



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE RUSSAS
CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

ROBÉRIO GIRÃO PINHO

ESTIMULANDO O PENSAMENTO COMPUTACIONAL NO ENSINO MÉDIO
UTILIZANDO O SOFTWARE SCRATCH.

RUSSAS

2024

ROBÉRIO GIRÃO PINHO

ESTIMULANDO O PENSAMENTO COMPUTACIONAL NO ENSINO MÉDIO
UTILIZANDO O SOFTWARE SCRATCH.

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação do Campus de Russas da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Ciência da Computação.

Orientador: Prof. Dr. Markos Oliveira Freitas.

RUSSAS

2024

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas

Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

P724e Pinho, Robério Girão.

Estimulando o Pensamento Computacional no ensino médio utilizando o software Scratch. / Robério Girão Pinho. – 2024.

173 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, 3, Russas, 2024.
Orientação: Prof. Dr. Markos de Freitas Oliveira.

1. Scratch. 2. Pensamento Computacional. 3. ensino médio. 4. programação. 5. raciocínio lógico. I. Título.
CDD

ROBÉRIO GIRÃO PINHO

ESTIMULANDO O PENSAMENTO COMPUTACIONAL NO ENSINO MÉDIO
UTILIZANDO O SOFTWARE SCRATCH.

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Ciência da Computação
do Campus de Russas da Universidade Federal
do Ceará, como requisito parcial à obtenção do
grau de bacharel em Ciência da Computação.

Aprovada em: 17 de Setembro de 2024

BANCA EXAMINADORA

Prof. Dr. Markos Oliveira Freitas (Orientador)
Universidade Federal do Ceará (UFC)

Profa. Dra. Jacilane de Holanda Rabelo
Universidade Federal do Ceará (UFC)

Prof. Dr. Joaquim Bento Cavalcante Neto
Universidade Federal do Ceará (UFC)

Dedico este trabalho à minha família e amigos pelo apoio constante, e aos meus professores pelo incentivo ao longo da minha trajetória acadêmica. Agradeço a todos que acreditaram em mim e estiveram ao meu lado nesta jornada.

AGRADECIMENTOS

Agradeço imensamente aos meus pais, Eliete e Adalto, por todo o amor, apoio e incentivo que me deram ao longo de toda a minha vida. Sem vocês, esta conquista não seria possível. Agradeço também aos meus amigos mais queridos, Isabella, Luan, Wenyo, Lázaro e Douglas por tantos anos ao meu lado, me dando forças e nunca me deixando desistir.

Aos meus colegas de faculdade, quero agradecer pelas valiosas trocas de conhecimento e pelas amizades duradouras, especialmente a Gabriel Rocha, Moisés, Matheus, Pedro Gabriel, Israel, Leonardo, Juan e todos os demais da turma de 2019.1 do curso de Ciência da Computação com quem pude compartilhar momentos inesquecíveis.

A todos os meus professores, que generosamente compartilharam seu vasto conhecimento e contribuíram para meu crescimento acadêmico e profissional, expresse minha sincera gratidão. Suas orientações e dedicação foram pilares fundamentais do meu aprendizado. Gostaria de fazer um agradecimento especial ao meu orientador, Markos Oliveira, cuja orientação, paciência e apoio constantes desempenharam um papel vital no desenvolvimento deste trabalho. Agradeço também à professora Jacilane Rabelo por todo o apoio e por sempre me incentivar a buscar novos desafios. Agradeço ao professor Hugo pela oportunidade de executar este trabalho em sua turma e realizar as atividades da pesquisa.

Por fim, agradeço a todas as pessoas que, direta ou indiretamente, me encorajaram, apoiaram e estiveram ao meu lado durante esta jornada. Suas palavras incentivadoras e críticas construtivas foram essenciais para superar os obstáculos. Para todos vocês, minha mais profunda gratidão. Este trabalho é o resultado de um esforço coletivo e do apoio de pessoas que acreditaram em mim. Sem a presença de vocês, nada disso teria sido alcançado. Muito obrigado!

The sun will rise and we will try again

Stay alive, stay alive for me

— Twenty One Pilots, Truce (2013)

RESUMO

Este estudo investiga a utilização do Scratch como ferramenta para o ensino de programação e desenvolvimento do Pensamento Computacional em alunos do primeiro ano do ensino médio de uma escola profissionalizante. A pesquisa buscou identificar o nível de conhecimento prévio dos alunos em computação, avaliar sua satisfação com o uso do Scratch e analisar o impacto da ferramenta no desenvolvimento do raciocínio lógico e da criatividade. Através de um curso estruturado com desafios progressivos e com a criação de um projeto próprio, os alunos foram introduzidos aos conceitos básicos de programação de forma lúdica e visual. Para coletar dados, foram aplicados questionários que investigaram o perfil dos alunos, suas experiências prévias com computação e sua percepção sobre experiência em computação promovida com as aulas. Os resultados demonstraram que o Scratch se mostrou uma ferramenta eficaz para a aprendizagem de programação, mesmo para alunos com pouco conhecimento prévio em computação. A maioria dos alunos relatou alta satisfação com o uso da ferramenta e percebeu um desenvolvimento significativo em seu raciocínio lógico e na capacidade de resolver problemas de forma criativa, habilidades que são fundamentais no Pensamento Computacional. Além disso, os desafios propostos contribuíram para o engajamento dos alunos e para a construção de projetos cada vez mais complexos. Embora o Scratch tenha se mostrado uma ferramenta efetiva para o ensino de programação, alguns conceitos, como estruturas de repetição, demandaram mais tempo para serem completamente compreendidos pelos alunos, o que exigiu ajustes no planejamento inicial do curso. A metodologia utilizada neste estudo pode servir como base para a implementação de projetos similares em outras escolas, contribuindo para a democratização do acesso à educação em programação e para o desenvolvimento de habilidades essenciais para o século XXI.

Palavras-chave: Scratch; Pensamento Computacional; ensino médio; programação; raciocínio lógico; criatividade

ABSTRACT

This study investigates the use of Scratch as a tool for teaching programming and developing Computational Thinking among first-year high school students in a vocational school. The research aimed to identify the students' prior knowledge of computing, evaluate their satisfaction with using Scratch, and analyze the impact of the tool on the development of logical reasoning and creativity. Through a structured course with progressive challenges and the creation of a personal project, students were introduced to basic programming concepts in a playful and visual manner. Data were collected through questionnaires that explored students' profiles, their prior experiences with computing, and their perceptions of the programming experience provided by the classes. The results showed that Scratch proved to be an effective tool for learning programming, even for students with limited prior knowledge in computing. Most students reported high satisfaction with the tool and observed significant development in their logical reasoning and problem-solving skills, which are fundamental to Computational Thinking. Additionally, the proposed challenges contributed to student engagement and the creation of increasingly complex projects. Although Scratch was shown to be an effective tool for teaching programming, some concepts, such as control structures, required more time to be fully understood by students, necessitating adjustments in the initial course planning. The methodology used in this study can serve as a basis for implementing similar projects in other schools, contributing to the democratization of access to programming education and the development of essential 21st-century skills.

Keywords: Scratch; Computational Thinking; high school; programming; logical reasoning; creativity

LISTA DE FIGURAS

Figura 1 – Pilares do Pensamento Computacional	21
Figura 2 – Página de criação de projetos no Scratch	25
Figura 3 – Ilustração da atividade desenvolvida na etapa 1 com Scratch.	35
Figura 4 – Ilustração da atividade desenvolvida nas etapas 2 e 3 com Scratch.	36
Figura 5 – Comparação dos resultados de diferentes abordagens de ensino entre dois semestres.	37
Figura 6 – Exemplo de implementação de uma fórmula da física no <i>software</i> VisuAlg. .	38
Figura 7 – Exemplo de abordagem matemática no <i>software</i> Scratch.	39
Figura 8 – Exemplo de aplicação de gamificação entre equipes utilizando o <i>software</i> Robocode.	39
Figura 9 – Resultados da pesquisa de satisfação dos alunos em relação à oficina.	42
Figura 10 – Resultados da pesquisa de facilidade de uso da ferramenta Scratch.	43
Figura 11 – Expressões aritméticas e lógicas no Scratch.	64
Figura 12 – Combinação de blocos de expressões no Scratch.	64
Figura 13 – Padrão arredondado para combinação de blocos do Scratch.	66
Figura 14 – Bloco de repetição "repita ... vezes" utilizado nos desafios.	67
Figura 15 – Área de palco e de ações utilizados nos desafios.	67
Figura 16 – Bloco de repetição "sempre" aninhado com outros blocos de repetição. . . .	68
Figura 17 – Bloco de código da solução proposta para o desafio.	69
Figura 18 – Bloco de código da solução feita pelos alunos.	69
Figura 19 – Bloco de código de demonstração da criação e da execução de funções no Scratch.	70
Figura 20 – Distribuição de alunos por sexo.	73
Figura 21 – Distribuição de alunos por idade.	73
Figura 22 – Distribuição de renda por faixa salarial.	74
Figura 23 – Distribuição de nível de escolaridade dos pais por aluno.	74
Figura 24 – Distribuição de alunos que residem em russas.	75
Figura 25 – Distribuição de transporte utilizado para chegar à escola por aluno.	75
Figura 26 – Distribuição de alunos por tempo médio para chegar à escola.	76
Figura 27 – Distribuição de alunos repetentes.	76
Figura 28 – Contagem de alunos que gostam de matemática.	77

Figura 29 – Contagem de alunos que tem acesso a computador em casa.	78
Figura 30 – Contagem de alunos que tem acesso a internet em casa.	78
Figura 31 – Distribuição de alunos por nível de conhecimento em informática básica. . .	79
Figura 32 – Distribuição de alunos que já tiveram contato com programação.	79
Figura 33 – Distribuição de alunos que já tiveram contato com programação visual. . . .	80
Figura 34 – Relação entre alunos que tiveram contato com programação e acesso a computador.	82
Figura 35 – Relação entre alunos que tiveram contato com programação visual e acesso a computador.	83
Figura 36 – Relação entre os níveis de conhecimento em informática dos alunos e acesso a computador.	84
Figura 37 – Questão 1 do modelo dTECT em relação à qualidade institucional.	85
Figura 38 – Questão 2 do modelo dTECT em relação à qualidade institucional.	85
Figura 39 – Questão 3 do modelo dTECT em relação à qualidade institucional.	86
Figura 40 – Questão 4 do modelo dTECT em relação à experiência em computação dos alunos.	87
Figura 41 – Questão 5 do modelo dTECT em relação à experiência em computação dos alunos.	87
Figura 42 – Questão 6 do modelo dTECT em relação à experiência em computação dos alunos.	88
Figura 43 – Questão 7 do modelo dTECT em relação à experiência em computação dos alunos.	88
Figura 44 – Questão 8 do modelo dTECT em relação à experiência em computação dos alunos.	89
Figura 45 – Questão 9 do modelo dTECT em relação à experiência em computação dos alunos.	89
Figura 46 – Questão 10 do modelo dTECT em relação à percepção de aprendizado dos alunos.	90
Figura 47 – Questão 11 do modelo dTECT em relação à percepção de aprendizado dos alunos.	91
Figura 48 – Questão 12 do modelo dTECT em relação à percepção de aprendizado dos alunos.	91

Figura 49 – Questão 13 do modelo dTECT em relação à percepção de aprendizado dos alunos.	92
Figura 50 – Quantidade de alunos por tempo de uso do computador.	93
Figura 51 – Quantidade de alunos que pretendem seguir carreira na programação.	98
Figura 52 – Tabela de estatística descritiva sobre as notas do alunos participantes da dinâmica final	101
Figura 53 – Tabela de estatística descritiva sobre as notas do alunos em relação ao acesso a computador	102

LISTA DE TABELAS

Tabela 1 – Comparação entre os trabalhos relacionados e o presente trabalho	44
Tabela 2 – Cronograma das aulas	47
Tabela 3 – Questionário inicial	55
Tabela 4 – Questionário final	58

LISTA DE ABREVIATURAS E SIGLAS

AVA	Ambiente Virtual de Aprendizagem
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
EEMTI	Escolas de Ensino Médio em Tempo Integral
FUNCAP	Fundação Cearense de Apoio ao Desenvolvimento Científico
MEC	Ministério da Educação
POO	Programação Orientada a Objeto
SBIE	Simpósio Brasileiro de Informática na Educação
SEDUC	Secretaria da Educação
TCC	Trabalho de Conclusão de Curso
UFC	Universidade Federal do Ceará

SUMÁRIO

1	INTRODUÇÃO	16
1.1	Motivação e oportunidades	17
1.2	Objetivos	18
1.3	Organização	19
2	FUNDAMENTAÇÃO TEÓRICA	20
2.1	Introdução	20
2.2	Pensamento Computacional	20
2.2.1	<i>Decomposição</i>	22
2.2.2	<i>Reconhecimento de padrões</i>	22
2.2.3	<i>Abstração</i>	23
2.2.4	<i>Algoritmos</i>	23
2.2.5	<i>Considerações</i>	24
2.3	Scratch	24
2.4	Programe_CE	25
2.4.1	<i>Módulo 1</i>	26
2.4.2	<i>Módulo 2</i>	26
2.4.3	<i>Módulo 3</i>	26
2.4.4	<i>Considerações</i>	27
2.5	Projeto LearningLab	27
2.6	Modelo de avaliação dTECT	28
2.7	Considerações finais	29
3	REVISÃO BIBLIOGRÁFICA	30
3.1	Introdução	30
3.2	Metodologia da pesquisa	30
3.2.1	<i>Identificação da problemática</i>	30
3.2.2	<i>Escolha do repositório</i>	31
3.2.3	<i>Estabelecer critérios para inclusão dos artigos da pesquisa</i>	31
3.2.4	<i>Caracterizar os artigos</i>	31
3.2.5	<i>Analisar os artigos e discutir os dados</i>	32
3.2.6	<i>Resultados da análise</i>	32

3.2.7	<i>Apresentar conclusões</i>	32
3.3	Trabalhos relacionados	33
3.3.1	<i>Desenvolvendo o raciocínio lógico no ensino médio: uma proposta utilizando a ferramenta Scratch (MOTA et al., 2014)</i>	33
3.3.2	<i>Programação visual para introdução ao ensino de programação na educação superior: uma análise prática (RIBAS et al., 2016)</i>	34
3.3.3	<i>O uso de softwares educativos para introdução de lógica de programação no ensino de base e superior (VIANA; PORTELA, 2019).</i>	37
3.3.4	<i>O ensino da programação e o desenvolvimento do pensamento lógico: uma revisão sistemática de literatura (SHIMASAKI; PRADO, 2021)</i>	40
3.3.5	<i>Instituto de Hackers: o Pensamento Computacional aplicado ao Ensino Médio Integrado Profissionalizante (RODRIGUES; SOUZA, 2021)</i>	41
3.3.6	<i>Análise comparativa</i>	43
3.4	Considerações finais	44
4	METODOLOGIA	45
4.1	Introdução	45
4.2	Planejamento das Aulas	45
4.2.1	<i>Questões éticas</i>	45
4.2.2	<i>Ementa</i>	46
4.2.3	<i>Cronograma</i>	47
4.2.4	<i>Material utilizado</i>	50
4.2.5	<i>Metodologia das aulas</i>	51
4.2.6	<i>Proposta de avaliação</i>	52
4.2.6.1	<i>Avaliação de aprendizado</i>	52
4.2.6.2	<i>Avaliação do experimento</i>	54
4.2.6.2.1	Questionário inicial	54
4.2.6.2.2	Questionário final	58
4.3	Aplicação das Aulas	61
4.3.1	<i>Perfil da escola</i>	61
4.3.2	<i>Perfil do aluno</i>	61
4.3.3	<i>Descrição das Aulas</i>	62
4.4	Considerações finais	70

5	RESULTADOS	72
5.1	Introdução	72
5.2	Questionário inicial	72
5.2.1	<i>Questões sociodemográficas</i>	73
5.2.2	<i>Questões de pesquisa</i>	77
5.2.3	<i>Análise sobre questionário inicial</i>	81
5.3	Questionário final	84
5.3.1	<i>Questões do modelo dTECT</i>	84
5.3.2	<i>Questões de pesquisa</i>	92
5.3.3	<i>Análise sobre questionário final</i>	99
5.4	Dinâmica final	100
5.4.1	<i>Resultados da dinâmica final</i>	100
5.4.2	<i>Análise comparativa dos resultados da dinâmica</i>	101
5.5	Considerações finais	103
6	CONCLUSÃO	105
6.1	Principais contribuições	105
6.2	Trabalhos futuros	106
	REFERÊNCIAS	108
	APÊNDICE A –SLIDES UTILIZADOS NAS AULAS	111
	APÊNDICE B –PROPOSTA DE DINÂMICA PARA ATIVIDADE FINAL	152
	APÊNDICE C –RESULTADOS COLETADOS	155
	ANEXO A –TERMO DE CONSENTIMENTO LEARNINGLAB	170
	ANEXO B –TERMO DE ASSENTIMENTO LEARNINGLAB	171

1 INTRODUÇÃO

O século XX foi marcado por uma revolução tecnológica que transformou radicalmente a sociedade. A invenção da televisão em 1923, por exemplo, revolucionou a forma como consumimos informação e entretenimento (FISHER; FISHER, 1997). No entanto, foi a computação que impulsionou uma verdadeira revolução digital.

