão uma análise mais abrangente da efetividade da Abordagem Construtivista de Programação em diferentes cenários educacionais.

Referências

- ALMEIDA, B. O.; ALVES, L. R. G. Letramento digital em tempos de covid-19: uma análise da educação no contexto atual. *Debates em Educação*, v. 12, n. 28, p. 1–18, ago. 2020. Disponível em: <<https://www.seer.ufal.br/index.php/debateseducacao/article/view/10282>>. Citado na página 18.
- ALVES, G.; REBOUÇAS, A.; SCAICO, P. Coding dojo como prática de aprendizagem colaborativa para apoiar o ensino introdutório de programação: Um estudo de caso. In: *Anais do XXVII Workshop sobre Educação em Computação*. Porto Alegre, RS, Brasil: SBC, 2019. p. 276–290. ISSN 2595-6175. Disponível em: <<https://sol.sbc.org.br/index.php/wei/article/view/6636>>. Citado na página 57.
- AMARAL, M. M. d.; SANTOS, R. d.; SANTOS, E. Diários online como dispositivos de pesquisa-formação: uma conversa com o filme avatar. *Pro-Posições*, v. 33, p. e20190124, abr. 2022. Disponível em: <<https://periodicos.sbu.unicamp.br/ojs/index.php/proposic/article/view/8668977>>. Citado na página 88.
- ANGELO, D. F. dos S.; OLIVEIRA, N. S.; MAGALHÃES, N. M. de A.; SOUSA, M. N. A. de; ALMEIDA, E. P. de O. Metodologias ativas e sua implementação no processo de ensino-aprendizagem: uma revisão integrativa. *Editora Licuri*, p. 126–143, 2023. Citado 2 vezes nas páginas 50 e 52.
- AZEVEDO, L. C. R. de. *Compreendendo os discursos sobre adoção de um ambiente virtual de aprendizagem no Bacharelado de Sistemas de Informação da UNIRIO durante o ensino remoto*. 2022. UNIRIO. Citado na página 117.
- BELLUCCI, F. *Peirce's Speculative Grammar: Logic as Semiotics*. Routledge, 2019. (Routledge studies in American philosophy). ISBN 9780367887032. Disponível em: <<https://books.google.com.br/books?id=g7A5zAEACAAJ>>. Citado na página 39.
- BENNEDSEN, J.; CASPERSEN, M. E. Failure rates in introductory programming: 12 years later. *ACM Inroads*, Association for Computing Machinery, New York, NY, USA, v. 10, n. 2, p. 30–36, apr 2019. ISSN 2153-2184. Disponível em: <<https://doi.org/10.1145/3324888>>. Citado na página 20.
- BERBEL, N. As metodologias ativas e a promoção da autonomia dos estudantes (vol. 32). *Semina: Ciências Sociais e Humanas, Londrina*. Disponível em: www.uel.br/revistas/uel/index.php/seminasoc/article/view/10326/10999 Acesso em, v. 22, 2019. Citado na página 50.
- BISSONI, E.; MORETTO, M. Interação e diálogo como princípio de processos formativos: uma análise das narrativas de professores participantes do curso de formação em letramento em programação. *Horizontes*, v. 41, n. 1, p. e023013, jul. 2023. Disponível em: <<https://novoshorizontes.usf.emnuvens.com.br/horizontes/article/view/1581>>. Citado na página 49.
- BOSSE, Y.; GEROSA, M. Reprovações e trancamentos nas disciplinas de introdução à programação da universidade de são paulo: Um estudo preliminar.

In: *Anais do XXIII Workshop sobre Educação em Computação*. Porto Alegre, RS, Brasil: SBC, 2015. p. 426–435. ISSN 2595-6175. Disponível em: <<https://sol.sbc.org.br/index.php/wei/article/view/10259>>. Citado na página 20.

BRASIL. *LEI Nº 13.979, DE 6 DE FEVEREIRO DE 2020*. [S.l.], 2020. Disponível em: <<https://www.in.gov.br/en/web/dou/-/lei-n-13-979-de-6-de-fevereiro-de-2020-242078735>>. Acesso em: 12 ago. 2021. Citado na página 18.

BRASIL. *PORTARIA Nº 343, DE 17 DE MARÇO DE 2020*. [S.l.], 2020. Disponível em: <<https://www.in.gov.br/en/web/dou/-/portaria-n-343-de-17-de-marco-de-2020-248564376>>. Acesso em: 12 ago. 2021. Citado na página 19.

BRESSAN, M. L. Q.; AMARAL, M. A. Avaliando a contribuição do scratch para a aprendizagem pela solução de problemas e o desenvolvimento do pensamento criativo. *REVISTA INTERSABERES*, v. 10, n. 21, p. 509–526, dez. 2015. Disponível em: <<https://www.revistasuninter.com/intersaberes/index.php/revista/article/view/866>>. Citado na página 68.

CARVALHO, F.; PIMENTEL, M. Atividades autorais online: aprendendo com criatividade. In: . SBC Horizontes, 2020. ISSN 2175-9235. Disponível em: <<http://horizontes.sbc.org.br/index.php/2020/11/atividadesautorais/>>. Acesso em: 26 de maio de 2022. Citado 2 vezes nas páginas 27 e 28.

CECHINEL, A.; FONTANA, S. A. P.; DELLA, K. G. P.; PEREIRA, A. S.; PRADO, S. S. do. Estudo/análise documental: uma revisão teórica e metodológica. *Criar Educação*, v. 5, n. 1, 2016. Citado na página 91.

COELHO, J. A.; SOUZA, G. H.; ALBUQUERQUE, J. Desenvolvimento de questionários e aplicação na pesquisa em informática na educação. *Metodologia de Pesquisa em Informática na Educação: Abordagem Quantitativa de Pesquisa*. Porto Alegre: SBC. *Série Metodologia de Pesquisa em Informática na Educação*, SBC, v. 2, 2020. Citado 2 vezes nas páginas 89 e 90.

CRESWELL, J. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. SAGE Publications, 2014. ISBN 9781452226101. Disponível em: <https://books.google.com.br/books?id=4uB76IC_pOQC>. Citado na página 107.

CYSNEIROS, P. G. Papert, seymour. a máquina das crianças: repensando a escola na era da informática. *Revista Entreideias: educação, cultura e sociedade*, v. 12, n. 12, jul. 2008. Disponível em: <<https://periodicos.ufba.br/index.php/entreideias/article/view/2971>>. Citado na página 33.

ESTÁCIO, B. J. da S.; PRIKLADNICKI, R. A utilização da programação em par distribuída no ensino de programação.”. 2013. Citado na página 59.

EZENWOYE, O. What language? - the choice of an introductory programming language. In: *2018 IEEE Frontiers in Education Conference (FIE)*. [S.l.: s.n.], 2018. p. 1–8. Citado na página 80.

FERNANDES, A. L. M.; FERREIRA, M.; CALVOSA, M. V. D. A proposta de um framework para analisar o comportamento do cliente digital por meio do business model canvas. *Revista de Casos e Consultoria*, v. 12, n. 1, p. e26972, dez. 2021. Disponível em: <<https://periodicos.ufrn.br/casoseconsultoria/article/view/26972>>. Citado na página 71.

FERREIRA, A.; MELHOR, A.; BARRETO, J.; PAIVA, L.; MATOS, E. Experiência prática interdisciplinar do raciocínio computacional em atividades de computação desplugada na educação básica. In: *Anais do XXI Workshop de Informática na Escola*. Porto Alegre, RS, Brasil: SBC, 2015. p. 256–265. ISSN 0000-0000. Disponível em: <<https://sol.sbc.org.br/index.php/wie/article/view/16525>>. Citado na página 36.

FERREIRA, R.; BRAGA, M. Ensino de programação e as estratégias pedagógicas utilizadas no brasil. *Revista Contexto & Educação*, v. 38, n. 120, p. e11377, jun. 2023. Disponível em: <<https://revistas.unijui.edu.br/index.php/contextoeducacao/article/view/11377>>. Citado na página 52.

FONSECA, A. B. da S.; TELES, D. R. S.; SOUSA, M. C. de B.; LOYOLA, M. E. S. Célestin freinet na luta por uma pedagogia aliada ao meio social. *Revista Profissão Docente*, v. 5, n. 11, 2005. Citado na página 50.

FREIRE, P. Pedagogia do oprimido. rev. e atual. *Rio de Janeiro: Paz e Terra*, p. 95–101, 2011. Citado na página 50.

FREIRE, P. *Pedagogia da autonomia: Saberes necessários à prática educativa*. Paz e Terra, 2014. ISBN 9788577532261. Disponível em: <<https://books.google.com.br/books?id=Ae4nAwAAQBAJ>>. Citado na página 85.

GOMES, G.; NISHI, Y.; RAMOS, P.; SILVA, L.; GUERRA, E.; WIESE, I.; BOSSE, Y. Um recurso educacional para desenvolver a habilidade da percepção de padrões de equívocos com aprendizes de programação. In: *Anais do III Simpósio Brasileiro de Educação em Computação*. Porto Alegre, RS, Brasil: SBC, 2023. p. 328–336. ISSN 0000-0000. Disponível em: <<https://sol.sbc.org.br/index.php/educomp/article/view/23903>>. Citado na página 20.

GROTTA, A.; PRADO, E. P. Um ensaio sobre a experiência educacional na programação de computadores: a abordagem tradicional versus a aprendizagem baseada em projetos. In: *Anais do XXVI Workshop sobre Educação em Computação*. Porto Alegre, RS, Brasil: SBC, 2018. ISSN 2595-6175. Disponível em: <<https://sol.sbc.org.br/index.php/wei/article/view/3496>>. Citado na página 60.

GUARDA, G.; REZENDE, S. de; PINTO, S. Ludicidade como abordagem pedagógica para o ensino de algoritmos de pesquisa sequencial e binária. In: *Anais do II Simpósio Brasileiro de Educação em Computação*. Porto Alegre, RS, Brasil: SBC, 2022. p. 118–124. ISSN 0000-0000. Disponível em: <<https://sol.sbc.org.br/index.php/educomp/article/view/19205>>. Citado na página 64.

HERNÁNDEZ, R. V. R.; DOMÍNGUEZ, J. C.; MENDOZA, A. L.; IZAGUIRRE, J. A. H.; IBARRA, C. M. J. Comparison of pair and solo programming through software metrics in university students' projects. *RIDE. Rev. Iberoam. Investig. Desarro. Educ, Guadalajara*, v. 10, n. 19, 2019. Disponível em: <http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S2007-74672019000200030&lng=es&nrm=iso>. Citado na página 59.