Conforme apresentado em Petzold (1999), os primeiros computadores modernos surgiram na década de 1940, mas foi a invenção dos transistores nos anos 1950 que possibilitou o desenvolvimento de máquinas menores, mais eficientes e acessíveis. Esse avanço na eletrônica pavimentou o caminho para a popularização dos computadores pessoais, que se concretizou na década de 1970, com o lançamento dos primeiros modelos domésticos em 1975.

As bases teóricas da computação moderna foram estabelecidas por grandes nomes como Claude Shannon, que aplicou a lógica booleana aos circuitos elétricos, e Alan Turing, que formalizou o conceito de algoritmo e a máquina de Turing. Essas contribuições foram fundamentais para o desenvolvimento dos computadores modernos e para a criação do conceito de programa armazenado, que é a base da arquitetura dos computadores atuais (PETZOLD, 1999).

A partir dos anos 1960, a internet começou a se formar como ARPANET, evoluindo para a World Wide Web em 1989 por Tim Berners-Lee. Este desenvolvimento transformou a internet em uma ferramenta essencial para a comunicação global, conectando pessoas, empresas e governos. Esses avanços não só facilitaram a comunicação e o acesso à informação, mas também impulsionaram novas indústrias e mudaram a forma como trabalhamos, aprendemos e interagimos, prometendo um futuro de contínua inovação tecnológica (LINS, 2013).

Diante desse cenário de constante evolução tecnológica, torna-se essencial que as futuras gerações estejam preparadas para lidar com essas inovações. O ensino de computação nas escolas desempenha um papel crucial nesse contexto. Ao introduzir conceitos de programação, Pensamento Computacional e uso crítico da tecnologia desde cedo, as escolas capacitam os alunos a serem não apenas consumidores, mas também criadores de tecnologia, como descrito em um dos três pilares da Computação Criativa em Brennan *et al.* (2014).

O Pensamento Computacional, em particular, é uma habilidade fundamental que envolve a resolução de problemas de maneira lógica e estruturada. Ele se baseia em quatro pilares principais: decomposição, reconhecimento de padrões, abstração e pensamento algorítmico. A decomposição permite dividir problemas complexos em partes menores e mais gerenciáveis. O

reconhecimento de padrões ajuda a identificar semelhanças e diferenças que podem simplificar a solução de problemas. A abstração foca nas informações essenciais, ignorando detalhes irrelevantes, enquanto o pensamento algorítmico envolve a criação de passos claros e definidos para resolver um problema (WING, 2006; BARR *et al.*, 2011).

A Base Nacional Comum Curricular (BNCC) de Computação (Ministério da Educação; BNCC, 2022), estabelece diretrizes para o ensino de computação ao longo da educação básica, promovendo o desenvolvimento do Pensamento Computacional e do raciocínio lógico. Essas competências são fundamentais para a formação de cidadãos preparados para enfrentar os desafios e aproveitar as oportunidades da sociedade digital.

Portanto, a integração do ensino de computação nas escolas representa um grande desafio, mas também uma oportunidade única para preparar os estudantes para o futuro. Somente assim será possível garantir que todos os alunos tenham a oportunidade de desenvolver as habilidades essenciais para prosperar em um mundo cada vez mais tecnológico e interconectado.

1.1 Motivação e oportunidades

A motivação inicial para a escolha do tema deste trabalho surgiu da experiência do autor deste trabalho como tutor no projeto *Programe_CE* (SEDUC/CE; FUNCAP, 2021), uma iniciativa do Governo do Ceará em parceria com a Universidade Federal do Ceará. Durante um ano e meio, o projeto proporcionou ao autor uma imersão no ensino dos fundamentos da computação em Escolas de Ensino Médio em Tempo Integral (EEMTI), por meio de aulas práticas em laboratórios de informática.

Nesse contexto, as atividades desenvolvidas envolviam a resolução de problemas em diversas áreas, utilizando conceitos de matemática, lógica e programação. O foco era desenvolver habilidades de Pensamento Computacional e *Design Thinking*, com o uso de ferramentas como Scratch e Python.

A experiência de acompanhar os alunos nessa jornada de ensino-aprendizagem revelou a necessidade de aprofundar a investigação sobre o uso e o ensino de tecnologia no ensino médio, especialmente no que diz respeito ao desenvolvimento do Pensamento Computacional.

Nesse contexto, o projeto *LearningLab* (2023) (Laboratório de Ensino e Pesquisa de Tecnologias Alinhadas à Gestão do Conhecimento e Inovação em Processos de Software) se alinha perfeitamente com a proposta de difundir o conhecimento gerado no âmbito acadêmico. O projeto promove a participação dos seus membros em cursos práticos, palestras e atividades

de pesquisa e extensão voltadas para os cursos de Ciência da Computação e Engenharia de Software.

O autor desta pesquisa, que também é membro do projeto no setor de extensão, colaborou com o LearningLab, o orientador da pesquisa e uma escola de ensino médio no desenvolvimento e aplicação de um experimento. Este experimento, no formato de um curso de programação utilizando a ferramenta Scratch, teve como objetivo promover o Pensamento Computacional entre os alunos do ensino médio.

1.2 Objetivos

Diante desse cenário, a presente pesquisa busca contribuir para o campo da educação em computação, investigando o uso do Scratch no desenvolvimento do Pensamento Computacional em alunos do ensino médio.

O objetivo geral deste trabalho é explorar o uso do Scratch como Ferramenta para Ensino de Pensamento Computacional através da aplicação de aulas de programação com alunos do ensino médio, buscando identificar as principais dificuldades enfrentadas pelos alunos ao aprender programação visual com Scratch, e como essas dificuldades podem ser mitigadas.

Para atingir o objetivo geral, são propostos os seguintes objetivos específicos:

- **Desenvolver uma metodologia:** Elaborar e implementar uma sequência didática utilizando o Scratch, com foco no desenvolvimento de habilidades de Pensamento Computacional e de conceitos de fundamentos de programação.
- **Aplicar a metodologia:** Aplicar a sequência didática em uma turma de alunos do ensino médio, coletando dados sobre seus perfis socioeconômicos, experiências prévias com computação e suas percepções sobre as atividades por meio de questionários e da análise de projetos desenvolvidos pelos alunos.
- **Avaliar a metodologia aplicada:** Analisar quantitativa e qualitativamente os dados coletados a fim de identificar o impacto da metodologia proposta no desenvolvimento das habilidades de Pensamento Computacional dos alunos e as percepções dos alunos sobre o processo de aprendizagem.
- **Identificar desafios e oportunidades:** Identificar os principais desafios e oportunidades relacionados à integração do desenvolvimento do Pensamento Computacional com Scratch no contexto educacional brasileiro, com base nos resultados da pesquisa.

1.3 Organização

Este trabalho está estruturado em seis capítulos. O Capítulo 2 relata os principais conceitos e ferramentas abordadas neste trabalho. Em seguida, o Capítulo 3 apresenta uma análise dos estudos anteriores sobre o tema. A metodologia empregada na pesquisa, uma pesquisa-ação com abordagem mista, é detalhada no Capítulo 4. Os resultados obtidos a partir da aplicação da metodologia são apresentados e discutidos no Capítulo 5. Por fim, o Capítulo 6 apresenta as conclusões da pesquisa, destacando as principais contribuições e limitações do estudo, além de sugerir direções para futuras pesquisas.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Introdução

Este capítulo expõe os principais conceitos que fundamentam a presente pesquisa, com o propósito de facilitar a compreensão do estudo. Na Seção 2.2, discute-se o conceito de Pensamento Computacional. Em seguida, a Seção 2.3 aborda a ferramenta Scratch e sua abordagem de programação estruturada voltada para o ensino. A Seção 2.4 apresenta o projeto do Governo do Estado do Ceará chamado Progame_CE que abordou o Pensamento Computacional em escolas de ensino médio utilizando o *software* Scratch. Na Seção 2.5 apresenta o projeto LearningLab como facilitador para esta pesquisa. Em seguida é descrito o modelo de avaliação dTECT na Seção 2.6 e por fim são apresentadas as considerações finais na Seção 2.7.

2.2 Pensamento Computacional

O Pensamento Computacional refere-se a uma forma de abordar problemas e processos que incorpora conceitos e técnicas fundamentais da ciência da computação. Essa abordagem envolve a capacidade de decompor um problema complexo em partes menores, identificar padrões, desenvolver algoritmos para resolver essas partes e projetar soluções eficientes.

Para Wing (2006), o Pensamento Computacional envolve a solução de problemas, o design de sistemas e a compreensão de comportamento humano, aproveitando os conceitos fundamentais da ciência da computação. Para isso, o Pensamento Computacional inclui uma série de ferramentas mentais que refletem a vastidão do campo da ciência da computação como capacidade de abstração, de pensar recursivamente, de decompor problemas grandes em problemas menores, de criar mecanismos de prevenção, dentre outros. Pensamento Computacional é uma habilidade fundamental para todos, não somente para cientistas da computação. Além da leitura, da escrita e da aritmética, o Pensamento Computacional deveria ser uma habilidade analítica de todas as pessoas.

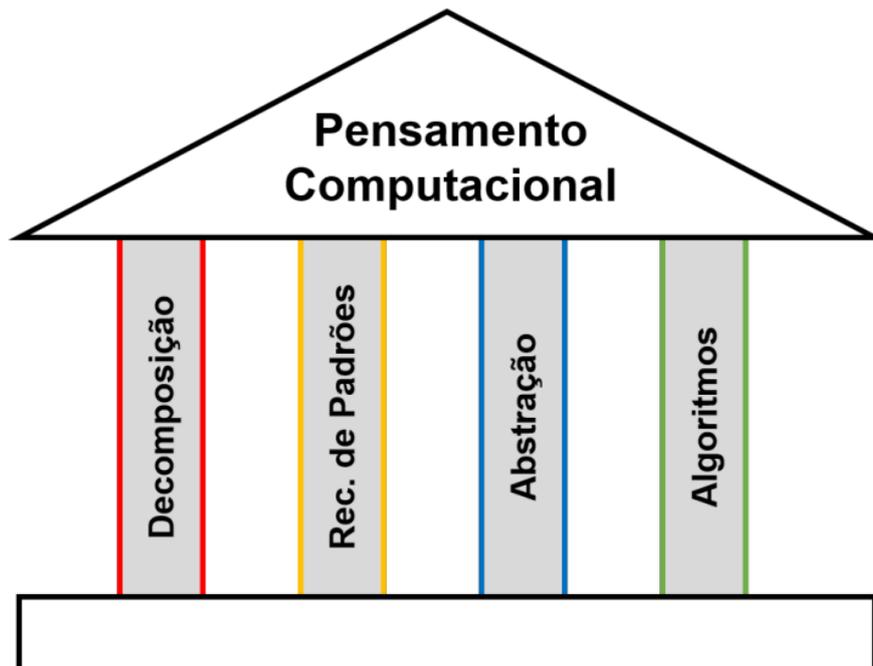
De acordo com Barr *et al.* (2011), A importância do Pensamento Computacional reside em sua aplicação da tecnologia informática a praticamente todos os campos de estudo, o que alterou a forma como o trabalho é realizado hoje. Embora a mente humana seja, de longe, a ferramenta mais poderosa para a resolução de problemas que possuímos, a capacidade de potencializar o poder do pensamento humano com computadores e outras ferramentas digitais

tornou-se uma parte essencial de nossa vida cotidiana e de nosso trabalho.

Contudo, é importante frisar que Pensamento Computacional é diferente de informática. A informática utiliza-se de aparelhos computacionais e da execução de programas para resolver uma tarefa enquanto o Pensamento Computacional, embora utilize-se muitas vezes de *softwares*, não obrigatoriamente precisará deles para alcançar seus objetivos. Como já foi dito, o Pensamento Computacional é um modo de pensar sobre problema utilizando a lógica algorítmica e as metodologias aplicáveis na computação, ou seja, é uma maneira de pensar soluções para problemas sem necessariamente utilizar uma máquina como auxiliadora.

Pesquisas lideradas pela instituição Code.org (2023), Liukas (2015), Learning (2023), Grover e Pea (2013) e o guia para professores de Csizmadia *et al.* (2015) originaram o que foi chamado de “Quatro Pilares do Pensamento Computacional” e são sintetizados nos trabalhos de Brackmann (2017) e Vicari *et al.* (2018). Como foi citado, as bases do Pensamento Computacional podem ser resumidas em quatro pilares principais: Decomposição, Reconhecimento de Padrões, Abstração e Algoritmos (Figura 1). Assim, nas seções seguintes são descritos mais detalhadamente cada um desses conceitos.

Figura 1 – Pilares do Pensamento Computacional



Fonte: Brackmann (2017).

2.2.1 Decomposição

Para Liukas (2015), esse é um procedimento no qual os problemas são decompostos em partes menores. A autora ilustra esse conceito por meio de exemplos simples, como a decomposição de refeições, receitas culinárias e as fases de um jogo. Trata-se de dividir um problema ou sistema complexo em componentes menores, tornando-os mais compreensíveis e mais fáceis de se resolver. Quando um problema não é decomposto, sua resolução se torna mais difícil, visto que, ao lidar com muitos estágios diferentes de um mesmo problema, seu gerenciamento acaba escalando em complexidade. Uma abordagem para simplificar a resolução desse problema é dividi-lo em partes menores e resolvê-las individualmente, gerando uma modularização do problema e garantindo uma maior atenção aos detalhes de cada parte para se chegar a uma solução geral. Ela ainda destaca que programadores frequentemente utilizam essa técnica para quebrar um algoritmo em várias partes menores para facilitar sua compreensão e manutenção. Na programação, a decomposição pode ser exemplificada por meio de elementos como funções, procedimentos, objetos, módulos, entre outros, presentes no código-fonte.

2.2.2 Reconhecimento de padrões

De acordo com Brackmann (2017), o reconhecimento de padrões é um modo de resolver problemas utilizando soluções que já foram estabelecidas com base em experiências anteriores. Nesse método é de extrema importância fazer questionamentos como “Esse problema é similar a um outro que já tenha resolvido?” ou “Como esse problema é diferente de outros que já foram resolvidos?” para que se aproveite de estratégias já estabelecidas para resolver novos desafios. Algoritmos projetados para resolver um problema específico têm a capacidade de ser adaptados para lidar com vários problemas semelhantes. Quando necessário, o algoritmo pode aplicar uma solução de maneira generalizada.

Generalização é o conceito principal envolvido no reconhecimento de padrões. A generalização, de forma simples, é a capacidade de aplicar o que aprendemos em uma situação específica a novas situações similares. É como usar conhecimento adquirido em uma situação anterior para lidar com situações relacionadas que não foram vistas antes. Em termos de algoritmo, um modelo generalizado é capaz de fazer previsões precisas mesmo em dados que não foram usados durante o treinamento pois conseguem mapear padrões de dados e adaptar para novas situações.