HOLANDA, W. Duarte de; FREIRE, L. de P.; COUTINHO, J. Cássia da S. Estratégias de ensino-aprendizagem de programação introdutória no ensino superior: uma revisão sistemática da literatura. *Revista Novas Tecnologias na Educação*, v. 17, n. 1, p. 527–536, jul. 2019. Disponível em: <<https://seer.ufrgs.br/index.php/renote/article/view/95905>>. Citado na página 44.

JENKINS, T. On the difficulty of learning to program. In: CITESEER. *Proceedings of the 3rd Annual Conference*. 2002. (LTSN Centre for Information and Computer Sciences, 2002), p. 53–58. Disponível em: <<https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=ed6b9c9da3ddea0f884351f98155b051ec371dc3>>. Citado na página 21.

JÚNIOR, G. P. S. *Pensamento Computacional com a Educação On-line no Ensino Médio Integrado*. 348 p. Tese (Doutorado) — Universidade Federal de Sergipe, 2023. Citado na página 35.

KAMPFF, A. J. C.; LOPES, T.; ALVES, I. M.; SOUZA, V. C. de; RIGO, S.; MARSON, F. Pensamento computacional no ensino superior: Relato de uma oficina com professores da universidade do vale do rio dos sinos. In: *Anais dos Workshops do V Congresso Brasileiro de Informática na Educação (CBIE 2016)*. [s.n.], 2016. v. 5, n. 1, p. 1316. Disponível em: <<https://walgprog.gp.utfpr.edu.br/2016/assets/arquivos/artigos/S4A4.pdf>>. Citado na página 34.

LEITE, K. N. S.; SOUSA, M. N. A. de; NASCIMENTO, A. K. F.; SOUZA, T. A. de. Utilização da metodologia ativa no ensino superior da saúde: revisão integrativa. *Arquivos de Ciências da Saúde da UNIPAR*, v. 25, n. 2, 2021. Citado na página 50.

MABEL, E.; TAVARES, J.; BURI, Y.; PEREIRA, M. Codetesting: Sistema de apoio ao ensino de lógica de programação. In: *Anais Estendidos do III Simpósio Brasileiro de Educação em Computação*. Porto Alegre, RS, Brasil: SBC, 2023. p. 22–23. ISSN 0000-0000. Disponível em: <https://sol.sbc.org.br/index.php/educomp_estendido/article/view/24141>. Citado na página 20.

MACHADO, L. D. P.; BERKENBROCK, C. D. M.; FREITAS, L. E. R. d. Uma abordagem colaborativa para aprendizagem de programação de computadores com a utilização de dispositivos móveis. *Revista de Sistemas de Informação da FSMA*, n. 24, p. 39–54, 2019. Citado na página 57.

MARQUES, D.; COSTA, L.; SILVA, M.; REBOUÇAS, A. Atraindo alunos do ensino médio para a computação: Uma experiência prática de introdução a programação utilizando jogos e python. In: *Anais do XVII Workshop de Informática na Escola*. Porto Alegre, RS, Brasil: SBC, 2011. p. 1138–1147. ISSN 0000-0000. Disponível em: <<https://sol.sbc.org.br/index.php/wie/article/view/21724>>. Citado na página 66.

MATOS, E.; COUTINHO, C.; SANTOS, E.; SANTOS, J. dos; AZEVEDO, L.; TAVARES, G.; ZABOT, D. Desenvolvendo e descobrindo competências no currículo escolar por meio do raciocínio computacional. In: *Anais do Simpósio Brasileiro de Educação em Computação*. Porto Alegre, RS, Brasil: SBC, 2021. p. 255–262. ISSN 0000-0000. Disponível em: <<https://sol.sbc.org.br/index.php/educomp/article/view/14492>>. Citado na página 37.

MEDEIROS, J. O.; RIBEIRO, R. do C.; SOUSA, M. N. A. de. Mapa conceitual como ferramenta de aprendizagem: revisão integrativa da literatura. *SANARE-Revista de Políticas Públicas*, v. 19, n. 2, 2020. Citado na página 50.

MEDEIROS, R. P. *Hello, world: uma análise sobre dificuldades no ensino e na aprendizagem de introdução à programação nas universidades*. 193 p. Tese (Doutorado) — Universidade Federal de Pernambuco, 2019. Disponível em: <<https://repositorio.ufpe.br/handle/123456789/34464>>. Citado na página 20.

MONCLAR, R. S.; SILVA, M. A.; XEXÉO, G. Jogos com propósito para o ensino de programação. *Anais do XVII Simpósio Brasileiro de Jogos e Entretenimento Digital - SBGames, Foz do Iguaçu - PR*, p. 1132–1140, 11 2018. Citado na página 65.

MOREIRA, M. V. C.; SIMIÃO-FERREIRA, J.; DE-CARVALHO, P. S. Metodologias ativas e estratégias para estimular o engajamento criativo dos estudantes por meio do design thinking e tecnologias digitais. *v. XVI. n. 2 (2022): REVISTA ANÁPOLIS DIGITAL - ISSN 2178-0722*, p. 98–110, dez. 2022. Disponível em: <<https://portaleducacao.anapolis.go.gov.br/revistaanapolis/wp-content/uploads/2023/03/a7.pdf>>. Citado na página 55.

NUNES, M. M.; SOUSA, T.; MARTINHAGO, A.; NUNES, M. M. Uso da lógica de programação para potencializar o pensamento criativo em crianças do ensino básico. In: SBC. *Anais dos Workshops do Congresso Brasileiro de Informática na Educação, Fortaleza - CE*. 2018. v. 7, n. 1, p. 325–328. Disponível em: <<http://www.repositorio.ufc.br/handle/riufc/44007>>. Citado na página 68.

NÓVOA, A. Prefácio. In: JOSSO, MARIE-CHRISTINE. *Experiências de vida e formação*. São Paulo: Cortez Editora, 2004. p. 11–34. Citado na página 86.

OLIVEIRA, A. C. A. de. A contribuição do design thinking na educação. *Revista e-TECH: Tecnologias para Competitividade Industrial - ISSN - 1983-1838*, p. 105–121, dez. 2014. Disponível em: <<https://etech.emnuvens.com.br/revista-cientifica/article/view/454>>. Citado na página 54.

OLIVEIRA, E. d. S. G. de; VASCONCELLOS, P. R. Grupo focal em informática na educação: diálogo, conflito, consenso? p. 24, 2021. Citado 2 vezes nas páginas 92 e 93.

OLIVEIRA, S. L. d.; SIQUEIRA, A. F.; ROMÃO, E. C. Aprendizagem baseada em projetos no ensino médio: estudo comparativo entre métodos de ensino. *Bolema: Boletim de Educação Matemática*, UNESP - Universidade Estadual Paulista, Pró-Reitoria de Pesquisa, Programa de Pós-Graduação em Educação Matemática, v. 34, n. 67, p. 764–785, May 2020. ISSN 0103-636X. Disponível em: <<https://doi.org/10.1590/1980-4415v34n67a20>>. Citado na página 60.

PAPERT, S. *Mindstorms: Children, Computers, and Powerful Ideas*. USA: Basic Books, Inc., 1980. Citado na página 30.

PAPERT, S.; VALENTE, J.; BITELMAN, B. *Logo: computadores e educação*. Brasiliense, 1980. Disponível em: <<https://books.google.com.br/books?id=nGgOHQAACAAJ>>. Citado na página 32.

RABELO, A.; MAIA, L. C. G.; PARREIRAS, F. S. Performance analysis of computer science students in programming learning. In: *Anais do XXVI Workshop sobre Educação em Computação*. Porto Alegre, RS, Brasil: SBC, 2018. ISSN 2595-6175. Disponível em: <<https://sol.sbc.org.br/index.php/wei/article/view/3499>>. Citado na página 20.

RIBEIRO, L.; CASTRO, A.; FRÖHLICH, A. A.; FERRAZ, C. A. G.; FERREIRA, C. E.; SEREY, D.; CORDEIRO, D. de A.; AIRES, J.; BIGOLIN, N.; CAVALHEIRO, S. Diretrizes da sociedade brasileira de computação para o ensino de computação na educação básica. *Sociedade Brasileira de Computação*, 2019. Relatório Técnico nº 001/2019. Disponível em: <<https://sol.sbc.org.br/livros/index.php/sbc/catalog/view/60/263/505-1>>. Citado na página 30.

RIBEIRO, N.; MARTINS, L.; BERSANETTE, J. Panorama brasileiro do ensino e aprendizagem de programação de computadores na educação básica. In: *Anais do XXX Workshop sobre Educação em Computação*. Porto Alegre, RS, Brasil: SBC, 2022. p. 346–356. ISSN 2595-6175. Disponível em: <<https://sol.sbc.org.br/index.php/wei/article/view/20843>>. Citado 2 vezes nas páginas 44 e 46.

SÁ-SILVA, J. R.; ALMEIDA, C. D. d.; GUINDANI, J. F. et al. Pesquisa documental: pistas teóricas e metodológicas. *Revista brasileira de história & ciências sociais*, São Leopoldo, v. 1, n. 1, p. 1–15, 2009. Citado na página 90.

SANTAELLA, L. *Neo-Humano: A Sétima Revolução Cognitiva do Sapiens*. Paulus Editora, 2022. (Comunicação). ISBN 9786555627121. Disponível em: <<https://books.google.com.br/books?id=44qKEAAQBAJ>>. Citado na página 39.

SANTOS, C. *Estatística Descritiva : Manual de auto-aprendizagem*. [S.l.: s.n.], 2007. v. 2. ISBN 9789726184669. Citado na página 107.

SANTOS, E. *Pesquisa-Formação Na Cibercultura*. Independently Published, 2019. ISBN 9781793247100. Disponível em: <https://books.google.com.br/books?id=C13_wAEACAAJ>. Citado na página 85.

SANTOS, L. R.; ANDRADE, E. L. de M.; FERNANDES, J. C. da C.; LIMA, E. F. de. As contribuições da teoria da aprendizagem de lev vygotsky para o desenvolvimento da competência em informação. *Revista Brasileira de Biblioteconomia e Documentação*, v. 17, p. 1–15, 2021. Citado 2 vezes nas páginas 126 e 127.

SANTOS, V. A. D.; DANTAS, V. R.; GONÇALVES, A. B. V.; HOLANDA, B.; BARBOSA, A. de Andrade GAIÃO e. O uso das ferramentas digitais no ensino remoto acadêmico: desafios e oportunidades na perspectiva docente. In: *Anais VII CONEDU - Edição Online*. [s.n.], 2020. p. 15–17. Disponível em: <<https://editorarealize.com.br/artigo/visualizar/69166>>. Citado na página 19.

SEBOLD, L. F.; MARTINS, F. E.; ROSA, R. da; CARRARO, T. E.; MARTINI, J. G.; KEMPFER, S. S. Metodologias ativas: uma inovação na disciplina de fundamentos para o cuidado profissional de enfermagem. *Cogitare Enfermagem*, v. 15, n. 4, 2010. Citado na página 50.

SEGURA, E.; KALHIL, J. B. A metodologia ativa como proposta para o ensino de ciências. *REAMEC-Rede Amazônica de Educação em Ciências e Matemática*, v. 3, n. 1, p. 87–98, 2015. Citado na página 51.

SELBY, C.; WOOLLARD, J. Computational thinking: the developing definition. In: . University of Southampton (E-prints), 2013. Disponível em: <<https://eprints.soton.ac.uk/356481/>>. Citado na página 30.

SILVA, C.; MATOS, E.; ZABOT, D.; TAVARES, G.; SANTOS, J. De "vale nota" a "jogo é prova!": raciocínio computacional e protagonismo docente como estímulo ao protagonismo discente no ensino de língua portuguesa. In: *Anais do XXX Workshop sobre Educação em Computação*. Porto Alegre, RS, Brasil: SBC, 2022. p. 133–144. ISSN 2595-6175. Disponível em: <<https://sol.sbc.org.br/index.php/wei/article/view/20825>>. Citado na página 37.

SILVA, G.; SANTOS, G.; RISSOLI, V. A importância da linguagem nativa para a aprendizagem significativa em lógica de programação. In: *Anais do XXXI Simpósio Brasileiro de Informática na Educação*. Porto Alegre, RS, Brasil: SBC, 2020. p. 1803–1812. ISSN 0000-0000. Disponível em: <<https://sol.sbc.org.br/index.php/sbie/article/view/12936>>. Citado na página 80.

SILVA, W. A. d.; KALHIL, J. B.; NICOT, Y. E. Uma análise comparativa das abordagens metodológicas que podem sustentar a utilização das tecnologias no processo de ensino e aprendizagem de ciências. *REAMEC - Rede Amazônica de Educação em Ciências e Matemática*, v. 3, n. 1, p. 5–24, dez. 2015. Disponível em: <<https://periodicoscientificos.ufmt.br/ojs/index.php/reamec/article/view/5303>>. Citado na página 33.

SOARES, M. Letramento e alfabetização: as muitas facetas. *Revista Brasileira de Educação*, SciELO Brasil, p. 5–17, 04 2004. Citado na página 47.

TORRES, P.; IRALA, E. Aprendizagem colaborativa: teoria e prática. In: *COMPLEXIDADE: REDES E CONEXÕES NA PRODUÇÃO DO CONHECIMENTO*. CURITIBA: SENAR. [S.l.], 2014. p. 61–94. ISBN 9788575651094. Citado na página 57.

UGULINO, W.; MARQUES, A. d. M.; PIMENTEL, M.; SIQUEIRA, S. W. Avaliação colaborativa: um estudo com a ferramenta moodle workshop. *XX Simpósio Brasileiro de Informática na Educação, Florianópolis/SC*, 2009. Citado na página 121.

UNESCO. *COVID-19 impact on education*. 2020. Disponível em: <<https://en.unesco.org/covid19/educationresponse>>. Acesso em: 12 ago. 2021. Citado na página 19.

VEIGA, I. P. A.; CASTANHO, M. E.; AMARAL, A. L. *Pedagogia universitária: a aula em foco*. [S.l.]: Papyrus Editora, 2000. Citado na página 50.

VELOSO, M. M. S. d. A.; BONILLA, M. H. S.; PRETTO, N. D. L. A cultura da liberdade de criação e o cerceamento tecnológico e normativo: potencialidades para a autoria na educação. *ETD Educação Temática Digital*, Scielo, v. 18, p. 43 – 59, 04 2016. ISSN 1676-2592. Disponível em: <http://educa.fcc.org.br/scielo.php?script=sci_arttext&pid=S1676-25922016000100043&nrm=iso>. Citado na página 62.

VIGOTSKY, L. *Formacao Social Da Mente*, a. Martins Fontes, 2007. (Psicologia e pedagogia). ISBN 9788533622647. Disponível em: <<https://books.google.com.br/books?id=ukbkPgAACAAJ>>. Citado na página 125.

WATSON, C.; LI, F. W. Failure rates in introductory programming revisited. In: . New York, NY, USA: Association for Computing Machinery, 2014. (ITiCSE '14), p. 39–44. ISBN 9781450328333. Disponível em: <<https://doi.org/10.1145/2591708.2591749>>. Citado 2 vezes nas páginas 20 e 21.

WELSH, M. *OMS declara emergência de saúde pública de importância internacional por surto de novo coronavírus*. 2020. Disponível em: <<https://www.paho.org/pt/news/30-1-2020-who-declares-public-health-emergency-novel-coronavirus>>. Acesso em: 12 ago. 2021. Citado na página 18.

WILDER-SMITH, A.; FREEDMAN, D. O. Isolation, quarantine, social distancing and community containment: pivotal role for old-style public health measures in the novel coronavirus (2019-nCoV) outbreak. *Journal of Travel Medicine*, v. 27, n. 2, 02 2020. ISSN 1708-8305. Disponível em: <<https://doi.org/10.1093/jtm/taaa020>>. Citado na página 18.

WING, J. M. Computational thinking. *Commun. ACM*, Association for Computing Machinery, New York, NY, USA, v. 49, n. 3, p. 33–35, mar 2006. ISSN 0001-0782. Disponível em: <<https://doi.org/10.1145/1118178.1118215>>. Citado na página 30.

WING, J. M. 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. O trabalho foi traduzido para o português por Cleverson Sebastião dos Anjos, professor da área de informática do Instituto Federal do Paraná. Disponível em: <<http://dx.doi.org/10.3895/rbect.v9n2.4711>>. Citado na página 30.

ZANETTI, H.; BORGES, M.; RICARTE, I. Pensamento computacional no ensino de programação: Uma revisão sistemática da literatura brasileira. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. [S.l.: s.n.], 2016. v. 27, n. 1, p. 21. Citado na página 78.

APÊNDICE A . Componente Curricular LP1

Competências a serem desenvolvidas:

- Desenvolver algoritmos através de divisão modular e refinamentos sucessivos.
- Interpretar pseudocódigos, algoritmos e outras especificações para codificar programas.
- Avaliar resultados de testes dos programas desenvolvidos.
- Integrar módulos desenvolvidos separadamente.
- Compreender o paradigma de orientação por objeto e sua aplicação em programação.
- Compreender a linguagem Java e ambientes de programação.
- Distinguir e avaliar linguagens e ambientes de programação, aplicando-os no desenvolvimento de software.

Habilidades:

- Selecionar e utilizar estruturas de dados na resolução de problemas computacionais.
- Utilizar editores de textos, compiladores e/ou interpretadores e ambientes de desenvolvimento na elaboração de programas (IDE), utilizando a linguagem orientada a objeto.
- Utilizar modelos, pseudocódigos e ferramentas na representação da solução de problemas.
- Elaborar e executar casos e procedimentos de testes de programas.
- Redigir instruções de uso dos programas implementados.
- Utilizar editores de textos, compiladores e/ou interpretadores e ambientes de desenvolvimento na elaboração de programas (IDE).

Conteúdo Programático:

- **Introdução ao conceito de algoritmo:** Ambiente da ferramenta de programação.
- **Tópicos preliminares:** Tipos Primitivos. Constantes. Variáveis. Expressões Aritméticas e Lógicas. Comandos de Entrada e Saída.

- **Estrutura Sequencial.**
- **Estruturas de Seleção:** Seleção Simples. Seleção Composta. Seleção Encadeada.
- **Estruturas de Repetição:** Repetição com teste no início. Repetição com teste no final. Repetição com variável de controle.
- **Estruturas de dados:** Variáveis Compostas Homogêneas. Variáveis Compostas Unidimensionais. Variáveis Compostas Multidimensionais. Conceitos Básicos de Ponteiro.
- **Noções de Módulos:** Sub-rotinas (Procedimentos e Funções).

APÊNDICE B . Novo planejamento da disciplina LP1



GOVERNO DO ESTADO DO RIO DE JANEIRO
SECRETARIA DE ESTADO DE CIÊNCIA, TECNOLOGIA E INOVAÇÃO
FUNDAÇÃO DE APOIO À ESCOLA TÉCNICA - FAETEC
ESCOLA TÉCNICA ESTADUAL MARIA MERCEDES MENDES TEIXEIRA - ETEMMT

CURSO TÉCNICO – SUBSEQUENTE DE INFORMÁTICA
PLANO DE DISCIPLINA – ANO LETIVO: 2022

LINGUAGEM DE PROGRAMAÇÃO I

1. COMPETÊNCIAS

- Conhecer as estruturas das linguagens de programação;
- Se expressar em diferentes línguas (blocos, fluxograma, pseudocódigo e língua JAVA);

2. HABILIDADES

- Desenvolver a habilidade em decompor, reconhecer padrões, abstrair e projetar algoritmos por meio de desenvolvimento de programas computacionais simples;
- Desenvolver a habilidade de aplicar as estruturas da linguagem de programação para o desenvolvimento de sistemas computacionais simples;
- Desenvolver a habilidade em programar na língua Java;

3. MATERIAL DIDÁTICO

- Apostila de referência disponibilizada pelo professor;
- Sites da internet;
- Vídeos da plataforma Youtube;
- Computadores;
- Quadro;
- Recursos multimídias;
- Google Classroom;
- Estacionamento Algorítmico;

4. REFERÊNCIAS BIBLIOGRÁFICAS

- Hora do código <https://hourofcode.com/br>
- Lightbot <https://lightbot.lu/>
- Scratch <https://scratch.mit.edu/>
- Calango https://www.sae.unb.br/cae/conteudo/unbfga/cb/new_ambientedealgoritmo.html
- JAVA <https://docs.oracle.com/en/java/>
- Exact Instructions Challenge https://www.youtube.com/watch?v=cDA3_5982h8
- Programação: O que é Algoritmo <https://www.youtube.com/watch?v=68ZbfArHKw8>



GOVERNO DO ESTADO DO RIO DE JANEIRO
SECRETARIA DE ESTADO DE CIÊNCIA, TECNOLOGIA E INOVAÇÃO
FUNDAÇÃO DE APOIO À ESCOLA TÉCNICA - FAETEC
ESCOLA TÉCNICA ESTADUAL MARIA MERCEDES MENDES TEIXEIRA - ETEMMT

CURSO TÉCNICO – SUBSEQUENTE DE INFORMÁTICA
PLANO DE DISCIPLINA – ANO LETIVO: 2022

LINGUAGEM DE PROGRAMAÇÃO I

5. ESTRATÉGIAS

- Aprender fazendo;
- Programação com blocos;
- Aprendizagem colaborativa;
- Aprendizagem baseada em projetos;
- Aprendizagem através de jogos;

6. CARGA HORÁRIA

Carga horária semanal: **8 tempos**

Duração do curso: **22 semanas**

7. INSTRUMENTOS DE AVALIAÇÃO

O aluno será avaliado pelo uso dos instrumentos listados abaixo:

- **Projeto do módulo** – Projeto autoral elaborado pelo aluno. Está avaliação vale 10,0 pontos para formação da média.