2.2.3 Abstração

Segundo Wing (2006), a abstração é o conceito central do Pensamento Computacional. Ela descreve a abstração como a habilidade de identificar e focar nas informações essenciais de um problema, deixando de lado os detalhes irrelevantes. Esse método simplifica a complexidade, permitindo que os programadores concentrem seus esforços nos aspectos mais críticos do problema. Por exemplo, ao desenvolver um jogo, a abstração pode consistir em definir as regras principais e ignorar detalhes menores que não influenciam a jogabilidade. O processo de abstrair informações é utilizado em diversos momentos, tais como:

1. Na escrita do algoritmo e suas iterações, abstraindo detalhes técnicos;
2. Na seleção dos dados importantes em uma base de dados robusta;
3. Na diferença da natureza de um indivíduo em comparação a um robô;
4. Na compreensão e organização de módulos em um sistema;
5. Na definição de regras de um jogo, ignorando detalhes menores como animações e gráficos.

Todas essas etapas envolvem reconhecer padrões e, principalmente, filtrar as informações intrínseca para se chegar na modelagem certa para a execução de determinada tarefa. Essa é a grande tarefa da abstração: decidir, em meio a um sistema complexo e com inúmeras variáveis, quais as informações são relevantes e quais podem ser desconsideradas para facilitar a compreensão e modelagem correta de determinado problema e, conseqüentemente, contribuir com sua resolução.

2.2.4 Algoritmos

A capacidade de resolver problemas de forma sistemática e eficiente é fundamental em diversas áreas do conhecimento. Csizmadia *et al.* (2015), em sua obra *Computational thinking - A guide for teachers*, enfatiza a importância do pensamento algorítmico. Em tradução livre, o autor destaca:

“O pensamento algorítmico precisa entrar em ação quando problemas similares precisam ser resolvidos repetidamente. Eles não precisam ser pensados novamente toda vez. É necessária uma solução que funcione todas as vezes. Aprender algoritmos para realizar multiplicação ou divisão na escola é um exemplo. Se regras simples forem seguidas com precisão, por um computador ou uma pessoa, a solução para qualquer multiplicação pode ser encontrada. Uma vez que o algoritmo é compreendido, não é necessário trabalhá-lo do zero para cada novo problema. O pensamento algorítmico é a capacidade de pensar em termos de sequências e regras como uma maneira de resolver problemas ou entender situações. É uma habilidade fundamental que os alunos desenvolvem ao aprenderem a escrever seus próprios programas de computador.”

Na área da informática, a definição clássica refere-se a algoritmo como a representação abstrata de um processo que recebe uma entrada, executa uma sequência finita de etapas e gera uma saída que atende a um objetivo específico. Assim, algoritmo é um modo de encadear operações para se chegar na resolução de um problema.

2.2.5 Considerações

Portanto, visto os quatro pilares do Pensamento Computacional, pode-se perceber que essa habilidade vai além das noções de programação. Ela envolve uma mentalidade analítica e uma abordagem estruturada para resolução de problemas, que pode ser aplicada em diversas áreas da vida, não apenas na programação ou na ciência da computação. É uma habilidade essencial no mundo moderno, onde a tecnologia desempenha um papel cada vez mais central em muitas atividades do cotidiano.

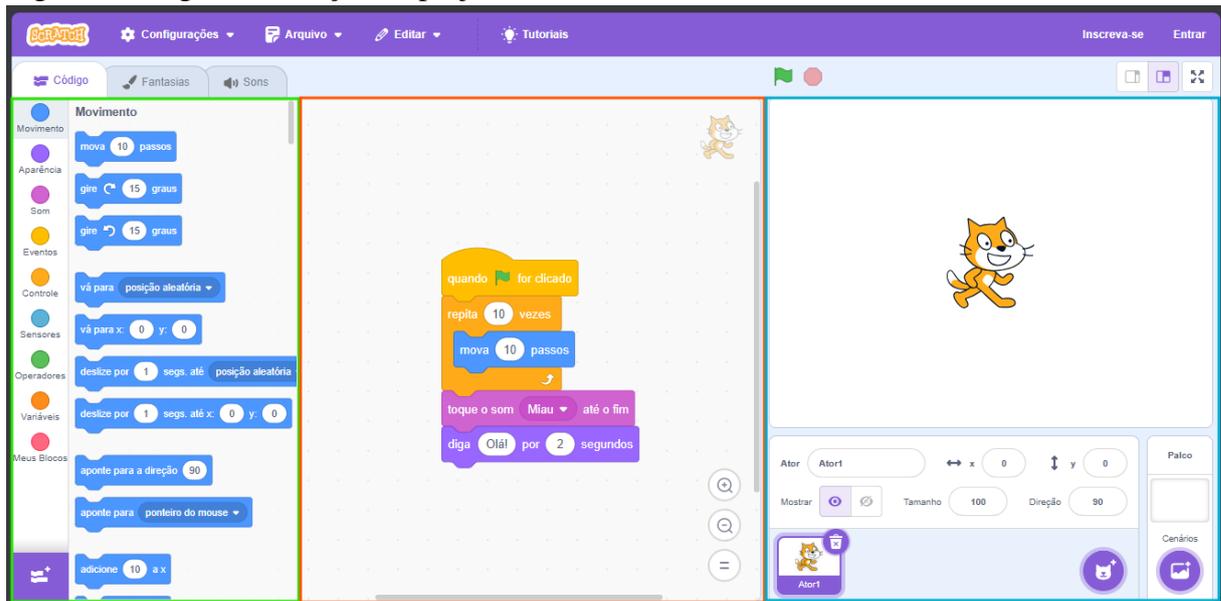
2.3 Scratch

Scratch é uma linguagem de programação visual e uma plataforma online criada pelo MIT Media Lab, liderado pelo professor Mitchel Resnick. Desde seu lançamento em 2007, Scratch tem como objetivo tornar a programação acessível e envolvente para crianças e jovens. Com uma interface intuitiva baseada em blocos, permite aos usuários desenvolver histórias interativas, jogos e animações, tornando mais fácil o entendimento dos conceitos de programação.

Segundo o site oficial do Scratch (2023b), a ferramenta busca principalmente promover o Pensamento Computacional e as habilidades de resolução de problemas. Ao criar projetos no Scratch, os alunos aprendem a dividir problemas complexos em partes menores, identificar padrões, abstrair conceitos e desenvolver algoritmos. Essas competências são essenciais não apenas para a programação, mas também para diversas áreas do conhecimento e para o cotidiano.

O Scratch utiliza uma abordagem de programação baseada em blocos. Conforme ilustrado na Figura 2, a interface pode ser dividida em três áreas principais. À esquerda, encontram-se os blocos coloridos que representam os comandos. Esses blocos podem ser arrastados para o centro da tela, onde são manipulados e encaixados com outros blocos para formar o código do programa. Quando o código é executado, o personagem localizado à direita da tela realiza as ações programadas nos blocos.

Figura 2 – Página de criação de projetos no Scratch



Fonte: Página de criação de (SCRATCH, 2023a).

Essa metodologia elimina a necessidade de digitação de código, reduzindo erros sintáticos e permitindo que os alunos se concentrem na lógica e no design de seus projetos. Além disso, a plataforma oferece uma vasta biblioteca de recursos, incluindo tutoriais, exemplos de projetos e uma comunidade ativa onde os usuários podem compartilhar e remixar trabalhos.

Portanto, o Scratch se destaca como uma ferramenta poderosa para a educação em programação, oferecendo uma maneira acessível e envolvente para que crianças e jovens desenvolvam habilidades essenciais para o século XXI. Ao promover o Pensamento Computacional e a criatividade, Scratch prepara os alunos para enfrentar os desafios do futuro, seja na área de tecnologia ou em qualquer outra disciplina.

2.4 Programe_CE

O Programe_CE foi desenvolvido pelo Governo do Ceará por meio da Secretaria da Educação (SEDUC) e da Fundação Cearense de Apoio ao Desenvolvimento Científico (FUNCAP) em parceria com o Programa Cientista Chefe. Teve início durante o segundo semestre de 2021 e encerrou suas atividades ao final do ano de 2022. O Programe_CE foi de um projeto piloto que, por meio da oferta de uma disciplina eletiva, levou o Pensamento Computacional, a Ciências de Dados e a Linguagem de Programação para alunos das EEMTI, possibilitando mais perspectivas acadêmicas e profissionais aos jovens.

O projeto foi implementado em várias EEMTI do estado do Ceará ao longo de três

semestres, e seus materiais foram organizados em três módulos, cada um acompanhado de apostilas. Os conteúdos abordados em cada módulo estão dispostos nas Seções 2.4.1 a 2.4.3.

2.4.1 Módulo 1

Ao longo da implementação do Módulo 1, a estratégia envolveu o estímulo à lógica proposicional e a conceitos matemáticos relacionados à computação nos primeiros meses. Posteriormente, foram introduzidos conceitos essenciais de computação, tais como hardware e *software*, noções de algoritmos e fluxogramas. O foco foi direcionado à resolução de desafios, incentivando o desenvolvimento do raciocínio lógico por meio da aplicação dos novos conhecimentos adquiridos.

2.4.2 Módulo 2

No Módulo 2, os conceitos abordados no último capítulo do módulo anterior foram aprofundados. Nessa etapa, as principais ferramentas utilizadas foram o Scratch, um ambiente de programação estruturada por blocos que permite explorar ideias em problemas e cenários lúdicos e a linguagem de programação convencional Python. A utilização do Scratch teve como objetivo iniciar a exploração criativa, desenvolver o Pensamento Computacional e o raciocínio algorítmico, enquanto a linguagem Python foi empregada para aprimorar os conceitos básicos de programação aprendidos, como laços de repetição, condicionais, variáveis e funções, e aplicá-los em desafios matemáticos. Assim, o Módulo 2 também proporcionou aos estudantes o primeiro contato com uma linguagem de programação utilizada no mercado de trabalho.

2.4.3 Módulo 3

Por fim, no Módulo 3, foram abordados os conceitos do *Design Thinking*, uma abordagem criativa e centrada no usuário para resolver problemas, com foco em compreender as necessidades das pessoas para as quais você está projetando. Esse processo inclui empatia, geração de ideias, prototipagem e teste. Em seguida, foi apresentado o MIT App Inventor 2, uma ferramenta de código aberto usada para desenvolver aplicativos móveis para Android e que utiliza de conceitos semelhantes à programação estruturada em blocos, como visto no Scratch. Neste módulo, houve o acompanhamento dos alunos na criação de projetos próprios seguindo os conceitos do *Design Thinking* e posteriormente, sua implementação utilizando o MIT App

Inventor.

2.4.4 Considerações

Durante os três módulos, o projeto buscou ampliar o contato dos alunos com a tecnologia por meio de aulas presenciais realizadas nos laboratórios de informática. Essa iniciativa vai de encontro com o objetivo de tornar o acesso a informação mais acessível, estimular a criatividade e o raciocínio lógico, e promover o desenvolvimento do Pensamento Computacional no ensino básico. Além disso, teve um impacto significativo na comunidade científica do Estado do Ceará, ao capacitar novos tutores e incentivar a participação direta de alunos graduandos e graduados na área da computação nas salas de aula. Também é importante destacar que o autor desta pesquisa teve sua primeira experiência em docência como tutor por meio deste projeto, o que despertou seu interesse pela área acadêmica. Essa experiência motivou a busca por novas formas de contribuir para promover a inovação científica e tecnológica, servindo como inspiração para a elaboração desta monografia.

2.5 Projeto LearningLab

O LearningLab (2023) é um projeto de pesquisa e extensão situado no Campus de Russas da UFC. Seu principal objetivo é apoiar a formação dos estudantes dos cursos de Ciência da Computação e Engenharia de Software, implementando metodologias que garantam a permanência e conclusão dos cursos. Além disso, oferece cursos práticos e tutorias para aprimorar o desempenho acadêmico e divulgar técnicas de gestão do conhecimento utilizadas no mercado, incentivando a participação dos alunos em eventos de computação.

Com a missão de transformar o conhecimento teórico em práticas aplicáveis e ajudar os alunos a ingressarem no mercado de trabalho, o LearningLab busca se estabelecer como uma referência nacional em laboratórios de ensino e pesquisa na área de tecnologia. Para atingir esses objetivos, o projeto promove a capacitação contínua, a inovação e a colaboração entre os estudantes, preparando-os para os desafios do mercado.

Para alcançar esses objetivos, o projeto é estruturado de forma que seus membros atuem em setores específicos para a promoção do conhecimento. Diversos setores estiveram envolvidos na execução do presente experimento: o setor de mídias contribuiu para a divulgação científica, o setor de processos estabeleceu as normas seguidas em cada nova atividade do projeto,

e o setor de extensão buscou levar os conhecimentos construídos na faculdade para a comunidade externa.

Nesse contexto, o autor deste trabalho desempenhou um papel ativo como membro do LearningLab no setor de extensão. A atuação do projeto junto ao autor foi crucial na elaboração, validação e execução do experimento planejado, assegurando que todas as condições legais e éticas fossem rigorosamente atendidas durante a aplicação de testes e coleta de dados. A experiência adquirida no LearningLab forneceu suporte essencial e contribuiu significativamente tanto para o sucesso do projeto quanto para a formação dos alunos e do autor deste trabalho.

2.6 Modelo de avaliação dTECT

O modelo dTECT (Evaluating TEaching CompuTing) desenvolvido por Wangenheim *et al.* (2017) é uma ferramenta desenvolvida para avaliar a qualidade de unidades instrucionais no ensino de computação para a educação básica. Este modelo baseia-se na percepção dos alunos, coletada por meio de um questionário aplicado ao final de cada unidade instrucional. O objetivo do dTECT é analisar essas unidades para avaliar a qualidade em termos de experiência de computação e percepção de aprendizagem dos estudantes.

O desenvolvimento do modelo dTECT foi sistemático e incluiu a coleta de dados de 16 estudos de caso em 13 diferentes instituições de ensino, com respostas de 477 alunos. Os resultados indicam que o modelo é confiável e possui validade de constructo, demonstrando uma correlação aceitável entre quase todos os itens do instrumento de medição. Esses resultados permitem que pesquisadores e instrutores utilizem o modelo dTECT para avaliar e melhorar as unidades instrucionais, promovendo uma adoção mais eficaz e eficiente do ensino de computação.

A utilização do modelo dTECT no presente trabalho foi crucial para a obtenção de dados precisos e confiáveis sobre a percepção dos alunos em relação ao ensino de computação. Ao quantificar a qualidade da experiência de aprendizagem e identificar pontos fortes e fracos nas atividades propostas, o dTECT permitiu uma análise aprofundada dos resultados, possibilitando a realização de ajustes e melhorias na proposta pedagógica. Dessa forma, o modelo se mostrou uma ferramenta indispensável para a validação dos resultados da pesquisa e para a geração de conhecimento sobre o tema.

2.7 Considerações finais

Neste capítulo, foram explorados os conceitos de Pensamento Computacional e seus quatro pilares fundamentais, destacando a importância do desenvolvimento dessas habilidades para a resolução de problemas em diversas áreas além da computação. Foi discutido o *software* Scratch, que utiliza a programação estruturada por blocos para ensinar conceitos de programação de forma lúdica e acessível para jovens e crianças, promovendo o Pensamento Computacional. Também foi apresentado o modelo dTECT, utilizado para avaliar a percepção de qualidade no ensino de computação nas instituições de ensino básico.

O capítulo abordou ainda o projeto estadual Programe_CE, que implementou o ensino do Pensamento Computacional e incentivou o uso de ferramentas tecnológicas como Scratch e MIT App Inventor. Esse projeto visou impulsionar a inovação tecnológica de forma prática, oferecendo material validado e gratuito, e contribuiu significativamente para o desenvolvimento da presente pesquisa. Por fim, foi apresentado o LearningLab, um projeto de pesquisa e extensão no Campus de Russas da UFC, discutido seus objetivos e abordando sua atuação junto ao autor desta pesquisa para o experimento realizado.

3 REVISÃO BIBLIOGRÁFICA

3.1 Introdução

Por um longo período, a utilização do Pensamento Computacional esteve atrelada somente às aulas das disciplinas da área de ciências exatas, como Matemática e Física. No entanto, devido ao progresso tecnológico e às mudanças sociais em curso, as instituições educacionais precisaram-se adaptar e ampliar esse enfoque para outras disciplinas, promovendo abordagens interdisciplinares. Visto isso, a ferramenta Scratch surge como um auxiliador para o desenvolvimento dessa competência já que oferece um ambiente lúdico e educativo para desenvolver o raciocínio lógico. Com isso, é de grande importância para o presente trabalho entender como pesquisadores optaram por abordar o *software* Scratch em sala de aula e que resultados eles obtiveram. Assim, este capítulo apresenta a metodologia que foi adotada para a pesquisa dos artigos relacionados descritos por etapas na Seção 3.2, bem como seus respectivos resumos descritos na Seção 3.3 e, por fim, são expostas as considerações finais na Seção 3.4.

3.2 Metodologia da pesquisa

O desenvolvimento da pesquisa por trabalhos relacionados ao tema principal segue uma ordem bem estabelecida de etapas: a descrição do problema (Seção 3.2.1), a escolha do repositório base para a pesquisa (Seção 3.2.2), o estabelecimento de critérios de inclusão de artigos na pesquisa (Seção 3.2.3), a caracterização dos artigos selecionados (Seção 3.2.4), a organização dos dados extraídos da pesquisa (Seção 3.2.5), os resultados obtidos que demonstra os resultados das análises dos artigos (Seção 3.2.6) e por fim, a apresentação das principais conclusões acerca dos resultados da pesquisa e como será tratado em futuras aplicações (Seção 3.2.7). Contudo, é importante destacar que a metodologia por etapas abordada nesse capítulo da pesquisa foi fortemente embasada na metodologia descrita no artigo de revisão sistemática da literatura de Shimasaki e Prado (2021), que também faz parte dessa pesquisa.

3.2.1 Identificação da problemática

Atualmente, a tecnologia se faz presente no cotidiano de todas as pessoas. Ela é utilizada para as mais variadas tarefas, possibilitando facilitar tarefas complicadas e auxiliando na criação de novas formas de resolução de problemas. Visto isso, o modelo de ensino em todos os

níveis de escolaridade vem-se modificando com o tempo. Porém, o modelo tradicional de ensino ainda é predominante, tornando certos assuntos pouco atrativos e, conseqüentemente, mais difíceis de serem ensinados pelos professores e aprendidos pelos educandos. Logo, esta pesquisa busca analisar os trabalhos já existentes na literatura com o intuito de identificar diferentes metodologias e motivar os alunos utilizando novas ferramentas de ensino.

3.2.2 Escolha do repositório

A realização da pesquisa teve como enfoque o repositório do Portal de Periódicos da CAPES (Ministério da Educação; CAPES, 2020) mantido pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), vinculada ao Ministério da Educação (MEC). O Portal de Periódicos busca promover a democratização do acesso online à informação científica para os programas de pós-graduação no Brasil. O Portal de Periódicos da CAPES fornece uma variedade de conteúdos acadêmicos e científicos em formato digital, tais como diversas bases de dados com trabalhos acadêmicos e científicos, artigos de congressos e revistas, patentes, teses e dissertações, fornecendo uma abundância de materiais de todas as áreas do conhecimento.

3.2.3 Estabelecer critérios para inclusão dos artigos da pesquisa

Foram definidos os seguintes critérios para a inclusão dos artigos a serem analisados para a pesquisa em questão:

1. Artigos publicados no período de 2011 a 2023;
2. A priori, artigos disponíveis no repositório da CAPES;
3. Publicações em português;
4. Artigos com qualquer campo que contenha “Scratch” e qualquer campo que contenha “Educação Superior”;
5. Excluídas pesquisas que não abordam a temática pesquisada.

3.2.4 Caracterizar os artigos

Inicialmente foram encontrados vinte e dois artigos no período de 2011 a 2023 e, seguindo os critérios de inclusão, foram selecionados apenas aqueles que tinham como temática principal o uso do *software* Scratch voltado ao desenvolvimento do Pensamento Computacional e do raciocínio lógico. Desta forma, conseguiu-se caracterizar cinco artigos que se relacionavam

diretamente com a proposta da presente pesquisa. Dentre os resultados filtrados, o trabalho intitulado “O Ensino da Programação e o Desenvolvimento do Pensamento Lógico: uma Revisão Sistemática de Literatura” Shimasaki e Prado (2021) se destaca não só por disponibilizar uma série de resumos de artigos que também nortearam essa pesquisa, mas também por abordar o trabalho “Desenvolvendo o Raciocínio Lógico no Ensino Médio: uma proposta utilizando a ferramenta Scratch” Mota *et al.* (2014), fundamental para essa pesquisa. Ambos serão aprofundados na Seção 3.3.

3.2.5 *Analisar os artigos e discutir os dados*

Nesta etapa, busca-se ler os artigos selecionados e examinar as principais contribuições que cada um deles traz para o progresso da pesquisa. Dessa forma, foram organizados os resultados em resumos que podem ser vistos na Seção 3.3

3.2.6 *Resultados da análise*

Ao realizar o estudo dos artigos e trazer tudo que foi aprendido para a realidade da Universidade Federal do Ceará (UFC) — Campus Russas, foi percebido que era inoportuno aplicar os conceitos de Scratch para os alunos do campus de acordo com os resultados apresentados pelo professor da disciplina de Fundamentos de Programação. Chegou-se a essa conclusão porque a metodologia atualmente aplicada nas disciplinas de introdução a programação já supre, em grande parte, as carências e necessidades dos alunos ingressantes. Por consequência, foi mudada a abordagem que se usaria o Scratch, voltando-se o foco agora para o desenvolvimento do Pensamento Computacional para os alunos do ensino médio. Com isso, retornou-se aos critérios de inclusão e exclusão para que os resultados considerassem também o termo "Ensino Médio" em qualquer campo.

3.2.7 *Apresentar conclusões*

Desse modo, após acordado qual seria o público alvo, foi utilizado o conhecimento adquirido nas etapas anteriores para elaborar uma metodologia de aplicação do Scratch voltada para as turmas de ensino médio. Esta abordagem está descrita no Capítulo 4 deste presente trabalho e espera-se que o estudo possibilite o desdobramento de novas ideias e novos estudos envolvendo o Pensamento Computacional, o qual é um campo em constante evolução, apresentando

uma variedade de estratégias que merecem ser exploradas em pesquisas futuras.

3.3 Trabalhos relacionados

3.3.1 *Desenvolvendo o raciocínio lógico no ensino médio: uma proposta utilizando a ferramenta Scratch (MOTA et al., 2014)*

O trabalho aborda de forma sucinta a proposta de como aplicar um curso de lógica de programação utilizando o *software* Scratch para estudantes do ensino médio de uma escola pública. Também levanta hipóteses sobre seu impacto na estrutura de ensino atual e seus desdobramentos no ensino-aprendizado dos alunos e professores envolvidos.

Primeiramente, é maturado o pensamento sobre o uso do computador atualmente, visto que se tornou uma ferramenta não só para o ensino de informática, mas também para o desenvolvimento do Pensamento Computacional. Com isso, é notório que o computador passou de uma mera ferramenta para um novo modo de representar o conhecimento, gerar ideias e pensar estratégias para os problemas enfrentados. Portanto, para chegar nessa maneira de ensino, os professores vão além de simples instrutores que repassam seus conhecimentos sobre computadores. É preciso que haja o envolvimento de todos os segmentos da escola como professores, administradores, alunos e seus responsáveis para que as mudanças aconteçam e sejam perceptíveis na formação dos novos discentes.

Tendo isso em vista, uma abordagem para o ensino de programação nos níveis mais básicos de ensino pode ser a gamificação. O trabalho apresenta como sugestão o *software* Scratch que é descrito como “fácil de usar” devido ao seu ambiente de programação por blocos de comando e criação de cenários interativos, ao contrário de outras ferramentas que necessitam que se escreva o código em si. Nesse contexto, ele se torna uma ferramenta com grande potencial, já que elimina os problemas de sintaxe, e permite que os alunos foquem na resolução de problemas e no pensamento algorítmico. Também é importante pontuar que, apesar de seu ambiente mais lúdico, o Scratch abrange estruturas de repetição, estruturas condicionais, variáveis e blocos de comunicação utilizados para passagem de parâmetros para outros blocos.

O desenvolvimento sugerido das atividades com a ferramenta foram divididos em duas etapas principais. A primeira etapa consiste em aplicar oficinas de lógica de programação para um grupo selecionado de alunos e professores da escola que deseja implementar o projeto. São apresentados a todos os conceitos de lógica de programação por meio da ferramenta

e, de forma gradual, são introduzidos desafios de conceitos como por exemplo, problemas envolvendo condicional e repetição. Após a execução das oficinas e com as pessoas selecionadas devidamente preparadas para o uso da ferramenta, é iniciada a segunda etapa que visa replicar os conhecimentos de programação com Scratch no ambiente escolar e, para isso, os alunos que participaram da etapa anterior podem atuar agora como monitores. As oficinas, nessa segunda etapa, são realizadas junto ao laboratório de informática da própria escola com a frequência de duas vezes por semana. Os alunos atendidos nesta etapa cursam os dois últimos anos do ensino médio e os conteúdos agora são focados nas disciplinas de matemática e física, envolvendo problemas das áreas a serem resolvidos de forma não tradicional, e sim por meio de games e programas desenvolvidos no *software* Scratch. Para poder metrificar o projeto, são aplicados questionários antes da realização da oficina e após sua conclusão, visando identificar possíveis mudanças no pensamento sobre lógica de programação e como julgam ser seu nível de aprendizado. Também são avaliados o interesse e o envolvimento nas atividades realizadas.

Portanto, a realização do trabalho descrito e suas pesquisas podem gerar importante impacto no modo como o ensino na área de exatas é executado nas escolas e também, incorporar ao ambiente escolar o Pensamento Computacional e todas as suas potencialidades.

3.3.2 Programação visual para introdução ao ensino de programação na educação superior: uma análise prática (RIBAS et al., 2016)

O trabalho aborda de forma elaborada o uso da programação visual como metodologia de ensino para a introdução à lógica de programação em substituição a técnicas de ensino tradicionais, em cursos superiores de Ciência da Computação.

A motivação para essa abordagem visa compreender as dificuldades enfrentadas pelos alunos ao iniciar um curso de graduação nas áreas da computação, tais como assimilar conceitos abstratos e elementares e também implementá-los em seus programas. Além disso, resultados de pesquisas anteriores demonstram haver uma falta de estímulo aos estudantes, que tendem a se sentir desmotivados ao iniciarem os estudos em programação, acarretando em evasão e reprovação em cadeiras fundamentais para a progressão nos cursos de tecnologia.

Com o intuito de suprir essas deficiências, algumas alternativas estão sendo estudadas pelos professores visando contornar essas dificuldades e ao mesmo tempo estimular o Pensamento Computacional, visando sua aplicabilidade em diversas áreas além da computação. Dentre as possibilidades levantadas pelos professores, pode-se destacar a utilização de gamificação,

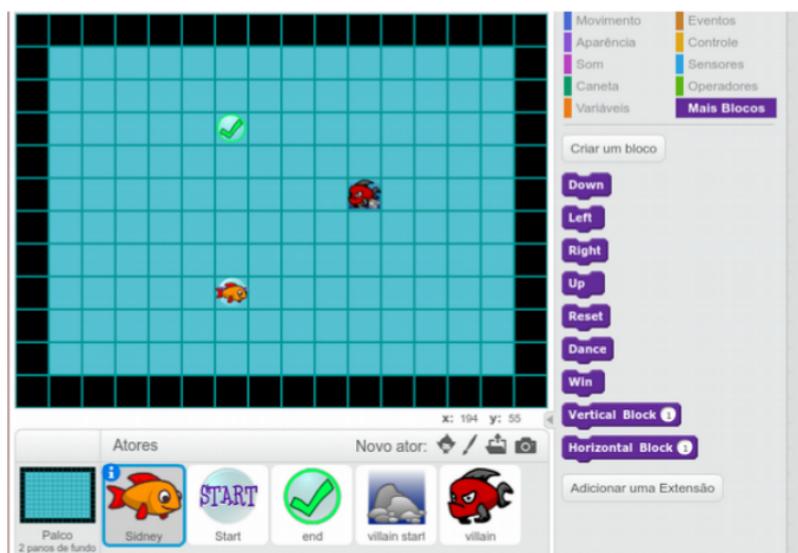
ambientes pedagógicos interativos e ferramentas de programação visual.

No artigo em questão, foi aplicada a ferramenta Scratch como recurso tecnológico que utiliza a programação estruturada em blocos para o desenvolvimento do Pensamento Computacional e da lógica de programação na turma do primeiro semestre do curso de Ciência da Computação de uma instituição de ensino. Em seguida, foram analisados os resultados obtidos para traçar um paralelo com a turma ingressante no semestre anterior. O ensino na turma anterior era baseado em Portugol, uma maneira de ensinar linguagem de programação de alto nível onde toda a sua sintaxe é implementada em português.

Como objetivo, o estudo visa o aperfeiçoamento das habilidades de lógica de programação, o mapeamento do nível de engajamento na disciplina e a diminuição dos índices de evasão das turmas de introdução a computação da universidade em que o estudo foi adotado.

Para a execução do estudo, foi utilizada uma abordagem de 3 etapas com vários desafios utilizando a ferramenta Scratch com o objetivo de promover a participação de estudantes com diferentes níveis de conhecimento sobre lógica de programação. Assim, nas duas primeiras etapas dos desafios, foi disponibilizado um conjunto limitado e simplificado de instruções para se utilizar em suas resoluções, visando estabelecer o primeiro contato com a ferramenta e estimular o pensamento lógico. Nessa fase, os estudantes eram submetidos à tarefa de deslocar um personagem da posição inicial “start” até a posição indicada por uma marcação de “V” em verde sem obstáculos e, posteriormente, com obstáculos (Figura 3).

Figura 3 – Ilustração da atividade desenvolvida na etapa 1 com Scratch.



Fonte: Ribas *et al.* (2016).

A segunda etapa contava com a adição de comandos como a condicional e estrutura de repetição também simplificados e orientados de acordo com os desafios. A intenção era incrementar a dificuldade à medida que os desafios daquela etapa eram concluídos. Para isso, foram adicionados portais e paredes para incentivar a capacidade de se adaptar a novas situações utilizando os comandos recém aprendidos (Figura 4).

Figura 4 – Ilustração da atividade desenvolvida nas etapas 2 e 3 com Scratch.



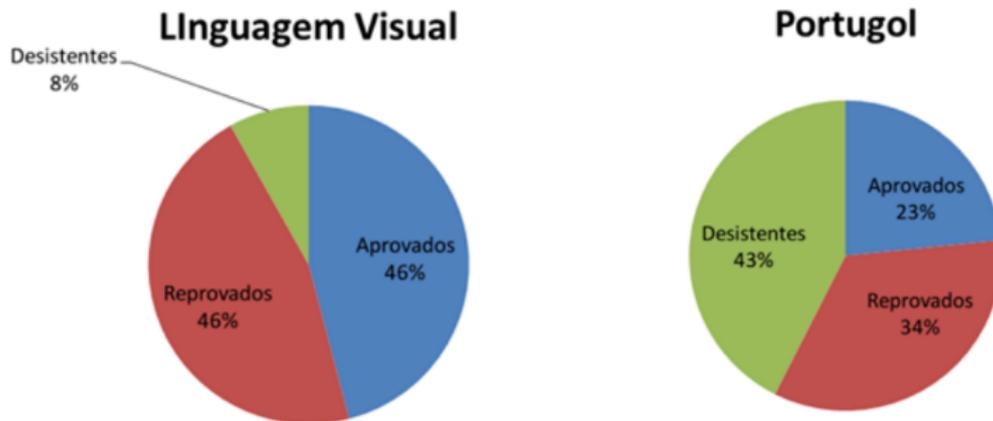
Fonte: Ribas *et al.* (2016).