8. CRITÉRIOS DE AVALIAÇÃO

Será considerado aprovado na disciplina o aluno que obtiver média, igual ou superior a 6,0 (seis).

(Se Média = $(1MB + 2MB) / 2 = 6,0$ Então: Aluno Aprovado).

O aluno que obtiver média aritmética inferior a 6,0 (seis), será submetido à Recuperação Final (RF). (Se Média = $(1MB + 2MB) / 2 < 6,0$ Então: Aluno em RF).

Para os alunos que ficaram para RF, será considerado aprovado na disciplina, o aluno que alcançar média igual ou superior a 6,0 (seis). (Se Média Final $> 6,0$ Então: Aluno Aprovado Senão: Aluno Reprovado).



GOVERNO DO ESTADO DO RIO DE JANEIRO
SECRETARIA DE ESTADO DE CIÊNCIA, TECNOLOGIA E INOVAÇÃO
FUNDAÇÃO DE APOIO À ESCOLA TÉCNICA - FAETEC
ESCOLA TÉCNICA ESTADUAL MARIA MERCEDES MENDES TEIXEIRA - ETEMMT

CURSO TÉCNICO – SUBSEQUENTE DE INFORMÁTICA
PLANO DE DISCIPLINA – ANO LETIVO: 2022

LINGUAGEM DE PROGRAMAÇÃO I

9. CONTEÚDO PROGRAMÁTICO (OBS.: CADA AULA CORRESPONDE A 8 (OITO) TEMPOS DE AULA)

	DESCRIÇÃO
Semana 1	PLANEJAMENTO
Semana 2	PLANEJAMENTO
Semana 3	BOAS-VINDAS AOS ALUNOS; ALINHAMENTO DE EXPECTATIVAS
Semana 4	INTRODUÇÃO; A HORA DO CÓDIGO; LIGHTBOT
Semana 5	ALGORITMO; PENSAMENTO COMPUTACIONAL
Semana 6	SCRATCH: INTRODUÇÃO; ELABORAÇÃO DE UM DESENHO QUALQUER
Semana 7	SCRATCH: EXPLORAR JOGOS; VARIÁVEIS E ESTRUTURAS DE REPETIÇÃO E CONDIÇÃO
Semana 8	SCRATCH: TRABALHAR COM UM JOGO; IMPLEMENTAR PROJETO DE JOGO
Semana 9	SCRATCH: PROJETO DE JOGO
Semana 10	SCRATCH: PROJETO DE JOGO
Semana 11	AVALIAÇÃO BIMESTRAL: APRESENTAÇÃO DO PROJETO DE JOGO
Semana 12	LINGUAGEM DE PROGRAMAÇÃO; CALANGO: ESTRUTURA BÁSICA; ESTRUTURA SE E ESCOLHA
Semana 13	CALANGO: ESTRUTURA ENQUANTO E PARA; PROCEDIMENTOS E FUNÇÕES
Semana 14	CALANGO: FUNÇÕES, VETORES E MATRIZES
Semana 15	JAVA: INTRODUÇÃO E ESTRUTURAS DE CONTROLE
Semana 16	JAVA: ESTRUTURAS DE REPETIÇÃO E FUNÇÕES



GOVERNO DO ESTADO DO RIO DE JANEIRO
SECRETARIA DE ESTADO DE CIÊNCIA, TECNOLOGIA E INOVAÇÃO
FUNDAÇÃO DE APOIO À ESCOLA TÉCNICA - FAETEC
ESCOLA TÉCNICA ESTADUAL MARIA MERCEDES MENDES TEIXEIRA - ETEMMT

CURSO TÉCNICO – SUBSEQUENTE DE INFORMÁTICA
PLANO DE DISCIPLINA – ANO LETIVO: 2022

LINGUAGEM DE PROGRAMAÇÃO I

	DESCRIÇÃO
Semana 17	JAVA: VETORES E MATRIZES
Semana 18	CALANGO E JAVA: PROJETO DE JOGO
Semana 19	CALANGO E JAVA: PROJETO DE JOGO
Semana 20	AVALIAÇÃO BIMESTRAL
Semana 21	REVISÃO
Semana 22	AVALIAÇÃO FINAL

10. PROJETO INTERDISCIPLINAR

Professores da disciplina de LINGUAGEM DE PROGRAMAÇÃO I

APÊNDICE C . Atividade Estacionamento Algorítmico

Estacionamento Algorítmico



Antes de iniciar:

1. Recorte o tabuleiro;
2. Recorte os carros nas linhas tracejadas;
3. Note que cada carro é identificado por uma letra (A-K e X) e cada caminhão é identificado por um número (1-4).

Objetivo do jogo: Retirar o carro vermelho (letra X) pela lateral direita (Saída) sem bater ou passar por cima dos demais carros e caminhões estacionados.

Instruções:

1. Escolha um dos desafios propostos;
2. Posicione os veículos conforme mostrado no desafio, respeitando as letras, números, posição e sentido de cada carro no tabuleiro;
3. Usando apenas os comandos (↑, ↓, →, ←), mova os veículos no sentido estacionado, ou seja, carros que estão no sentido vertical só podem andar verticalmente e carros estacionados na horizontal só podem andar horizontalmente. Não é permitido trocar o sentido do veículo ou fazer curvas.
4. Anote em uma folha¹ qual o veículo que você utilizou, quantas vezes ele se moveu e em que direção, até você conseguir remover o carro X do estacionamento. Dessa forma, você cria uma sequência de instruções de como solucionar um problema. Por exemplo:

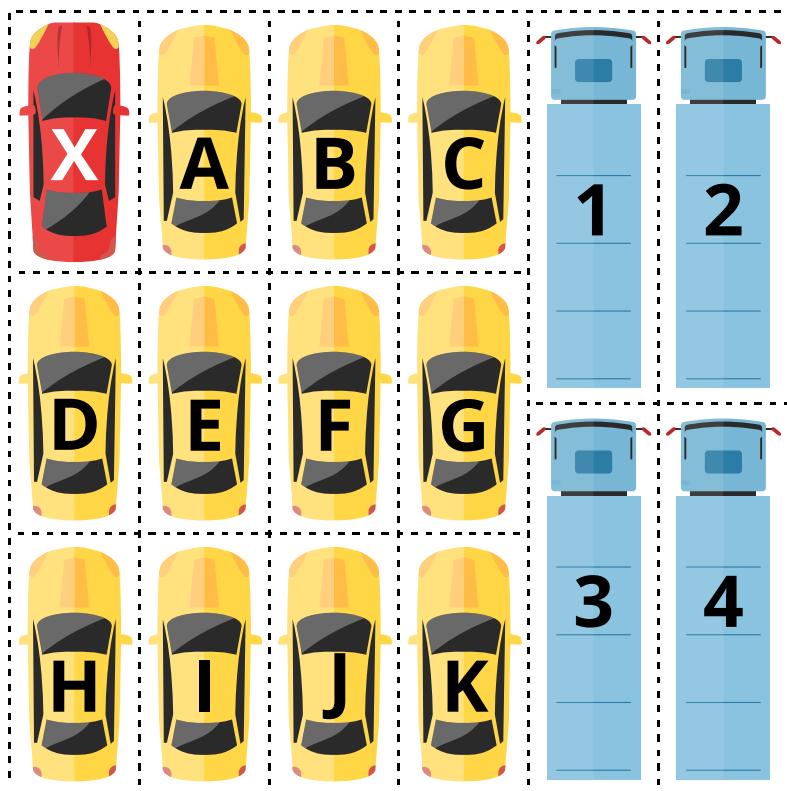
VEÍCULO	MOVIMENTOS				
C	←	←	←		
1	↓	↓	↓		
A	→				
2	↑				
B	↑				
4	←	←			
3	↓	↓			
X	→	→	→	→	

5. Quando você já estiver seguro de como funciona, tente usar loopings, como por exemplo:

VEÍCULO	MOVIMENTOS				
X	4x	→			

6. Após finalizado cada etapa, experimente resolver o desafio seguinte ou tentar resolver com uma quantidade de instruções menor.

¹ Você pode utilizar o formulário entregue pelo professor ou fazer as anotações no seu caderno