Por fim, na terceira etapa, os alunos foram submetidos a uma exploração livre da ferramenta, podendo utilizar todos os comandos disponíveis no mural de comandos do Scratch para resolver os desafios propostos. O objetivo era identificar como os alunos resolveriam os problemas propostos utilizando os conhecimentos e ferramentas adquiridas e avaliar a lógica utilizada para chegar no objetivo.

Como resultados, foram analisados os dados da avaliação teórico/prática dos alunos de primeiro semestre participantes do estudo e comparados com a turma ingressante no semestre anterior. A comparação direta mostrou um aumento na média geral das notas da turma em relação a turma anterior. Também foi observada uma grande diferença nos índices de reprovação e desistência, caindo drasticamente entre a abordagem da turma que utilizou programação estruturada e a turma que utilizou Portugol como abordagem para o ensino de lógica de programação (Figura 5).

Portanto, o trabalho em questão elaborou um estudo prático de uma ferramenta de programação visual com a finalidade de estimular o pensamento lógico e a aprendizagem de conceitos fundamentais para cursos superiores na área da computação. Os resultados se mostra-

Figura 5 – Comparação dos resultados de diferentes abordagens de ensino entre dois semestres.



Fonte: Ribas *et al.* (2016).

ram promissores para um uso de *softwares* como o Scratch para o estímulo da aprendizagem de lógica de programação. Embora existam outros fatores a se considerar para uma afirmação concreta sobre as vantagens do uso da ferramenta, como o empenho e o nível de conhecimento já adquirido pelos estudantes desde a educação básica em matemática e lógica, ainda assim, são notórios os benefícios de tal abordagem no aprendizado dos alunos.

3.3.3 *O uso de softwares educativos para introdução de lógica de programação no ensino de base e superior (VIANA; PORTELA, 2019).*

O artigo apresenta, de forma geral, estudos sobre as dificuldades enfrentadas no ensino de lógica de programação no ensino superior e os desdobramentos que os sucedem, como os altos níveis de evasão e reprovação. Adicionalmente, apresenta uma revisão de ferramentas adotadas em pesquisas anteriores a fim de avaliar suas características e analisar seus impactos no ensino-aprendizagem de programação de acordo com cada nível de ensino.

Inúmeras pesquisas apontam as dificuldades ao se iniciar a jornada no mundo da programação, seja para compreender problemas complexos ou para aplicar conceitos abstratos. Como consequência, são notórios os índices de desistência logo no primeiro semestre das disciplinas de lógica de programação nos cursos superiores na área da Computação.

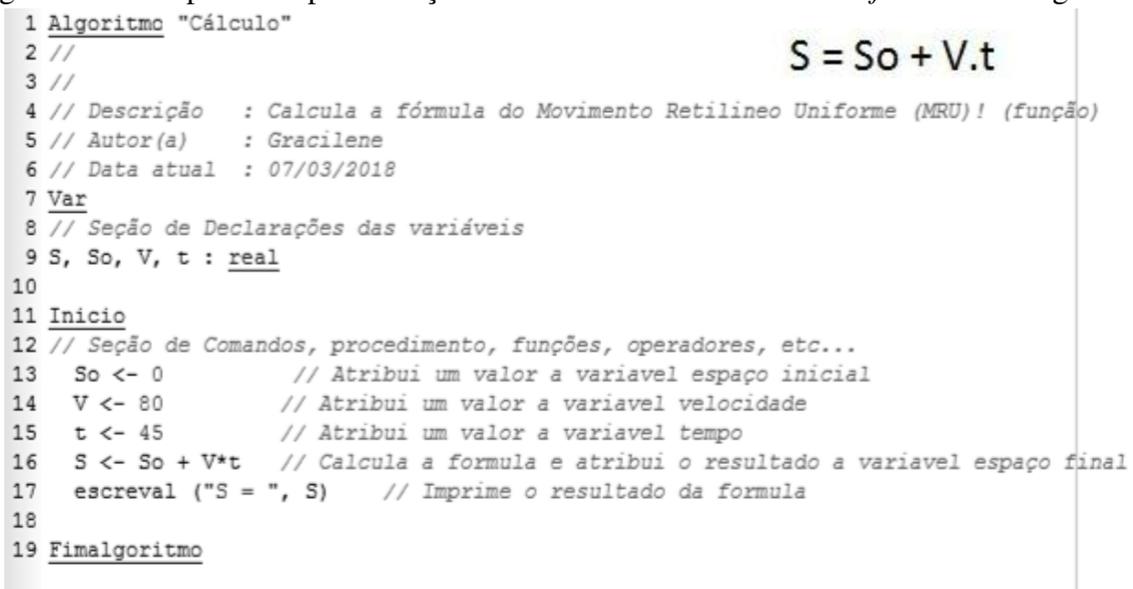
O problema vem sendo estudado e diversas estratégias têm-se tornado alvo de interesse dos pesquisadores, que utilizam metodologias ativas para ensinar a base da programação de forma mais simples e lúdica. O trabalho em questão aborda três ferramentas com esse propósito, utilizando elementos de gamificação e deixando mais compreensível certos elementos

fundamentais para o desenvolvimento do interesse do aluno na área.

Foi feita uma busca na literatura sobre *softwares* educativos nos anais de eventos e revistas nacionais na área de Ensino e Informática e foram identificados trabalhos utilizando as ferramentas Scratch, *VisuAlg* e *Robocode* que foram objeto de estudo para a pesquisa. Os três *softwares* passaram por avaliações de usabilidade utilizando os parâmetros da ISO/IEC 2003 e suas características foram classificadas de acordo com o nível de ensino adequado para cada um.

De acordo com os autores dessa pesquisa, o *software VisuAlg* teve o melhor desempenho geral e foi descrito como bastante intuitivo em seus ícones e menus, de fácil manuseio e aprendizado com destaque para a linguagem utilizada, sendo o “Portugol”, uma linguagem de programação com todos os elementos e palavras chaves em português e sua aplicação foi sugerida para estudantes do ensino médio e pré-universitários para aprender conceitos de raciocínio lógico e matemático (Figura 6).

Figura 6 – Exemplo de implementação de uma fórmula da física no *software VisuAlg*.



```

1 Algoritmo "Cálculo"
2 //
3 //
4 // Descrição : Calcula a fórmula do Movimento Retilíneo Uniforme (MRU)! (função)
5 // Autor(a) : Gracilene
6 // Data atual : 07/03/2018
7 Var
8 // Seção de Declarações das variáveis
9 S, So, V, t : real
10
11 Início
12 // Seção de Comandos, procedimento, funções, operadores, etc...
13 So <- 0 // Atribui um valor a variável espaço inicial
14 V <- 80 // Atribui um valor a variável velocidade
15 t <- 45 // Atribui um valor a variável tempo
16 S <- So + V*t // Calcula a fórmula e atribui o resultado a variável espaço final
17 escreval ("S = ", S) // Imprime o resultado da fórmula
18
19 Fimalgoritmo

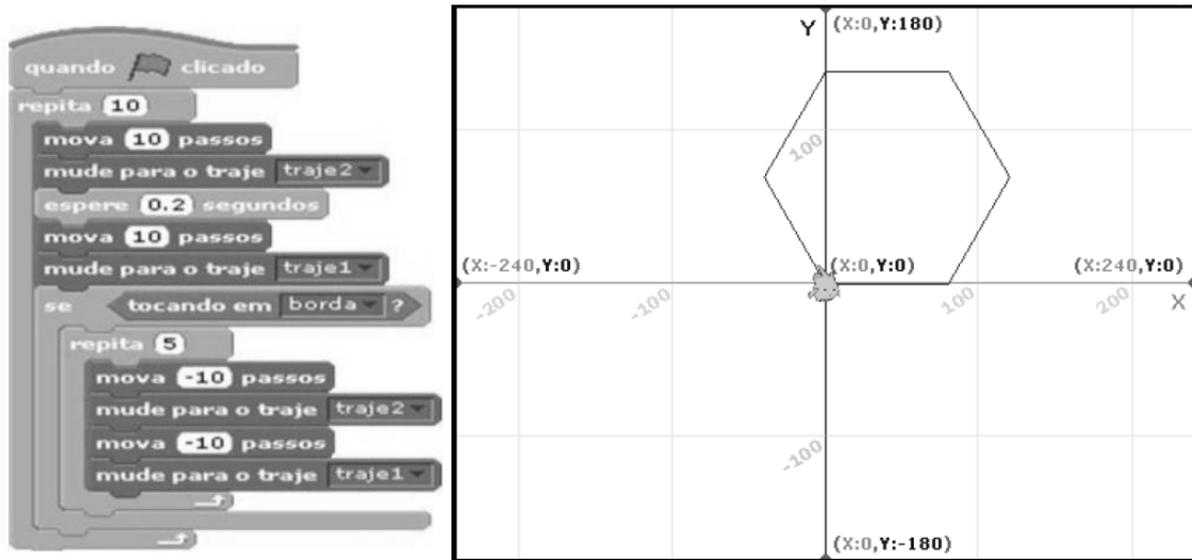
```

Fonte: Viana e Portela (2019).

A ferramenta Scratch vem em seguida, com uma abordagem mais dinâmica por utilizar programação estruturada em blocos, podendo ser criados desde desafios matemáticos, utilizando todos os conceitos de programação, até jogos interativos e cenários com personagens e narrativas (Figura 7). Os autores classificaram sua metodologia como apropriada para ensinar programação para alunos do ensino fundamental por sua forma mais descontraída e suas cores chamativas.

Por fim, foi analisada a ferramenta *Robocode* como método de gamificação para

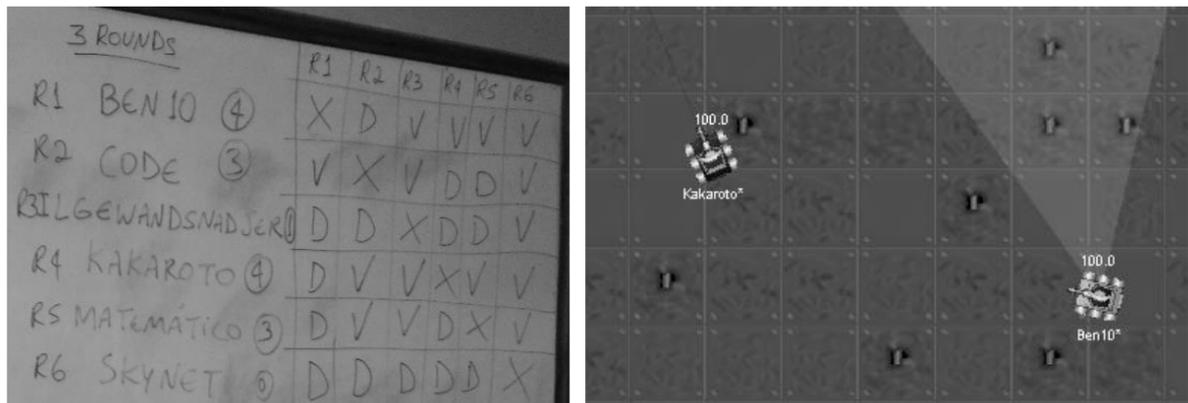
Figura 7 – Exemplo de abordagem matemática no *software* Scratch.



Fonte: Viana e Portela (2019).

ensinar a Programação Orientada a Objeto (POO) utilizando a linguagem Java (Figura 8). Ela foi classificada como adequada para o uso durante a graduação em cursos de computação, pois poderia auxiliar nos conceitos abstratos e complexos de POO e, também, por sua interface ser totalmente em inglês, o que dificultaria sua implementação em outros níveis disciplinares.

Figura 8 – Exemplo de aplicação de gamificação entre equipes utilizando o *software* Robocode.



Fonte: Viana e Portela (2019).

Para concluir, foram feitas observações sobre a aplicabilidade futura dessas ferramentas em estudos de casos para verificar com maior precisão os resultados das hipóteses levantadas no trabalho. Com isso, é de grande importância atentar para os resultados esperados e até que ponto tais ferramentas podem ser vantajosas em relação ao método de ensino de programação tradicional.

3.3.4 O ensino da programação e o desenvolvimento do pensamento lógico: uma revisão sistemática de literatura (SHIMASAKI; PRADO, 2021)

O artigo em questão busca fazer uma análise sobre pesquisas desenvolvidas no âmbito do ensino de lógica de programação utilizando como base metodológica a Revisão Sistemática da Literatura. O objetivo é ampliar os conhecimentos sobre o ensino da programação e o desenvolvimento do pensamento lógico, facilitando com que pesquisadores alcancem um referencial seguro e confiável sobre o objeto de estudo e possam então desenvolver e validar suas pesquisas.

O trabalho desenvolve uma série de etapas para selecionar, validar, analisar diversos artigos com a temática principal sendo o Pensamento Computacional. A primeira etapa busca a identificação da problemática principal, que consiste em analisar o uso do pensamento lógico e computacional para resolver problemas do cotidiano. Em seguida é delimitado que as pesquisas seriam realizadas somente no repositório do Simpósio Brasileiro de Informática na Educação (SBIE) e com filtros para os seis anos anteriores à data da pesquisa (2014 a 2019). Também foi decidido que só seriam analisados artigos em português e os resultados que derivam da busca pelas palavras chave “Lógica” e “Programação” no repositório.

Após a filtragem, foram encontrados vinte e dois artigos e selecionados dezoito que estavam em concordância com o objetivo da pesquisa. Os artigos selecionados foram lidos, debatidos e classificados de acordo com sua abordagem: voltados para o ensino infantil, ensino médio e ensino superior.

Nos trabalhos selecionados voltados para o ensino infantil, os pesquisadores constataram um grande enfoque numa abordagem de ensino mais lúdica, utilizando em sua maioria ferramentas de gamificação. Como resultado desses estudos, é notório os benefícios para a aprendizagem não só do raciocínio lógico como também do incentivo à criatividade nas crianças ao utilizar tais metodologias.

Já nos trabalhos filtrados como voltados ao ensino médio, pôde-se concluir que realizar tarefas que envolvem Lógica de Programação interfere positivamente nos alunos. Também, a utilização de jogos, como Scratch, na resolução de problemas contribui num melhor desempenho relacionado ao raciocínio lógico e matemático do corpo discente estudado.

E nas pesquisas designadas para o ensino superior, é evidente a atenção dada para a resolução das dificuldades que os alunos possuem relacionadas à aplicação do raciocínio lógico, principalmente em cursos voltados à computação. Para isso, diversos *softwares* de gamificação

como *KidCoder*, *Scratch* e *Lord of Code*; e ferramentas educacionais como *AdaptWeb* e Ambiente Virtual de Aprendizagem (AVA) foram estudadas com o intuito de facilitar a aprendizagem dos alunos e, ao mesmo tempo, incentivar a permanência nos cursos de graduação.

Por fim, foram apresentados os resultados da pesquisa. Conclui-se que as hipóteses levantadas inicialmente sobre o uso da Lógica Computacional para o desenvolvimento do raciocínio lógico dos estudantes podem ser observadas nos artigos analisados. Estes demonstram a importância do Pensamento Computacional e o seu potencial não somente para o desenvolvimento de novas ideias e novos estudos sobre o ensino da programação para os diferentes ambientes de aprendizado, mas também sua efetividade para despertar a curiosidade e o engajamento dos alunos, evitando ainda a evasão.

3.3.5 Instituto de Hackers: o Pensamento Computacional aplicado ao Ensino Médio Integrado Profissionalizante (RODRIGUES; SOUZA, 2021)

O trabalho aborda a temática do Pensamento Computacional voltado para alunos do ensino médio por meio de uma oficina utilizando ferramentas de programação visual para desenvolver essa habilidade. De início, buscou-se fazer um estudo para contextualizar o ensino médio profissional, suas características e suas relações com a abordagem proposta.

A prática pedagógica adotada foi fundamental para estabelecer os objetivos da oficina. Foi utilizada a proposta de ensino de Lave e Wenger (1991), que tem como base que as pessoas e o mundo social em que elas vivem e agem não podem ser separados. Assim, o contexto em que estão inseridos e as atividades cotidianas, como por exemplo atividades envolvendo problemáticas reais e com soluções colaborativas, podem refletir num melhor desempenho na aprendizagem. Com isso, novos conhecimentos e melhorias nas habilidades técnicas são formados a partir de conhecimentos prévios que foram adquiridos em situações vivenciadas anteriormente.

O método de pesquisa visou levantar a opinião de forma qualitativa e quantitativa dos alunos que participaram da oficina sobre sua experiência com a ferramenta Scratch, suas observações em relação ao aprendizado e percepção do Pensamento Computacional. As aulas foram aplicadas para alunos do 1.º, 2.º, 3.º e 4.º anos do ensino médio que tinham interesse em participar da oficina, sendo realizadas no contra turno para duas turmas em momentos distintos. Os assuntos abordados nas aulas abrangiam os tópicos de estrutura sequencial, decisão e repetição. As atividades eram correlacionadas com o conteúdo das disciplinas dos alunos do

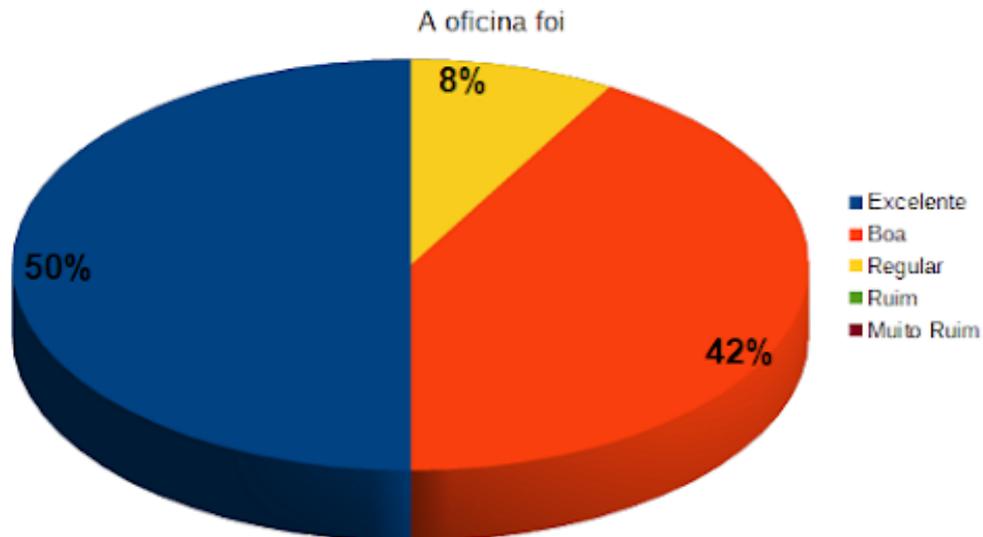
ensino médio.

Ao final da oficina, os alunos foram submetidos a responder um questionário com perguntas objetivas e perguntas abertas a fim de entender a percepção dos estudantes com relação à oficina. Com isso, foi elaborada uma análise sobre os dados de forma quantitativa e qualitativa utilizando estatística descritiva.

Os resultados da análise se mostraram bastante positivos. Por meio dos relatos descritos nas perguntas abertas, notou-se que os estudantes gostaram de participar da oficina, demonstraram interesse nas atividades propostas e relataram a facilidade para executá-las. Relataram também que gostaram da metodologia aplicada, da programação estruturada por blocos com Scratch, e demonstraram satisfação em relação a outras metodologias de ensino.

Sobre a avaliação quantitativa, foi possível perceber um grande nível de satisfação em relação à oficina e à maneira como foi desenvolvida ao responder a pergunta “A oficina foi:” referente à opinião sobre a oficina (Figura 9).

Figura 9 – Resultados da pesquisa de satisfação dos alunos em relação à oficina.

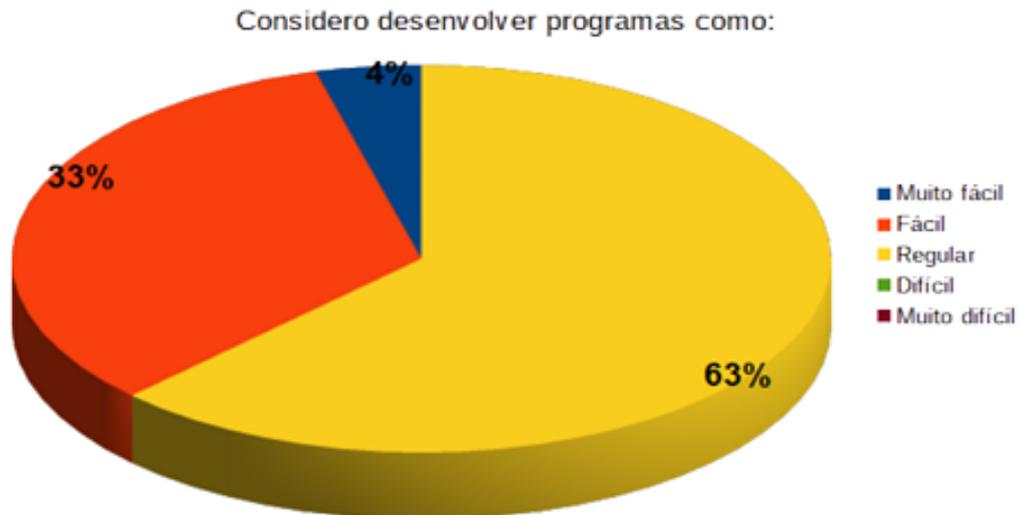


Fonte: Rodrigues e Souza (2021).

Também questionou-se sobre a ferramenta abordada e sobre sua facilidade de domínio e utilização. As respostas se concentraram majoritariamente em regular e fácil, evidenciando que a experiência da utilização do Scratch na oficina com relação à compreensão dos estudantes em conseguir produzir algum tipo de *software* foi positiva (Figura 10).

Além dos questionários, foram consideradas também as observações dos tutores durante a oficina. Percebeu-se que os estudantes foram muito participativos e se mantiveram focados nos objetivos das aulas. Durante as observações, notou-se um certo nível de dificuldade ao

Figura 10 – Resultados da pesquisa de facilidade de uso da ferramenta Scratch.



Fonte: Rodrigues e Souza (2021).

abordar uma atividade envolvendo estruturas de repetição causado por uma falta de compreensão dos padrões de repetição da atividade.

Apesar disso, a ferramenta Scratch demonstrou ser excelente para estimular a criatividade e autonomia dos alunos. Foi possível notar o desenvolvimento do espírito crítico ao serem instigados sobre a utilização da tecnologia nos dias atuais e também que aprender sobre ela é de grande importância para formação dos estudantes. Assim, percebeu-se que a abordagem educacional conseguiu promover o desenvolvimento do Pensamento Computacional entre os alunos de maneira lúdica e agradável. A metodologia apresentada foi bem recebida, de fácil compreensão, e contribuiu para facilitar o processo de aprendizagem.

3.3.6 *Análise comparativa*

Para melhor contextualizar o presente estudo, esta seção busca expor uma análise comparativa com as pesquisas mais relevantes da área, conforme apresentadas na revisão bibliográfica. A Tabela 1 sintetiza as principais características metodológicas de cada trabalho, permitindo uma avaliação mais aprofundada das contribuições originais desta pesquisa.

Tabela 1 – Comparação entre os trabalhos relacionados e o presente trabalho

Trabalhos	Aborda Scratch	Aborda Pensamento Computacional	Voltado para o Ensino Médio	Demonstra Desafios e Dinâmica	Analisa Questões Socioeconômicas
Mota <i>et al.</i> (2014)	X	X	X		
Ribas <i>et al.</i> (2016)	X	X		X	
Viana e Portela (2019)	X		X		
Shimasaki e Prado (2021)	X	X	X		
Rodrigues e Souza (2021)	X	X	X	X	
Presente trabalho	X	X	X	X	X

Fonte: Elaborada pelo autor (2024).

3.4 Considerações finais

Diante do contexto apresentado, a abordagem atual no ensino do raciocínio lógico pode ser aprimorada por meio da incorporação de novas ferramentas e metodologias, proporcionando um enriquecimento das aulas e promovendo um maior envolvimento dos alunos na compreensão do Pensamento Computacional. Dessa forma, os estudos e as pesquisas abordados neste capítulo revelam-se de extrema relevância para o progresso na área da educação e da inovação tecnológica. Suas contribuições não apenas evidenciam o potencial a ser explorado na ferramenta Scratch, mas também ressaltam que a sua aplicação em sala de aula representa um passo significativo para o desenvolvimento do Pensamento Computacional, tanto entre os alunos quanto entre os professores, desempenhando um papel crucial em todo o contexto escolar.

4 METODOLOGIA

4.1 Introdução

De acordo com Prodanov e Freitas (2013), a pesquisa exploratória serve para aprofundar o conhecimento sobre um tema a ser investigado, facilitando a definição do escopo da pesquisa, o estabelecimento de objetivos claros e a criação de hipóteses para resolver a problemática abordada no estudo.

O foco deste trabalho é realizar uma pesquisa exploratória com o propósito de compreender a experiência em computação dos alunos do ensino médio e analisar a percepção de aprendizagem sob o ponto de vista dos próprios alunos. Para alcançar esse objetivo, foram planejadas sete aulas teórico-práticas, nas quais os alunos do primeiro ano de uma turma de Desenvolvimento de Sistemas de uma escola de ensino médio profissionalizante do município de Russas, no estado do Ceará, foram introduzidos ao Pensamento Computacional e aos fundamentos da computação utilizando o *software* Scratch.

Na condução desta pesquisa, optou-se por métodos que permitissem a sumarização das principais questões relacionadas aos objetivos propostos, abordando tanto aspectos quantitativos quanto relatos dos participantes para a avaliação. Nesse contexto, adotou-se o modelo de avaliação DETECT (WANGENHEIM *et al.*, 2017) como base para a elaboração de questionários de feedback dos alunos sobre as aulas ministradas.

Assim, este capítulo tem como objetivo fornecer uma compreensão clara e detalhada do processo metodológico adotado, garantindo a transparência e a reprodutibilidade do estudo.

4.2 Planejamento das Aulas

4.2.1 Questões éticas

Durante a realização das aulas com alunos do ensino médio, foram consideradas questões éticas fundamentais para garantir a integridade e o respeito aos participantes. Para isso, foram adotados termos de responsabilidade ética, incluindo termos de consentimento e termos de assentimento fornecidos pelo projeto LearningLab que podem ser acessados nos anexos A e B.

Antes do início das aulas, os alunos foram devidamente informados sobre os objetivos da pesquisa e os procedimentos envolvidos. Os termos de consentimento foram disponibilizados e explicados detalhadamente, permitindo que os alunos e seus responsáveis tomassem uma

decisão informada sobre sua participação nas atividades. Além disso, foi fornecido um termo de assentimento para os alunos menores de idade, que garantia que sua participação era voluntária e que eles estavam cientes dos objetivos e dos procedimentos da pesquisa.

A coleta de dados dos alunos foi realizada com o devido consentimento e assentimento, garantindo assim a legitimidade e a ética da pesquisa. Essas medidas éticas foram essenciais para proteger os direitos e o bem-estar dos participantes, bem como para garantir a credibilidade e a integridade dos resultados obtidos.

Para efeitos de comparação entre os questionários aplicados no início e no final do curso, foram coletados os nomes dos alunos. Contudo, para assegurar a privacidade, esses dados serão apresentados neste trabalho de maneira anônima, adotando o padrão de "Aluno 1, Aluno 2, etc.", em uma ordem diferente da ordem alfabética dos nomes dos alunos. Além disso, todas as fotos e imagens das aulas e atividades que forem incluídas no trabalho terão os rostos dos alunos borrados, garantindo a proteção da identidade dos participantes.

4.2.2 Ementa

A ementa abrange uma introdução ao curso, incluindo a contextualização do tema e seus objetivos, formas de avaliação, atividades a serem realizadas em sala, e uma conversa sobre a relevância do conteúdo para a graduação e o mercado de trabalho. Os tópicos iniciais cobrem a definição do Pensamento Computacional e seus quatro pilares, além de uma introdução à ferramenta Scratch e à programação estruturada por blocos.

Posteriormente é abordada a criação de programas simples com foco em programação sequencial, comandos de entrada e saída, variáveis e expressões aritméticas e relacionais. As aulas também apresentam estruturas de decisão, incluindo comandos de decisão simples e aninhados, e estruturas de repetição, com diversos tipos de laços. Além disso, são exploradas as funções e a criação de programas utilizando funções.

O curso inclui atividades práticas como a criação de programas para soma de números, conversão de temperatura, verificação de números pares ou ímpares, e uma calculadora simples, entre outros. A avaliação dos projetos é realizada através de uma atividade final de aprendizado, seguida pela conclusão do curso com uma recapitulação dos conceitos aprendidos, aplicação de questionários de feedback, agradecimentos e encerramento.

4.2.3 Cronograma

A Tabela 2 mostra o cronograma com o conteúdo e o tipo de cada aula.

Tabela 2 – Cronograma das aulas

Aula	Descrição do conteúdo	Tipo de aula
Aula 1	<p>Tema: Introdução</p> <p>Objetivos específicos:</p> <ul style="list-style-type: none"> – Contextualizar a razão do curso e os objetivos a serem alcançados para a pesquisa e para os alunos; – Descrever as formas de avaliação; – Descrever as atividades a serem realizadas em sala; – Contextualizar o conteúdo do curso com a graduação e com mercado de trabalho; – Definir Pensamento Computacional; – Descrever os 4 pilares do Pensamento Computacional; – Estabelecer o primeiro contato ferramenta Scratch; – Demonstrar cada componente da ferramenta Scratch; – Descrever como é utilizada a programação estruturada por blocos; – Realizar atividade de iniciação do experimento (questionário inicial). 	Teórica

Continua na próxima página

Tabela 2 – Cronograma das aulas (continuação)

Aula	Descrição do conteúdo	Tipo de aula
Aula 2	<p>Tema: Primeiro programa e expressões</p> <p>Objetivos específicos:</p> <ul style="list-style-type: none"> – Descrever programação sequencial; – Descrever comandos de entrada e saída; – Criar programa com comandos de entrada e saída; – Descrever variáveis; – Criar programa com variáveis; – Utilizar variáveis para salvar dados de entrada; – Utilizar variáveis para mostrar dados de saída. – Descrever expressões aritméticas; – Descrever expressões relacionais; – Descrever expressões booleanas; – Utilizar expressões aritméticas e relacionais em conjunto com conhecimentos adquiridos da aula anterior; – Atividade 1: Criar programa - Soma de dois números; – Atividade extra: Criar programa - Conversão de temperatura. 	Teórica e Prática
Aula 3	<p>Tema: Estruturas de decisão</p> <p>Objetivos específicos:</p> <ul style="list-style-type: none"> – Definir estruturas de decisão; – Descrever teste de decisão; – Descrever o comando se ... então; – Descrever o comando se ... então ... senão; – Definir comando de decisão aninhado; – Criar programa com comandos de decisão; – Atividade 2: Criar programa - Número par ou ímpar. 	Teórica e Prática

Continua na próxima página

Tabela 2 – Cronograma das aulas (continuação)

Aula	Descrição do conteúdo	Tipo de aula
Aula 4	<p>Tema: Estruturas de repetição</p> <p>Objetivos específicos:</p> <ul style="list-style-type: none"> – Definir estruturas de repetição; – Descrever o comando repita ... vezes; – Descrever o comando repita até que ...; – Descrever o comando sempre (repetição em laço infinito); – Definir comando de repetição aninhado; – Criar programa com comandos de repetição; – Atividade 3: Criar programa - Urso caminhando. 	Teórica e Prática
Aula 5	<p>Tema: Funções</p> <p>Objetivos específicos:</p> <ul style="list-style-type: none"> – Definir funções; – Demonstrar entrada de dados para funções; – Criar programa com funções; – Atividade 4: Criar função - Calculadora de IMC; – Atividade extra: Criar programa - Cálculo do delta de uma equação do segundo grau. – Atividade extra: Criar programa - Determinação das raízes de uma equação do segundo grau. 	Teórica e Prática
Aula 6	<p>Tema: Avaliação dos Projetos</p> <p>Objetivos específicos:</p> <ul style="list-style-type: none"> – Apresentar atividade final de aprendizado: desenvolvimento de programas próprios utilizando conhecimentos das aulas e apresentação do seu funcionamento e código; – Avaliar atividade final de aprendizado. 	Prática

Continua na próxima página

Tabela 2 – Cronograma das aulas (continuação)

Aula	Descrição do conteúdo	Tipo de aula
Aula 7	<p>Tema: Avaliação dos Projetos (continuação)</p> <p>Objetivos específicos:</p> <ul style="list-style-type: none"> – Apresentar atividade final de aprendizado: desenvolvimento de programas próprios utilizando conhecimentos das aulas e apresentação do seu funcionamento e código; – Avaliar atividade final de aprendizado; – Cerimônia de premiação e fotos; – Realizar atividade de conclusão do experimento (questionário de <i>feedback</i>); – Agradecimentos; – Encerramento. 	Prática

Fonte: Elaborada pelo autor (2024).