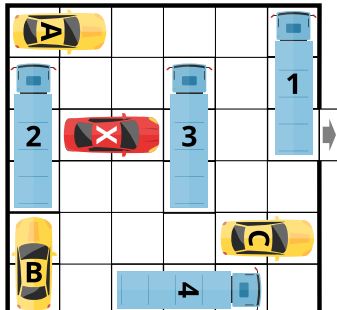


**PENSAMENTO
COMPUTACIONAL
PARA TODOS**
www.computacional.com.br

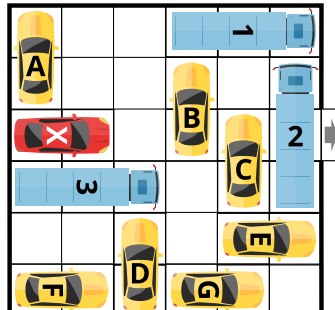


NÍVEL INICIANTE

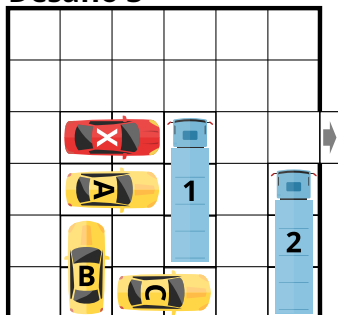
Desafio 1



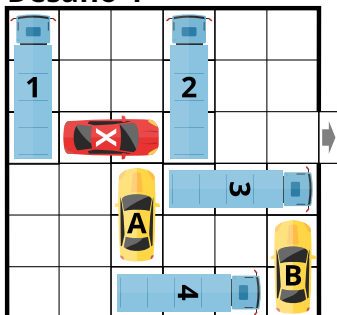
Desafio 2



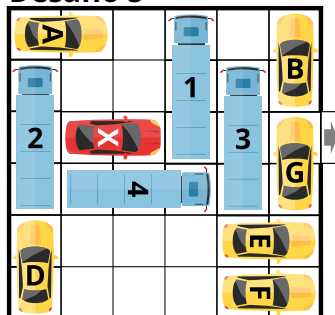
Desafio 3



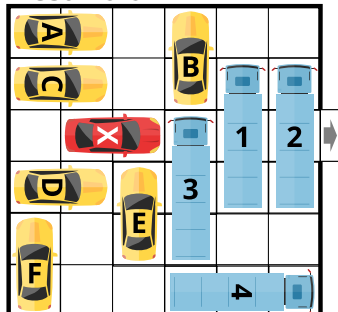
Desafio 4



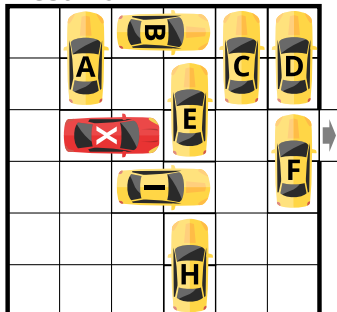
Desafio 5



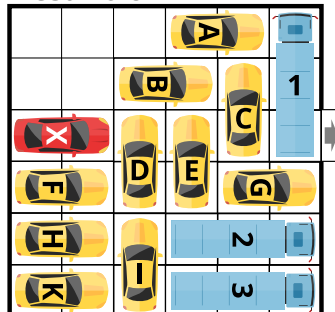
Desafio 6



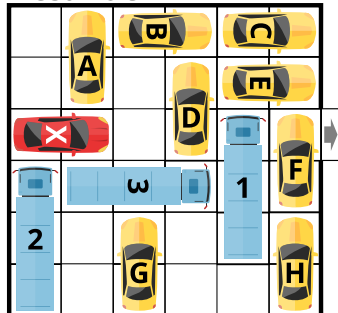
Desafio 7



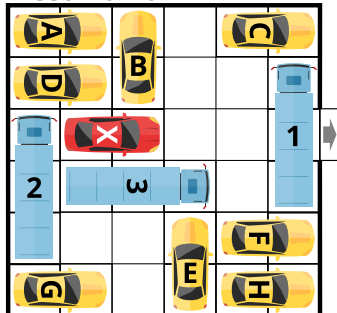
Desafio 8



Desafio 9

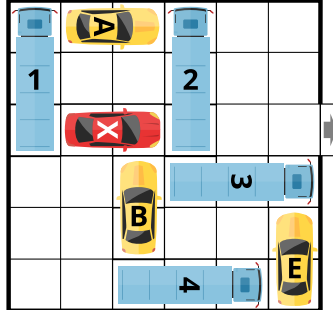


Desafio 10

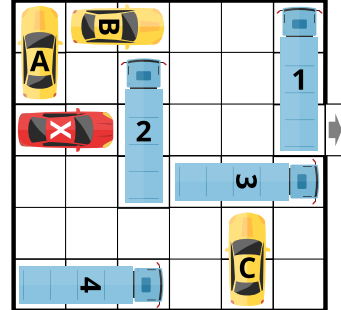


NÍVEL INTERMEDIÁRIO

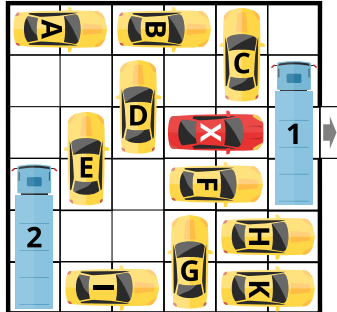
Desafio 11



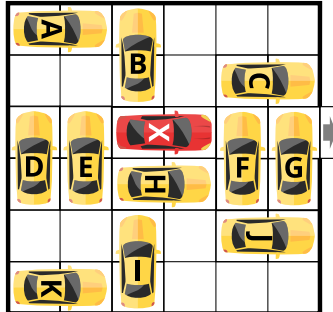
Desafio 12



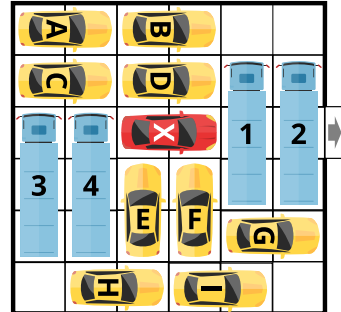
Desafio 13



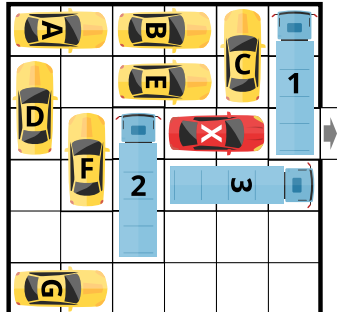
Desafio 14



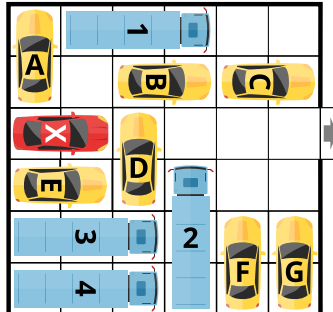
Desafio 15



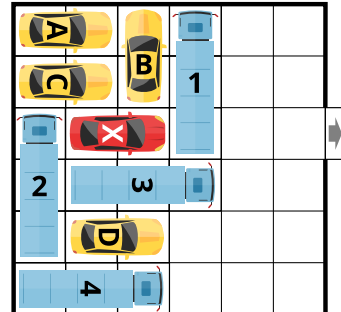
Desafio 16



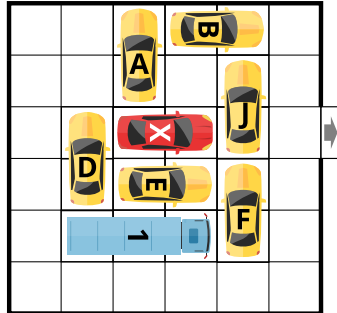
Desafio 17



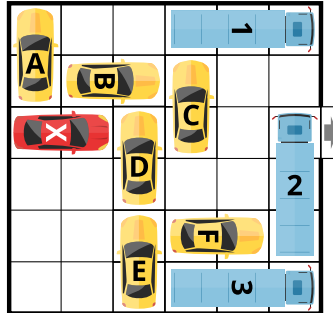
Desafio 18



Desafio 19



Desafio 20



NÍVEL INICIANTE

Como jogar: usando apenas os comandos →, ↓, ← e ↑, mova os carros, ou em linha vertical, ou em linha horizontal com o objetivo de tirar o carro X do estacionamento pela lateral direita (Saída).

NÍVEL INICIANTE

Como jogar: usando apenas os comandos →, ↓, ← e ↑, mova os carros, ou em linha vertical, ou em linha horizontal com o objetivo de tirar o carro X do estacionamento pela lateral direita (Saída).

VEÍCULO	MOVIMENTOS				
C	←	←	←		
1	↓	↓	↓		
A	→				
2	↑				
B	↑				
4	←	←			
3	↓	↓			
X	4x	→			

VEÍCULO	MOVIMENTOS				
C	←	←	←		
1	↓	↓	↓		
A	→				
2	↑				
B	↑				
4	←	←			
3	↓	↓			
X	4x	→			

[illegible][illegible][illegible][illegible][illegible][illegible][illegible][illegible][illegible][illegible][illegible]

**NÍVEL
INTERMEDIÁRIO**

Desafio 11

[illegible]

Desafio 12

[illegible]

Desafio 13

[illegible]

Desafio 14

[illegible][illegible]

Desafio 15

[illegible]

Desafio 16

[illegible]

APÊNDICE D . Termo de Consentimento Livre e Esclarecido



TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

Título: ENSINO DE PROGRAMAÇÃO NO ENSINO MÉDIO POR MEIO DE UMA ABORDAGEM LÚDICA, AUTURAL E COLABORATIVA

OBJETIVO DO ESTUDO: O objetivo deste projeto é identificar os desafios enfrentados por estudantes no aprendizado de programação e como uma nova abordagem pode contribuir no processo de ensino-aprendizagem. O contexto da pesquisa envolve estudantes do curso técnico de informática da Escola Técnica Estadual Maria Mercedes Mendes Teixeira que cursam a disciplina Linguagem de Programação 1.

ALTERNATIVA PARA PARTICIPAÇÃO NO ESTUDO: Você tem o direito de não participar deste estudo. Estamos coletando informações para entender como uma nova abordagem no processo de ensino-aprendizagem pode melhorar o ensino de programação para os estudantes. Se você não quiser participar do estudo, isto não irá interferir na sua vida profissional/estudantil.

PROCEDIMENTO DO ESTUDO: Se você decidir integrar este estudo, você participará de uma entrevista em grupo e/ou de uma entrevista individual que durará aproximadamente 1 hora. Suas respostas à entrevista serão utilizadas por nós como parte do objeto de pesquisa.

GRAVAÇÃO EM VÍDEO: Todas as entrevistas serão gravadas em vídeo. As gravações serão armazenadas em ambiente seguro. Os arquivos digitais com as gravações serão utilizados para coleta e divulgação dos dados. Se você não quiser ser gravado, você não poderá participar deste estudo.

REGISTROS FOTOGRÁFICOS: As entrevistas serão registradas em fotografias. Os arquivos digitais fotográficos serão utilizados para apresentação dos resultados da pesquisa. Se você não quiser ser fotografado, você não poderá participar deste estudo.

RISCOS: Você pode achar que determinadas perguntas incomodam a você, porque as informações que coletamos são sobre as suas experiências em sala de aula. Assim você pode escolher não responder quaisquer perguntas que o façam sentir-se incomodado/constrangido ou se julgar que sua privacidade está sendo invadida. Você pode desistir de participar da entrevista a qualquer momento, caso sinta-se desconfortável, estressado, cansado ou por qualquer outro motivo, inclusive se achar que a pesquisa está tomando muito do seu tempo e causando interferências à sua vida e rotina.

BENEFÍCIOS: Sua entrevista ajudará na pesquisa de novas abordagens para o ensino de programação, mas não será, necessariamente, para seu benefício direto. Entretanto, fazendo parte deste estudo você fornecerá mais informações que podem contribuir para a melhoria do aprendizado de programação por estudantes que ingressarem no curso.



CONFIDENCIALIDADE: Todas as informações obtidas por meio dessa pesquisa serão confidenciais e o sigilo sobre sua participação é assegurado. Nenhum dado será divulgado de maneira que possibilite sua identificação. Após coleta dos dados, todos os documentos e gravações produzidos serão transferidos para o computador do pesquisador e armazenados em ambiente seguro e desconectado da Internet (offline). Como foi dito acima, seu nome não aparecerá na dissertação, bem como em nenhum formulário a ser preenchido por nós.