4.2.4 *Material utilizado*

Durante a atuação do autor como tutor no projeto Progame_CE, foi adquirida uma compreensão abrangente dos elementos didáticos e das abordagens que podem ser utilizados em conjunto com a ferramenta Scratch para desenvolver o Pensamento Computacional dos alunos do ensino médio. Esse conhecimento, desenvolvido ao longo do projeto, motivou a decisão de utilizar os materiais já elaborados e validados pelos professores envolvidos no projeto para a pesquisa apresentada neste trabalho. Assim, os materiais empregados foram baseados nos recursos disponibilizados pelo site da SEDUC do Governo do Estado do Ceará, por meio do projeto Progame_CE.

Dada a amplitude do conteúdo abordado pelo projeto, optou-se por utilizar apenas o Capítulo 1 do Módulo 2, que trata dos Fundamentos em Programação Visual (CAVALCANTE NETO *et al.*, 2021). Esse capítulo utiliza o Scratch como ambiente de programação e apresenta as interfaces do programa, conceitos básicos de programação e algoritmos, operadores, estruturas de decisão e repetição, e funções.

A disposição deste material em aulas foi planejada com base na ementa da disciplina de Fundamentos de Programação (UFC) do curso de Ciência da Computação da UFC. Foram elaboradas 5 aulas em formato de slides (Apêndice A), abrangendo uma exposição detalhada da

metodologia e dos objetivos das aulas, uma introdução aos conceitos de Pensamento Computacional, além dos conteúdos teóricos e práticos utilizando o material relacionado à ferramenta Scratch. Por fim, foram propostos desafios para cada um dos conteúdos abordados em aula.

4.2.5 Metodologia das aulas

A metodologia adotada para as aulas de Pensamento Computacional e Scratch combinou abordagens teóricas e práticas para maximizar a compreensão dos alunos. Cada assunto abordado durante as aulas foi dividido em duas partes: uma teórica e outra prática. A parte teórica consistiu em explicações sobre os conceitos fundamentais da programação em Scratch ao mesmo tempo que se estabelecia paralelos com o mundo da programação convencional. Já a parte prática envolveu atividades onde os alunos aplicavam esses conceitos, desenvolvendo pequenos projetos no Scratch.

Ao final de cada aula, eram propostos desafios. Esses desafios tinham complexidade variável, ajustada de acordo com o progresso da turma, e tinham como objetivos consolidar o aprendizado e estimular o raciocínio lógico. Essa abordagem permitiu uma avaliação contínua e forneceu feedback imediato aos alunos.

Para as duas últimas aulas, foi proposta uma dinâmica mais abrangente. Os alunos deveriam utilizar todos os conteúdos aprendidos ao longo das aulas para criar seus próprios programas. Durante essas aulas finais, os alunos deveriam apresentar seus projetos para a turma, explicando o motivo da escolha do projeto, como o desenvolveram, demonstrando o programa em funcionamento e destacando a sua finalidade.

Como estímulo, foram oferecidas premiações para as três equipes que obtivessem os melhores desempenhos na dinâmica. A avaliação foi feita pelo autor deste trabalho, pelo professor da escola e pelos monitores do LearningLab, e media tanto a qualidade técnica quanto a criatividade e a aplicação prática. Além disso, todos os participantes receberam um kit de participação desenvolvido pelo LearningLab e um certificado ao final, reconhecendo o esforço e a participação ao longo do curso. Esse método não apenas incentivou a participação ativa dos alunos, mas também proporcionou uma experiência prática valiosa, consolidando o aprendizado de maneira significativa.

4.2.6 Proposta de avaliação

A proposta de avaliação para esta etapa do trabalho incluiu dois tipos de avaliação. O primeiro tipo foi destinado a avaliar o aprendizado dos alunos, utilizando atividades práticas e uma dinâmica avaliativa, conforme descrito na Seção 4.2.6.1. O segundo tipo foi voltado para a avaliação do experimento, por meio de um questionário inicial e um questionário de feedback, conforme descrito na Seção 4.2.6.2.

4.2.6.1 Avaliação de aprendizado

Para avaliar a aprendizagem dos alunos e o alcançar os objetivos propostos, duas modalidades de atividades foram utilizadas: desafios de aprendizado e dinâmica final.

Os desafios de aprendizado deveriam ser realizadas ao final da explicação de cada novo assunto, garantindo que os conteúdos estivessem frescos na mente dos alunos, facilitando a assimilação e aplicação prática dos conceitos.

Cada desafio era cuidadosamente elaborado para contextualizar o uso do novo tópico em situações reais, interessantes e, sempre que possível, lúdicas, como por exemplo, no desafio de fazer um personagem do Scratch andar de um ponto a outro da tela utilizando comandos de repetição. Essa atividade é um problema comum em jogos e animações, tornando a atividade mais engajadora e significativa para os alunos.

A resolução dos desafios poderia ser realizada em duplas ou trios, promovendo a colaboração, o compartilhamento de conhecimentos e o desenvolvimento de habilidades interpessoais. Os alunos trabalhavam juntos para encontrar soluções criativas e eficientes para os problemas propostos.

O tutor, juntamente com os monitores, deveriam estar presentes durante a realização dos desafios para auxiliar os alunos, não somente esclarecendo dúvidas, fornecendo dicas e estimulando o raciocínio lógico mas também, orientando os alunos na aplicação de conceitos de Pensamento Computacional, como o pensamento algorítmico e a decomposição de problemas, ferramentas essenciais para a resolução de desafios complexos.

Ao final de cada desafio, as soluções elaboradas pelos alunos deveriam ser apresentadas, contabilizadas e discutidas em conjunto, destacando as diferentes perspectivas. Em seguida, a solução elaborada pelo tutor deveria ser apresentada, enfatizando os pontos-chave do desafio e, assim, concluindo a atividade.

Já a dinâmica final, realizada nas duas últimas aulas do experimento, consistiu na criação de um programa em Scratch, utilizando os conteúdos aprendidos durante as aulas. Os alunos foram divididos em equipes de dois ou três integrantes, de acordo com sua própria escolha, e tiveram a liberdade de desenvolver um jogo, uma ferramenta ou uma história utilizando a linguagem de programação Scratch.

A avaliação dos programas levou em consideração os seguintes critérios:

- **Abrangência dos conteúdos:** O programa deveria utilizar o máximo de conteúdos vistos em sala de aula, demonstrando o aprendizado e a assimilação dos conceitos.
- **Objetivo claro:** O programa deveria ter um objetivo definido e compreensível, demonstrando a capacidade dos alunos de formular problemas e soluções.
- **Resolução de problemática:** O programa deveria resolver uma problemática específica, evidenciando a aplicação prática do Pensamento Computacional na criação de soluções inovadoras.
- **Originalidade:** O programa deveria ser uma criação original dos alunos, não permitindo cópias da internet, apenas implementações próprias.

Para auxiliar na avaliação dos programas e garantir a imparcialidade, foram convidados quatro integrantes do projeto LearningLab para atuarem como monitores. Cada monitor atribuiu uma nota ao programa, considerando a criatividade, o esforço demonstrado e o atendimento aos critérios mencionados anteriormente.

Assim, para a aprovação no curso, o aluno deve apresentar nota final com valor igual ou superior a 06 (seis), conforme as regras da escola. A nota final foi calculada como a média aritmética a seguir:

$$NF = (CT + IC)/2$$

onde:

- CT - Avaliação de Competências Técnica (se usou os conteúdos aprendidos nas aulas)
- IC - Avaliação de Inovação e Criatividade (média das notas dos monitores)

Ao final das apresentações e da avaliação, as três melhores equipes foram premiadas. Além disso, todos os participantes receberam um kit de participação como forma de reconhecimento pelo esforço e dedicação durante o experimento. A descrição completa das etapas e critérios da dinâmica pode ser encontrada no Apêndice B.

A dinâmica final proporcionou uma grande oportunidade para avaliar o aprendizado dos alunos de forma abrangente e autêntica. Através da criação de seus próprios programas, os

alunos puderam colocar em prática os conhecimentos adquiridos, demonstrando sua criatividade, capacidade de resolução de problemas e domínio da linguagem Scratch.

A avaliação dos programas por monitores externos contribuiu para a confiabilidade dos resultados, garantindo uma análise justa e imparcial. A premiação dos melhores trabalhos serviu como incentivo para os alunos e valorizou o esforço individual e em equipe.

Portanto, ao participarem das duas modalidades de avaliação, os alunos consolidavam os conhecimentos desenvolvidos ao longo das aulas e garantiam sua nota de participação necessária para sua aprovação, ganhando a certificação ao final do curso.

Assim, os resultados prévios das avaliações indicam que o experimento cumpriu seus objetivos, promovendo o aprendizado significativo dos alunos em programação e Pensamento Computacional. Além disso, destacou as dificuldades dos alunos com certos conteúdos, fornecendo *insights* sobre áreas que podem ser melhor aprofundadas em trabalhos futuros.

4.2.6.2 Avaliação do experimento

A coleta de dados para a avaliação do experimento foi realizada por meio de dois questionários online, disponibilizados aos alunos via Google Forms. O primeiro questionário, de caráter socioeconômico e pré-curso, teve como objetivo traçar um perfil inicial dos participantes, incluindo informações sobre idade, escolaridade, acesso à internet e familiaridade com ferramentas tecnológicas. O segundo questionário, aplicado ao final do curso, buscou coletar o feedback dos alunos sobre a experiência de aprendizagem, com questões abertas e fechadas que permitiram avaliar a eficácia das atividades propostas, a percepção dos alunos sobre o conteúdo abordado e a identificação de pontos que poderiam ser aprimorados em futuras intervenções. A utilização de ambos os questionários, com diferentes objetivos e formatos de perguntas, proporcionou uma análise mais completa e aprofundada dos dados, caracterizando a pesquisa como um estudo de natureza mista.

4.2.6.2.1 Questionário inicial

Para traçar um perfil abrangente dos participantes e embasar a metodologia das aulas, foi elaborado um questionário inicial em formato de formulário. Este questionário foi aplicado ao final da primeira aula e coletou dados socioeconômicos, demográficos e informações sobre as experiências prévias dos alunos com programação. As perguntas que compunham o questionário estão detalhadas na Tabela 3.

As informações obtidas pelo questionário permitiram um conhecimento mais aprofundado do perfil dos alunos, incluindo suas características socioeconômicas, familiaridade com tecnologia e vivências anteriores com programação. Com base nesse conhecimento, foi possível adaptar as aulas de forma personalizada, atendendo às necessidades e níveis de conhecimento de cada participante. Essa personalização otimizou o processo de ensino e aprendizagem, proporcionando um ambiente mais propício para o desenvolvimento individual de cada aluno.

Tabela 3 – Questionário inicial

No.	Descrição do item	Formato das respostas
1	Nome completo	Texto
2	Sexo	(1) Feminino (2) Masculino (3) Prefiro não informar (4) Outros...
3	Idade	(1) 14 (2) 15 (3) 16 (4) 17 (5) 18 (6) Outros...
4	Qual a renda mensal da sua família?	(1) Até 01 salário mínimo (2) Mais que 01 e até 02 salários mínimos (3) Mais que 02 e até 05 salários mínimos (4) Mais que 05 e até 08 salários mínimos (5) Mais que 08 salários mínimos

Continua na próxima página

Tabela 3 – Questionário inicial (continuação)

No.	Descrição do item	Formato das respostas
5	Qual o nível de escolaridade do seu pai?	(1) Da 1ª à 4ª série do Ensino Fundamental (antigo primário). (2) Da 5ª à 8ª série do Ensino Fundamental (antigo ginásio). (3) Ensino Médio (antigo 2º grau). (4) Ensino Superior. (5) Pós-graduação (Especialização, Mestrado ou Doutorado). (6) Não estudou. (7) Não sei.
6	Qual o nível de escolaridade da sua mãe?	(1) Da 1ª à 4ª série do Ensino Fundamental (antigo primário). (2) Da 5ª à 8ª série do Ensino Fundamental (antigo ginásio). (3) Ensino Médio (antigo 2º grau). (4) Ensino Superior. (5) Pós-graduação (Especialização, Mestrado ou Doutorado). (6) Não estudou. (7) Não sei.
7	Reside em Russas?	(1) Sim (2) Não
8	Qual principal transporte utiliza para chegar à escola?	(1) A pé. (2) Bicicleta. (3) Ônibus pago. (4) Ônibus escolar. (5) De carona com pais ou amigos (carro/-moto) (6) Outros...

Continua na próxima página

Tabela 3 – Questionário inicial (continuação)

No.	Descrição do item	Formato das respostas
9	Quanto tempo em média demora para chegar à escola?	Texto
10	Você já reprovou alguma vez?	(1) Sim (2) Não
11	Se respondeu sim para a pergunta anterior, quantas vezes você reprovou?	Texto
12	Você gosta de matemática?	(1) Sim (2) Não
13	Tem acesso a computador em casa?	(1) Sim (2) Não
14	Tem acesso a internet em casa?	(1) Sim (2) Não
15	Qual seu nível de conhecimento em informática básica?	(1) Nenhum - não tenho contato com o computador; (2) Mínimo - uso raramente; (3) Básico - tenho conhecimentos gerais, mas quase nunca uso; (4) Intermediário - tenho conhecimento médio e já sei usar sem dificuldades; (5) Avançado - tenho um conhecimento profundo e costumo usar diariamente.
16	Justifique a resposta anterior.	Texto
17	Você já teve algum contato com programação?	(1) Sim (2) Não
18	Se respondeu sim para a pergunta anterior, qual tipo de contato, conte sobre sua experiência.	Texto
19	Já teve contato com programação visual?	(1) Sim (2) Não

Continua na próxima página

Tabela 3 – Questionário inicial (continuação)

No.	Descrição do item	Formato das respostas
20	Se respondeu sim para a pergunta anterior, qual tipo de contato, conte sobre sua experiência.	Texto
21	O que você espera aprender nesse curso?	Texto

Fonte: Elaborada pelo autor (2024).

4.2.6.2.2 Questionário final

O questionário final foi aplicado ao final da última aula do curso e buscava coletar dados envolvendo a experiência dos alunos com relação ao curso, utilizando primeiramente o modelo de avaliação dTECT com perguntas objetivas e, em seguida, perguntas abertas sobre suas percepções positivas e negativas em relação às aulas desenvolvidas durante o curso. As perguntas elaboradas para este questionário estão descritas na Tabela 4.