DIVULGAÇÃO DOS RESULTADOS: Os resultados do estudo serão divulgados pelo pesquisador e por seu orientador na literatura especializada ou em congressos e eventos científicos. Nenhuma informação será utilizada em prejuízo de pessoas e/ou comunidades, inclusive em termos de autoestima, prestígio e/ou qualquer interesse econômico – financeiro.

DÚVIDAS E RECLAMAÇÕES: Esta pesquisa está sendo realizada no Grupo de Pesquisa sobre Tecnologias de Comunicação para Educação e Colaboração (ComunicaTEC). Possui vínculo com a Universidade Federal do Estado do Rio de Janeiro – UNIRIO através do Programa de Pós-Graduação em Informática (PPGI) do Departamento de Informática Aplicada, sendo o aluno de mestrado Ygor Santos Barros o pesquisador principal, sob a orientação do Prof. Dr. Mariano Pimentel. Os investigadores estão disponíveis para responder a qualquer dúvida que você tenha. Caso seja necessário, contate Ygor Barros no e-mail ygor.barros@edu.unirio.br ou telefone (21) 99165-6068, ou o Comitê de Ética em Pesquisa, CEP-UNIRIO no telefone (21) 2542-7796 ou e-mail cep@unirio.br. Você terá uma via deste consentimento para guardar com você. Você fornecerá nome, endereço e telefone de contato apenas para que a equipe do estudo possa lhe contatar em caso de necessidade.

Eu concordo em participar deste estudo.

Assinatura: _____

Data: 27/06/2022

Endereço de contato _____

Telefone de contato _____

Assinatura (Pesquisador):

Nome: YGOR SANTOS BARROS

Data: 27/06/2022

APÊNDICE E . Questionário

Questionário FIOCRUZ -

<https://questionario.fiocruz.br/index.php/admin/printablesurvey/sa/ind...>

Existe(m) 10 questão(ões) neste questionário.

Declaração de Consentimento

Li as informações acima, tenho ciência de que meus **dados pessoais não serão revelados** e que serão utilizados unicamente para esta pesquisa. Sendo assim, dou meu consentimento de livre e espontânea vontade para participar deste estudo.
*

❶ Escolha uma das seguintes respostas:

Favor escolher apenas uma das opções a seguir:

☐ Concordo em participar da pesquisa

Qual o seu gênero?

❶ Escolha uma das seguintes respostas:

❶ Se você escolher 'Outros:' por favor especifique a sua escolha no campo de texto.

Favor escolher apenas uma das opções a seguir:

☐ Feminino

☐ Masculino

☐ Prefiro não informar

☐ Outros

Qual a sua idade (em anos)?

❶ Somente um valor inteiro pode ser informado neste campo.

Por favor, coloque sua resposta aqui:

Qual(is) dispositivo(s) você utilizou para estudar o conteúdo da disciplina?

❗ Escolha a(s) que mais se adequem

Por favor, escolha as opções que se aplicam:

- ☐ Computador
- ☐ Notebook
- ☐ Tablet
- ☐ Smartphone

Qual(is) local(is) você utilizou para estudar o conteúdo da disciplina?

❗ Escolha a(s) que mais se adequem

Por favor, escolha as opções que se aplicam:

- ☐ Casa
- ☐ Trabalho
- ☐ Laboratório da escola
- ☐ Lan House
- ☐ Casa de familiar/amigo

Antes de iniciar o curso técnico, posso afirmar que

❗ Escolha uma das seguintes respostas:

Favor escolher apenas uma das opções a seguir:

- ☐ nunca tinha ouvido falar de Programação
- ☐ já havia escutado sobre Programação
- ☐ já conhecia Programação de vídeos que assisti no Youtube
- ☐ já realizei curso na área de Programação
- ☐ já trabalho com Programação

Na disciplina foram utilizados diversos recursos e realizadas diversas atividades para oportunizar a aprendizagem. Avalie o grau de **importância** que cada atividade/recurso para o seu aprendizado de programação, sendo 0 (zero) se a atividade/recurso foi irrelevante para o seu processo de aprendizagem, 4 (quatro) se a atividade/recurso foi muito relevante, ou use os valores intermediários entre estes extremos para expressar sua avaliação. Assinale "Não se aplica" caso você não tenha participado da atividade.

Por favor, escolha a resposta adequada para cada item:

	X (Não se aplica)	0 (Irrelevante)	1	2	3	4 (Muito importante)
Hora do Código	☐	☐	☐	☐	☐	☐
Lightbot	☐	☐	☐	☐	☐	☐
Scratch	☐	☐	☐	☐	☐	☐
Dinâmica "Estacionamento algorítmico"	☐	☐	☐	☐	☐	☐
Recursos audio visuais	☐	☐	☐	☐	☐	☐
Pseudo código (Calango)	☐	☐	☐	☐	☐	☐
Projeto de jogo em Scratch	☐	☐	☐	☐	☐	☐
Projeto de jogo em Calango	☐	☐	☐	☐	☐	☐
Projeto de jogo em Java	☐	☐	☐	☐	☐	☐
Listas de Exercícios	☐	☐	☐	☐	☐	☐
Tradução de pseudo código para Java	☐	☐	☐	☐	☐	☐

	X (Não se aplica)	0 (Irrelevante)	1	2	3	4 (Muito importante)
Atividades em pares	☐	☐	☐	☐	☐	☐
Material disponibilizado (Apresentações em PDF)	☐	☐	☐	☐	☐	☐
Curso de Java na plataforma Oracle Academy	☐	☐	☐	☐	☐	☐

Explique por que algumas atividades/recursos tiveram mais importância para o seu aprendizado de programação.

Por favor, coloque sua resposta aqui:

Explique por que algumas atividades/recursos tiveram menos importância para o seu aprendizado de programação.

Por favor, coloque sua resposta aqui:

Que alterações na disciplina você sugere para o próximo semestre? Pode ser a inclusão ou exclusão de algum tema, atividade, recurso etc.

Por favor, coloque sua resposta aqui:

Enviar questionário

Obrigado por ter preenchido o questionário.

APÊNDICE F . Termo de Autorização - Imagem e Som



Termo de Autorização para uso de Imagem e Voz

Nome completo: _____
CPF: _____ E-mail: _____

Nome do evento:

Apresentação dos projetos de jogos da disciplina Linguagem de Programação 1 do curso técnico de Informática da Escola Técnica Estadual Maria Mercedes Mendes Teixeira.

Data de realização: 27/06/2022

Termo de autorização:

Autorizo, para todos os fins em direito admitidos, a utilização da minha imagem e voz constantes em fotos, gravações e filmagens decorrentes da minha participação no evento acima especificado, sob a responsabilidade da Universidade Federal do Estado do Rio de Janeiro (UNIRIO), sendo que a referência ao meu nome, que constitui um direito moral, deverá ser respeitada sempre.

As imagens e voz poderão ser exibidas na dissertação de Mestrado do aluno Ygor Santos Barros com orientação do Professor Dr. Mariano Pimentel, em apresentações audiovisuais dela, em publicações e divulgações disponibilizadas em acesso aberto, e do Repositório Institucional da UNIRIO, bem como de outros sistemas de disseminação da informação e do conhecimento.

A autorização neste termo especificada é gratuita e por prazo indeterminado. Por ser esta a expressão de minha vontade, nada terei a reclamar a título de direitos conexos à minha imagem e voz.

Rio de Janeiro, 27 de junho de 2022.

Assinatura

APÊNDICE G . Apresentação Grupo Focal



GOVERNO DO ESTADO DO RIO DE JANEIRO
SECRETARIA DE ESTADO DE CIÊNCIA, TECNOLOGIA E INOVAÇÃO
FUNDAÇÃO DE APOIO À ESCOLA TÉCNICA
ESCOLA TÉCNICA ESTADUAL MARIA MERCEDES MENDES TEIXEIRA (ETEMMMT)

GRUPO FOCAL



FAETEC Secretaria de
Ciência, Tecnologia
e Inovação



GOVERNO DO ESTADO
RIO DE JANEIRO

GRUPO FOCAL LP1 2022.1



Quais são os pontos fortes e fracos da disciplina Linguagem de Programação 1?

GRUPO FOCAL LP1 2022.1

O que vocês acharam da Programação com blocos (Hora do Código, Lightbot e Scratch)? Ela deve ser mantida ou retirada na próxima turma da disciplina de programação do nosso curso?

GRUPO FOCAL

Quais são os pontos fortes e fracos da disciplina Linguagem de Programação 1?

GRUPO FOCAL LP1 2022.1

O que vocês acharam da Programação com blocos (Hora do Código, Lightbot e Scratch)? Ela deve ser mantida ou retirada na próxima turma da disciplina de programação do nosso curso?

Quais são os pontos fortes e fracos da disciplina Linguagem de Programação 1?

GRUPO FOCAL

O que vocês acharam da dinâmica “Estacionamento Algorítmico”? Ela deve ser mantida ou retirada na próxima turma da disciplina de programação do nosso curso?

GRUPO FOCAL LP1 2022.1



GRUPO FOCAL LP1 2022.1



GRUPO FOCAL LP1 2022.1



GRUPO FOCAL LP1 2022.1



APÊNDICE H . Transcrição do Grupo Focal

Quais são os pontos fortes e fracos da disciplina Linguagem de Programação 1?

Aluno 1: “Eu acho que o ponto forte dessa disciplina é que como a gente está numa turma pequena a gente consegue dividir informação muito fácil um com o outro e pela metodologia que o senhor aplica é fácil entender a matéria já que é dividida tudo certinho em *slide* ou se a gente tem uma dúvida em um ponto específico, a gente vai naquele ponto e já está lá tudo bonitinho para a gente, tem exemplos, tem tudo. Ponto fraco ...”

Aluno 2: “Bom eu acho que um ponto fraco seria assim o pouco tempo porque é uma matéria, uma linguagem que eu acho que a gente teria que ter mais tempo para poder aprender porque são detalhes, são muitas informações que se você não praticar no dia a dia você acaba se perdendo e esquecendo a informação.”

Aluno 3: “Se tiver tempo para fazer mais exercício em sala você tem um comprometimento maior.”

Aluno 1: “Ou mais projetos também porque eu senti que esse projeto agora do show do milhão por ele, não sei se para todo mundo mas para mim foi, por ele ter um prazo tão grande eu acabei deixando para fazer nas duas últimas semanas ao invés de começar quando ele deu, então se tivesse um prazo mais curto com uma coisa relativamente mais simples, também não que tenha sido difícil mas foi um negócio complexo de elaborar, a gente conseguiria elaborar mais do que a gente aprende mais em curto prazo.”

Aluno 4: “... Então um projeto inicial e a gente ir aprimorando ele até chegar ao final com ele todo otimizado também seria bom.”

Aluno 5: “Outro ponto que eu achei bom foi o fato de ser projeto e não ser prova porque prova a gente vai tipo só tentar gravar ali ou memorizar, esse projeto do show do milhão é claro que a gente aprendeu bem mais do que ficar só estudando ali, eu aprendi bem mais tudo, o *if*, o *else*, tudo e a gente praticou mais também.”

Aluno 1: “E o que eu achei legal foi que são vários caminhos para você fazer a mesma coisa tanto que a gente vê os projetos de todo mundo, por mais que todo mundo tenha acertado em certo ponto, é ele tem uma certa diferença de um para o outro, a estrutura do código, então eu acho que quando a gente está tentando aprender a fazer a gente acaba pesquisando na internet e pesquisando em lugares diferentes, coisas diferentes que vai acabar ajudando o nosso código e quando a gente trás para a sala não fica uma coisa tão linear de todo mundo ter feito a mesma coisa, todo mundo fez um pouco de cada e aí acaba que todo mundo aprendeu um pouco de coisas diferentes.”

Aluno 6: “Vale contar que a pseudo língua, Calango, ajudou porque o primeiro contato que a gente teve com a linguagem é estranho né se eu desse de cara com um *switch* e chaves, o Calango...”

Momento em que a discussão se eleva e os estudantes falam simultaneamente: “é a diferença do calango para o java... java dava tempo de aprender no Youtube e o Calango deu para entender legal né...”

Aluno 6: “... o calango nos trouxe a lógica e claro que a didática do Calango nos preparou para poder enfrentar o Java...”

Aluno 7: “Isso é um ponto forte...”

Aluno 6: “... porque quando a gente lidou com Java na outra disciplina foi mais tranquilo...”

Momento em que os estudantes falaram simultaneamente e não é possível transcrever as falas.

Aluno 2: “Uma coisa que eu percebi também que foi assim, a gente fez os códigos, colocou para rodar, por exemplo a gente sabe que sabe fazer porém na hora de explicar fica mais complicado porque assim, a gente aprendeu um pouco daquele projeto ali que nós elaboramos porém na hora de explicar para os demais ainda está meio meio que enrolado né, acho que precisa um pouco mais de prática.”

Aluno 8: “Essa questão também professor, eu tive muita dificuldade de baixar o Calango e executar, pedi ajuda e muitos me ajudaram mas mesmo assim eu não consegui, então eu fiquei assim sem poder praticar por causa dessa dificuldade que eu tive enquanto o equipamento né... se tivesse essa proposta aqui em sala, determinado um tempo de fazer mas eu já estava muito perdida para poder fazer... se tivesse um tempo separado no laboratório que a gente pudesse vim na unidade e elaborar já ajudaria, como foi no meu caso...”

Aluno 2: “...ou até mesmo eu tenho computador e tive que usar um computador emprestado né mas no caso delas que tiveram dificuldades, de trazer o computador para ver uma maneira de fazer a instalação do aplicativo porque a gente ainda não tem total conhecimento no equipamento, na máquina que é o computador... então acho que seria uma opção para ajudar também quem está com dificuldade para poder instalar esses programas ai no computador... Eu, o Lucas teve de entrar no meu pelo Teams para poder instalar o NetBeans porque eu também não estava conseguindo.”

O que vocês acharam da Programação com blocos (Hora do Código, Lightbot e Scratch)? Ela deve ser mantida ou retirada na próxima turma da disciplina de programação do nosso curso?

Aluno 5: “Mesmo que eu tenha chegado depois, o fato deles terem apresentado

o trabalho me interessou mais a querer aprender, a querer ver, pensar e tentar... então mesmo que a gente, os demais que chegaram depois não apresentaram o trabalho, no meu ver eu consegui fazer em casa, o joguinho de boa e me ajudou na construção, no fato da construção da programação eu comecei ali... eu tive que botar o movimento, eu tive que fazer na ordem certinha porque senão não ia rodar.”

Aluno 3: “Só de ter uma interface gráfica já dá uma ajuda.”

Aluno 7: “Eu acho que esse a programação de blocos ajudou a gente a entender os algoritmos, que para você executar você precisa dos códigos por etapas.”

Aluno 5: “E o próprio site do Scratch já deixa ali delimitado, esse aqui você fala, esse aqui você movimento, assim, assim sucessivamente... só vai arrastando e é bem didático.”

Aluno 4: “E o tempo que foi usado também foi bom porque não pegou muito tempo da matéria e deu para aprender legal.”

Pesquisador intervém para estimular o diálogo entre os estudantes, questionando se as tecnologias contribuíram para o aprendizado deles. Os estudantes acenam positivamente com a cabeça e o diálogo retorna.

Aluno 4: “Sim porque a gente está montando o bloco e vendo o que está acontecendo e vê o resultado.”

Momento em que a discussão se eleva e os estudantes falam simultaneamente: “é igual a construir uma casa e ter a planta da casa... exatamente, ele ajuda na programação e no algoritmo...”

Aluno 5: “Ele que me deu interesse, porque assim que eu cheguei no curso já tinha o projeto dos caras, de montar um jogo e eu já estava caraca o bagulho *inaudível, seguido por muitas gargalhadas* os moleques estão montando um jogo me ajudou a querer e aprender a fazer.”

O que vocês acharam da dinâmica “Estacionamento Algorítmico”? Ela deve ser mantida ou retirada na próxima turma da disciplina de programação do nosso curso?

Aluno 3: “Eu acho que pode ser usado mais no começo.”

Aluno 4: “Durante o Scratch, que dá para fazer esse conjunto...”

Aluno 3: “... que aí já cria aquela sequência de você conseguir pensar em sequência né.”

Aluno 4: “... te ajuda na lógica ...”

Aluno 3: “... na lógica né, de seguir as linhas, de escrever as linhas dos códigos em sequência ...”

Aluno 6: “... em contar que a gente teve oportunidade de fazer isso em equipe, em equipe ...”

Pesquisador intervém para estimular o diálogo entre os estudantes, questionando se a dinâmica deve ser mantida para o próximo semestre. Os estudantes acenam positivamente com a cabeça e o diálogo retorna com as falas acontecendo simultaneamente. É possível ouvir os estudantes responderem com “sim, total”. Sem muito diálogo, o pesquisador tenta promover o diálogo provocando os estudantes com a pergunta extra “O que vocês achariam de levar a dinâmica do Estacionamento Algorítmico para dentro do Scratch?” Os estudantes acenam positivamente com a cabeça e o diálogo retorna com as falas.

Aluno 2: “eu ia falar de usar ele para criar um programa e utilizar ele dentro do programa seria até um pouco mais desafiador.”

Aluno 3: “acho interessante mas é um pulo ousado aí você sair de mexer no tabuleiro para criar no Scratch.”

Aluno 5: “olha aí o projeto para a próxima turma *gargalhadas*.”

Aluno 3: “agora da zero para todo mundo e coloca a gente para fazer como projeto final *gargalhadas*.”

Pesquisador questiona se a dinâmica proporcionou aprendizado de lógica e algoritmo para os estudantes e eles acenam positivamente com a cabeça, concordando com o pesquisador.

O que vocês acharam da utilização de pseudo código (Calango)? Ela deve ser mantida ou retirada na próxima turma da disciplina de programação do nosso curso?

Momento em que a discussão se eleva e os estudantes falam simultaneamente indicando positivamente o uso do Calango.

Aluno 1: “eu acho que tive a mesma ideia do Nathan que para fazer o Java já foi melhor que no Calango, que como o Calango é um pseudo código tem alguma função que a gente vai precisar e as vezes não tem, por exemplo esse negócio da tabela ASCII eu não entendi direito como funciona no Calango.”

Aluno 3: “mas vale ressaltar que vocês já chegaram com a explicação no meio do caminho... a gente que chegou *inaudível... então vocês que chegaram depois de nós, da massa maior, vocês pegaram menos explicação no Calango...”

Aluno 6: “Eu tinha medo de código e o Calango diminuiu isso.”

Aluno 1: “É por isso que a gente se identificou mais com Java porque era uma coisa mais fluida.”

Aluno 9: “e ainda tem outra coisa, o Calango ajudou a gente a entender de forma

mais fácil as funções, a maneira de declarar... quando a gente foi para o Java era só a gente lembrar, a estrutura *if* lembra a estrutura do se.”

Aluno 1: “mas por exemplo, você acha que para fazer alguma coisa você precisaria de um auxílio do Calango ou você conseguiria fazer em Java?”

Aluno 9: “Assim, de cara?”

Aluno 1: “De cara, pra fazer.”

Aluno 9: “Não.”

Aluno 1: “Tem que ter o Calango pra isso então.”

Aluno 9: “Não tipo, não conseguiria porque ao ver aquela lista de códigos tudo em inglês, porra eu ia me assustar mas como o Calango estava em Português *inaudível*.”

Aluno 2: “é uma estrutura de base para você aprender o Java.”

Aluno 3: “e tipo assim, eu sei falar inglês e isso não é uma dificuldade na tradução ...”

Momento em que os estudantes falaram simultaneamente e não é possível transcrever as falas.

Aluno 3: “você entender exatamente o que cada linha que o... eu parei quando eu li *package, main, string* não sei o que lá, eu falei não dá para mim e tal...”

Aluno 6: “mano, escreva ou *System.out.println*, que isso cara...”

Aluno 1: “eu acho que um dos problemas que o Java e o Calango foi que nessa tradução alguns detalhes não existem e que no Java existem, por exemplo abrir chave que é um negócio que você tem que fazer por costume, abrir e fechar e ver onde está... no Calango não tem isso e tipo ajuda só que quando você vai passar para o Java você se perde porque você não tinha muito daquilo... então pô quantas vezes o meu código sangrou porque eu esqueci de colocar uma chave então...”

Pesquisador intervém questionando se a disciplina estivesse sido iniciada diretamente com Java no lugar do Calango, os estudantes teriam mais ou menos dificuldade.

Aluno 1: “Não, ia ser mais difícil. Mais dificuldade. Facilitou.”

Aluno 7: “só ia estar aqui eu e o Diego...”

Aluno 6: “teria um impacto bizarro... o Calango nos trouxe a visão de lógica e estrutura, o Java a gente conseguiu montar *inaudível* ... compreender uma nova linguagem, serviu para ver problemas com estrutura, coisa que o Calango matou... o Calango nos deu segurança para poder ir no Java...”

Momento em que os estudantes falaram simultaneamente e não é possível transcrever as falas.

Aluno 2: “o que aconteceu é que vocês entraram quase no final então por isso que você está sentindo a dificuldade no Calango... eu que peguei um pouco mais a frente já vi o Calango como uma estrutura base para eu ter uma noção quando chegou no Java... é que ele pegou pouco o Calango entendeu.”

Aluno 6: “o Calango é nossa referência relacionado a estrutura.”

Aluno 5: “no meu ponto de ver foi mais porque eu cheguei na metade para o fim mas na metade que eu cheguei já aprendi sobre variável, sobre o escreva, o escreval e consegui transformar em Java... só que quando eu transformei em Java eu deixei um pouco o Calango de lado... quando eu e o Diego montou o projeto, eu sabia que ele tinha mais conhecimento em Calango então eu preferi deixar o Calango com ele. Como a gente não mora perto, ele não tinha como fazer dupla comigo, aí foi dividindo o código, ele foi para o Calango e eu fui para o Java. Aí nosso código funcionou, ele começou no Calango e eu transformei em Java. Eu fiz a ajuda e ele jogou no Calango... Ajudou, realmente ajudou porque uma pessoa chegar e ver tipo *if else*, *while*, não seria nem a dificuldade mas a desistência das pessoas...”

Aluno 7: “ia ser muito complicado...”

Aluno 3: “é mais amigável para iniciar o ...”

Aluno 1: “é mais amigável.”

Aluno 7: “porque depois que entrou no Java a gente já estava mais familiarizado...”

Aluno 5: “se eu tivesse pegado o Calango desde o começo também aprenderia melhor igual eu aprendi o Java mas como eu cheguei na metade eu aprendi mais o Java... mais pode manter o Calango.”

Aluno 7: “porra, se tivesse começado com o Java eu não estaria aqui não...”

O que vocês acharam da utilização de Java? Ela deve ser mantida ou retirada na próxima turma da disciplina de programação do nosso curso?

Aluno 1: “pode manter, pode manter. Java é muito bom. Java é bom.”

Aluno 3: “Java é muito bom.”

Aluno 1: “Professor, você é que vai dar essa matéria do segundo módulo? (JAVA) ... Vamos fazer um abaixo assinado para ser você...”

Aluno 6: “Tem que manter no primeiro para a pessoa chegar no segundo já sabendo o que ela vai encontrar porque se a pessoa se acostumar com o Calango e depois der de cara com o Java, vai ver aquela outra realidade... então não, tem que ter aquilo em mente, o Calango é inicial para eu aprender os comandos, o que é função, vetores, variáveis... então, pegou isso, leva para o Java... faz o show do milhão no Calango, sofremos... faz no Java também e sofre o dobro... mas foi bom porque agora eu não tenho mais medo

de abrir o Java porque abre o Calango "uh que bom", abre o Java "que isso!"... o Diego chegou aqui e falou vamos mexer no Java, aqui o suor... já viram o meme do *inaudível* suando, eu estava igual a ele cara..."

Momento em que os estudantes falaram simultaneamente e todos eles concordam com a manutenção do Java na disciplina.

Aluno 5: "Mas o que realmente ajudou mesmo foi quando o senhor chegou na introdução do Java que o senhor não mostrou só o começo do Java, comparou no PDF o Calango com o Java, mostrando a diferença."

Pesquisador intervém questionando se a tradução dos códigos do Calango para o Java ajudaram no aprendizado dos estudantes. Todos os alunos acenaram positivamente com a cabeça concordando com a afirmação.

Aluno 5: "Mesmo para quem não tinha entendido o Calango, mesmo tendo chegado depois deu para entender..."

Aluno 1: "é exatamente isso aí..."

Aluno 5: "... o senhor começou do zero, variável e variável aqui Calango e no Java, escreva e *System.out.println*, foi exatamente assim..."

Aluno 1: "... e esse sistema é bom porque ele é direcional porque quem aprendeu mais Java e quer aprender Calango consegue por esse sistema..."

Aluno 5: "... dá para transformar um no outro... professor orgulhoso meu trabalho foi bem feito *gargalhadas*"

O que vocês acharam na aprendizagem baseada em projetos (jogos digitais)? Ela deve ser mantida ou retirada na próxima turma da disciplina de programação do nosso curso?

Aluno 1: "eu acho que só deve ser feita dessa maneira por todos os professores... vamos abolir a prova no papel..."

Aluno 3: "... é isso aí! ..."

Aluno 1: "... porque é um método antigo isso aí que inventou na França..."

Aluno 8: "É mais dinâmico né professor."

Aluno 2: "É porque você... é a prática, tudo é a prática e ali você está praticando, você está fazendo. Não é no papel onde eu vou ter que decorar não, ali você está praticando."

Aluno 1: "E o melhor disso tudo que eu acho é que não avalia a gente pelo conhecimento que a gente tem naquele momento, avalia a gente pelo processo de ter chegado lá, porque por exemplo ninguém aqui eu acho que fez o projeto totalmente em dupla, todo mundo "ô como fez isso aqui, pô não to conseguindo" e numa prova isso é

cola e em um projeto isso é ajuda mútua que acabou levando todo mundo a ter caminhos diferentes, trabalho em equipe...”

Aluno 3: “Até porque assim é o mercado de trabalho também né, não adianta, não desenvolve nada sozinho.”

Aluno 6: “Eu não entendia muito bem inicialmente sobre função e sobre tabela ASCII, eu tava vendo sobre isso em aula e não entendia mas não está lá no meu projeto? A gente tem que fazer e bater cabeça e compreende o que que é função... cheguei Diego e ele começou a apontar, o cara apontou e eu fui pegando a ideia e saquei... o que é tabela ASCII, quando eu abri o site e fui ver o que era, os números e as letras, funcionou...”

Aluno 2: “É eu tive que pesquisar para saber o modo aleatório lá que é a biblioteca *random*...”

Aluno 6: “Agora na prova, eu ia ler, passei e depois acabou...”

Aluno 1: “Eu não sei se todo mundo está se sentindo dessa forma mas pelo que eu vi eu não consegui fazer o pulo e a eliminação de respostas mas eu vi que pessoas conseguiram e eu fiquei curioso para ver como é o código e entender como eles fizeram isso porque como a gente tentou fazer e se dedicou para aquilo, quando a gente não consegue a gente se sente frustrado de certa forma e quando vê que uma pessoa que está no mesmo nível que a gente conseguiu, a gente consegue ler aquilo ali e falar “pô essa pessoa aqui está entendendo as coisas no mesmo nível que eu consegui fazer isso então eu vou tirar dúvida com ela porque ela vai saber me passar porque ela entendeu de uma forma”. Eu estava conversando com o Lucas e ele veio me passar o código e me explicar como faz.”

Aluno 5: “Na aula do Thiago toda, o Daniel o tempo todo "mano, me explica isso aqui? Mas se eu usar isso aqui, vai dar certo?" Eu falei "vai cara, bota tudo dentro do if que vai dar bom!" Ai ele "vou tentar em casa." Aí de madrugada ele manda "deu certo aqui o projeto, obrigado pela ajuda" Eu mostrei o código do Java e ele estava escrevendo o código no e-mail... quando foi hoje aí ele, "deu tudo certo"... cara foi tudo em conjunto, ninguém fez sozinho, realmente ninguém ficou sozinho... e outra parte do projeto também que é bom é que a gente sempre vai na prática, meu projeto deu 670 linhas, imagina fazer isso na prova, não ia funcionar legal. E como a gente ia saber que funcionou?”

Aluno 1: “Não tem como, o aplicativo te mostra onde você está errando e você procura.”

Aluno 5: “Ia passar para o professor na reza filho, testa lá...”

Aluno 3: “Ou seja, ia ter que aumentar o número de projetos em projetos menores.”

Aluno 2: “E o pior que a gente roda, roda, roda e parece que está funcionando perfeito e quando vai rodar aí parece que tinha alguma coisa *gargalhadas*”

Aluno 4: “o pessoal da informática por ser projeto de jogo já estimula.”

Aluno 1: “Outro ponto disso professor, é que quando a gente vai fazer o projeto e a gente tem prazo e a gente pode falar com as outras pessoas, você se sente confortável em errar. Numa prova a gente fica muito nervoso para acertar o máximo de coisas possíveis e a gente não se dá o trabalho de ver o que errou e tentar consertar, a gente quer acabar com aquele momento de sofrimento o mais rápido possível. O projeto a gente queria ter até mais prazo para implementar mais coisas.”

Aluno 5: “Isso foi o que todo mundo achou.”

O que vocês acharam da avaliação da disciplina? O que vocês sugerem de modificação para a próxima turma da disciplina de programação do nosso curso?

Aluno 3: “Cara sinceramente, eu acho que não precisa mudar nada. Foi uma apresentação descontraída, ninguém ficou nervoso, todo mundo brincou com o outro, teve acidente no meio do percurso *gargalhadas*”

Aluno 1: “Nada. Eu também acho que não. Está bom do jeito que está.”

Pesquisador intervém questionando se a avaliação colaborativa aplicada para o projeto foi interessante e se deveria ser mantida para as próximas turmas.

Aluno 1: “Foi interessante dessa forma.”

Aluno 5: “Para mim foi algo novo, eu nunca tinha visto esse negócio.”

Aluno 2: “A gente viu uma forma de, ao mesmo tempo de estar avaliando, se auto avaliar né... Vejo uma coisa interessante vou usar na minha avaliação também, vou usar na minha apresentação então assim é uma forma que foi feita para avaliar e se auto avaliar... tipo eu não fiz tão bem quanto ele mas não é uma crítica, é construtivo né, então foi bem interessante.”

Aluno 1: “Eu acho que também funciona muito bem para a gente meio que ficar com o pé no chão porque quando a gente faz e funciona, está tudo bem, isso que importa. Mas quando a gente compara com o das outras pessoas a gente fala "pô essa pessoa fez uma coisa que eu nem sequer tentei fazer". Talvez o meu esforço tenha sido menor, de certa forma, que o daquela pessoa. Então se tem que avaliar por nota, e tem um topo que é 10, então na minha visão, eu não poderia ter uma nota igual a daquela pessoa que eu sei que se esforçou mais do que eu. Então acho que funciona muito bem para entender onde a gente está faltando e saber que, mesmo que funcionou, não é porque funcionou que está 100% bom.”

Aluno 8: “Você cria um senso de critério né.”

O pesquisador realizou uma pergunta adicional aos estudantes para entender um pouco mais sobre a experiência de trabalhar em duplas durante a elaboração do projeto. Todos os estudantes responderam positivamente à pergunta e sugeriram que poderiam ter

mais atividades em que a programação pudesse ser compartilhada com um colega de turma. No entanto, como ponto negativo da experiência, os estudantes apontaram a dificuldade de se encontrarem pessoalmente devido à distância de suas residências. Isso dificultou em alguns momentos a elaboração do projeto, uma vez que a escola não disponibiliza acesso aos laboratórios fora do horário de aula.