Tabela 4 – Questionário final

No.	Descrição do item	Formato das respostas
1	Nome completo	Texto
2	A oficina foi	(1) Muito Divertida (2) Divertida (3) Chata (4) Muito chata
3	O tempo das aulas passou	(1) Muito rápido (2) Rápido (3) Devagar (4) Muito devagar
4	A oficina foi	(1) Excelente (2) Boa (3) Regular (4) Ruim

Continua na próxima página

Tabela 4 – Questionário final (continuação)

No.	Descrição do item	Formato das respostas
5	Mostrei ou vou mostrar meu programa de computador para outras pessoas	(1) Sim (2) Não
6	Quero aprender mais sobre como fazer programas de computador	(1) Sim (2) Não
7	Fazer um programa de computador é	(1) Muito Divertido (2) Divertido (3) Chato (4) Muito chato
8	Eu gosto de fazer programas de computador	(1) Sim (2) Não
9	A computação é útil no dia a dia	(1) Sim (2) Não
10	Eu penso em trabalhar com computação	(1) Sim (2) Não
11	A oficina foi	(1) Muito fácil (2) Fácil (3) Difícil (4) Muito Difícil
12	Eu consigo fazer programas de computador	(1) Sim (2) Não
13	Eu consigo explicar para um amigo/amiga como fazer um programa de computador	(1) Sim (2) Não
14	Fazer um programa de computador é	(1) Muito fácil (2) Fácil (3) Difícil (4) Muito Difícil

Continua na próxima página

Tabela 4 – Questionário final (continuação)

No.	Descrição do item	Formato das respostas
15	Qual a porcentagem de tempo estimada você acha que usou o computador durante o curso?	(1) 20% do tempo ou menos (2) 40% a 60% do tempo (3) 60% a 80% do tempo (4) 80% do tempo ou mais
16	Diga com as suas palavras, o que você aprendeu durante as aulas?	Texto
17	O que você achou sobre Pensamento Computacional? Como isso pode ajudar em outras disciplinas?	Texto
18	Houve dificuldades com a manipulação do Scratch? Quais dificuldades você sentiu?	Texto
19	Houve dificuldades em algum conteúdo específico ou em alguma aula? Quais dificuldades você sentiu?	Texto
20	O que mais gostou de trabalhar durante o curso?	Texto
21	O que menos gostou de trabalhar durante o curso?	Texto
22	O que achou da metodologia das aulas? O que melhoraria?	Texto
23	Você tem interesse em seguir nessa área de programação?	(1) Não (2) Sim, já pretendia seguir (3) Sim, mas surgiu interesse durante o curso (4) Ainda não sei

Fonte: Elaborada pelo autor (2024).

Dessa forma, as respostas obtidas nesta etapa serão analisadas detalhadamente no Capítulo 5, onde serão exploradas as questões de pesquisa e será discutido como as perguntas do formulário auxiliaram na investigação das questões abordadas neste trabalho.

4.3 Aplicação das Aulas

4.3.1 Perfil da escola

A instituição escolhida para a realização das aulas foi a Escola Estadual de Educação Profissional Professor Walquer Cavalcante Maia, localizada no município de Russas, no interior do Ceará. Esta escola atende estudantes das localidades próximas e funciona em tempo integral. Além das disciplinas curriculares regulares, a escola oferece cursos extracurriculares voltados para áreas específicas, preparando os alunos para a atuação no mercado de trabalho.

No ano de 2024, a escola oferece turmas em Agronegócio, Contabilidade, Desenvolvimento de Sistemas e Enfermagem, como parte de seus cursos extracurriculares. Para a execução das aulas de Pensamento Computacional e Scratch, a estrutura utilizada incluiu um laboratório de informática equipado com 25 computadores. Além disso, foi disponibilizado um projetor para a ministração das aulas, facilitando a apresentação dos conteúdos teóricos e práticos.

4.3.2 Perfil do aluno

Para a aplicação das aulas, foi selecionada a turma de Desenvolvimento de Sistemas da E.E.E.P. Professor Walquer Cavalcante Maia. Esta turma é composta por 45 alunos com interesse e aptidão na área de tecnologia da informação, demonstrando uma forte motivação para aprender e aplicar conceitos de programação.

A escolha da turma de Desenvolvimento de Sistemas visou maximizar o engajamento e o aproveitamento das atividades propostas, assegurando uma experiência significativa e relevante para os participantes.

Os alunos dessa turma possuem um perfil diversificado, com diferentes níveis de experiência prévia em programação. Alguns têm conhecimentos básicos, enquanto outros têm maior familiaridade com linguagens de programação e desenvolvimento de *software*. Essa variedade de conhecimentos e habilidades dentro da turma proporcionou um ambiente rico para a troca de experiências e para o aprendizado colaborativo.

4.3.3 *Descrição das Aulas*

As aulas tiveram início em 10 de Abril e foram concluídas em 9 de Maio de 2024, sendo realizadas nas quartas e quintas durante o período da manhã, das 9:30 às 11:10 no laboratório de informática da própria escola. O laboratório possuía 25 computadores para 45 alunos, portanto, as atividades práticas foram realizadas em duplas. Todos os horários e datas foram previamente acordados entre o tutor, o autor deste trabalho, o orientador de TCC, o projeto LearningLab e o professor da turma do primeiro ano de Sistemas Distribuídos da escola.

Todas as aulas contaram com a presença do tutor, que foi o principal responsável pela apresentação das aulas, além dos monitores para auxiliar durante as apresentações e atividades, e o professor responsável pela turma como representante da escola. A seguir, detalha-se o que ocorreu em cada semana durante os dias designados para o curso:

Quarta, 10 de Abril de 2024

Na primeira aula, o foco principal foi a interação entre o autor deste trabalho e a turma, visando estabelecer um primeiro contato com os alunos, discutir suas expectativas em relação ao curso, avaliar o nível de conhecimento dos estudantes e desenvolver a base teórica sobre Pensamento Computacional, de modo que os objetivos do curso e do experimento fossem compreendidos pelos alunos.

Essa aula inaugural foi dividida em dois momentos distintos. O primeiro momento ocorreu dentro da sala de aula, onde foi utilizada uma televisão para a apresentação do conteúdo teórico acerca do Pensamento Computacional, discutindo sobre os seus quatro pilares (decomposição, reconhecimento de padrões, abstração e algoritmos) e sobre sua relevância no desenvolvimento de uma base de conhecimento necessária para a resolução de problemas.

O segundo momento se deu no laboratório de informática da própria escola, onde os alunos tiveram seu primeiro contato com a ferramenta Scratch através da aplicação web disponível no site do Scratch (2023b). Neste momento, os alunos receberam orientações sobre os diferentes campos do Scratch, incluindo sua área de blocos, área de montagem do programa, área de visualização, opções de escolha de personagem e cenário. Foi possível notar a empolgação e a curiosidade dos alunos para explorar e tentar criar programas, além de seus questionamentos acerca das possibilidades que o Scratch oferece, como criar jogos e programar interações específicas, como fazer um personagem caminhar e falar.

Ao final da aula, foi solicitado que os alunos preenchessem um questionário inicial, conforme descrito na Seção 4.2.6.2.1. Esse questionário ajudaria a traçar o perfil dos alunos,

permitindo compreender seus níveis de familiaridade com a tecnologia e com a programação.

Quinta, 11 de Abril de 2024

A partir da segunda aula, o conteúdo foi ministrado integralmente no laboratório, e a metodologia adotada consistiu na divisão dos conteúdos em teóricos e práticos para cada assunto abordado durante as aulas. Os alunos, agora familiarizados com os componentes do Scratch e os conceitos de Pensamento Computacional, foram introduzidos às aulas específicas com a abordagem dos fundamentos de programação utilizando a ferramenta.

O ponto de partida foi compreender a programação sequencial, que os alunos relacionaram com o conceito de Algoritmo discutido na aula anterior sobre Pensamento Computacional. Em seguida, foram apresentados os conceitos envolvendo os comandos de entrada e saída e sua implementação no Scratch. O desafio relacionado aos comandos de entrada e saída foi bastante intuitivo, e os alunos entenderam facilmente a proposta sugerida, que consistia em fazer o gatinho dar as boas-vindas aos alunos no Scratch – o famoso "hello world". O desafio incluía o gatinho perguntar o nome do aluno, o aluno responder, e o gatinho responder com a saudação "É um prazer te conhecer". Os alunos sugeriram que seria interessante que o gatinho falasse o nome do aluno juntamente com a saudação, e com esse objetivo, iniciou-se a discussão teórica sobre variáveis. O desafio relacionado à manipulação de variáveis foi exatamente o sugerido pelos alunos; agora, o programa já desenvolvido precisava ser melhorado, incorporando o nome dos alunos fornecido no comando de entrada na saudação e utilizando os comandos de saída e variáveis.

A segunda aula também incluiu a introdução aos conceitos de expressões aritméticas e lógicas dentro do Scratch como mostrado na Figura 11. Os alunos não tiveram nenhuma dificuldade sequer com a manipulação de operações lógicas, incluindo as expressões relacionais envolvendo maior que, menor que, igualdade, entre outros, e as expressões booleanas envolvendo OR, AND e NOT. Os alunos foram instruídos a observarem as diferenças entre os blocos aritméticos, que tinham formas arredondadas, e os blocos lógicos, que tinham formato de hexágono, associando essas formas aos encaixes e seus retornos (aritméticos retornando um numeral e lógicos retornando um valor booleano) como mostra a Figura 12.

Os desafios propostos para o final dessa aula envolviam todos os conceitos abordados até então e tinham uma escala de dificuldade de "desafio", "desafio extra" e "desafio hard". O primeiro consistia em fazer o gato receber dois números e falar a soma deles; o segundo envolvia o gato receber três números e calcular a média deles; e o terceiro, o mais difícil, envolvia os

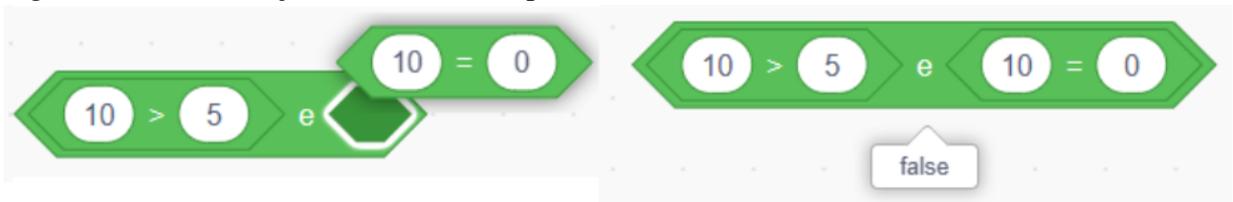
alunos entenderem a fórmula relacional entre Celsius e Fahrenheit e a utilizarem para fazer o gato receber a temperatura em Celsius e falar a temperatura convertida em Fahrenheit.

Figura 11 – Expressões aritméticas e lógicas no Scratch.



Fonte: Elaborado pelo autor (2024).

Figura 12 – Combinação de blocos de expressões no Scratch.



Fonte: Elaborado pelo autor (2024).

Quarta, 17 de Abril de 2024

Não houve aula devido à semana de provas na escola.

Quinta, 18 de Abril de 2024

Não houve aula devido à semana de provas na escola.

Quarta, 24 de Abril de 2024

Na terceira aula, foram abordadas as estruturas de decisão, nas quais os alunos compreenderam facilmente a ideia de estruturas condicionais. Foi apresentada a estrutura do bloco condicional "se ... então", que permite a execução de uma parte do programa dependendo da satisfação de uma condição lógica, apelidada de "pergunta lógica" ou simplesmente "pergunta". Os alunos associaram imediatamente o bloco de expressões lógicas visto na aula anterior ao local onde é feita a "pergunta" para alterar o fluxo do programa.

O exercício proposto envolveu fazer o gato pular utilizando condicionais. Os alunos notaram que o bloco "a tecla espaço está pressionada?" se encaixaria no lugar onde era feita a condição lógica, retornando uma resposta booleana que poderiam utilizar na estrutura de decisão.

Também foi apresentado e contextualizado o bloco condicional "se ... então ... senão", e foi mostrada sua diferença em relação ao "se ... então" apresentado anteriormente. O exercício proposto agora envolvia utilizar um número como entrada e identificar se esse número era par ou ímpar. Os alunos desenvolveram a "pergunta" necessária para satisfazer o comando de decisão a partir da noção de como distinguir um número par de um número ímpar usando os componentes lógicos e matemáticos.

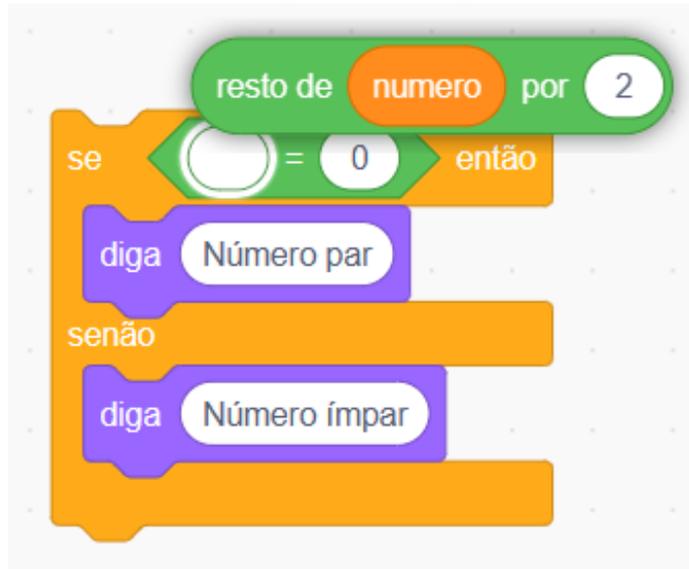
Inicialmente, os alunos sugeriram que "se um número terminar com 0, 2, 4, 6 ou 8, ele é par", mas perceberam que testar cada possibilidade individualmente seria complicado devido à infinidade de números. Então, uma aluna sugeriu utilizar a divisão por 2, ideia que foi desenvolvida pelo autor e pelos alunos.

Para ilustrar essa abordagem, foi utilizado o quadro branco do laboratório para escrever as ideias e operações necessárias. A divisão do número 10 por 2 foi proposta, resultando em 5 com resto 0. Em seguida, a divisão do número 11 por 2 foi realizada, resultando em 5 com resto 1. Após realizar mais algumas operações com números pares e ímpares, os alunos notaram um padrão relacionado ao resto da divisão. Eles então propuseram a seguinte pergunta para a estrutura condicional: "E se usássemos o resto da divisão como critério? Por exemplo, 'se o resto da divisão da minha variável por 2 for igual a zero, execute o primeiro bloco; caso contrário, execute o segundo'". Este exercício mental reforça a importância do reconhecimento

de padrões, um dos pilares do Pensamento Computacional, na resolução de problemas.

Ao explorar a ferramenta Scratch sem a assistência do professor, a maioria dos alunos concluiu que poderiam utilizar o bloco “resto de ... por ...” para implementar a estrutura de decisão proposta no exercício. Este bloco estava localizado junto aos blocos de operações e possuía o formato arredondado típico de uma operação aritmética, conforme ilustrado na figura 13.

Figura 13 – Padrão arredondado para combinação de blocos do Scratch.



Fonte: Elaborado pelo autor (2024).

Quinta, 25 de Abril de 2024

Na quarta aula, foram abordados os temas das estruturas de repetição, e os alunos foram introduzidos a alguns conceitos relacionados ao plano cartesiano para compreender as dimensões do palco e como poderiam utilizar as coordenadas de x e y do personagem para resolver os desafios. Também receberam uma revisão sobre testes de decisão utilizando estruturas lógicas, pois isso seria importante para a realização dos desafios seguintes.

A aula teve início com os conceitos básicos sobre estrutura de repetição e sua importância na criação de programas complexos, após isso foram introduzidos os desafios envolvendo cada variação da estrutura de repetição no Scratch.

Os alunos foram introduzidos ao comando “repita ... vezes” e desafiados a construir um programa que fizesse o gato contar de 1 até 100. Após um momento desenvolvendo como criar o algoritmo para o desafio, todos conseguiram entender que precisariam utilizar uma variável para somar 1 a ela e fazer o gato falar esse número a cada repetição como mostra a Figura 14.

Figura 14 – Bloco de repetição "repita ... vezes" utilizado nos desafios.

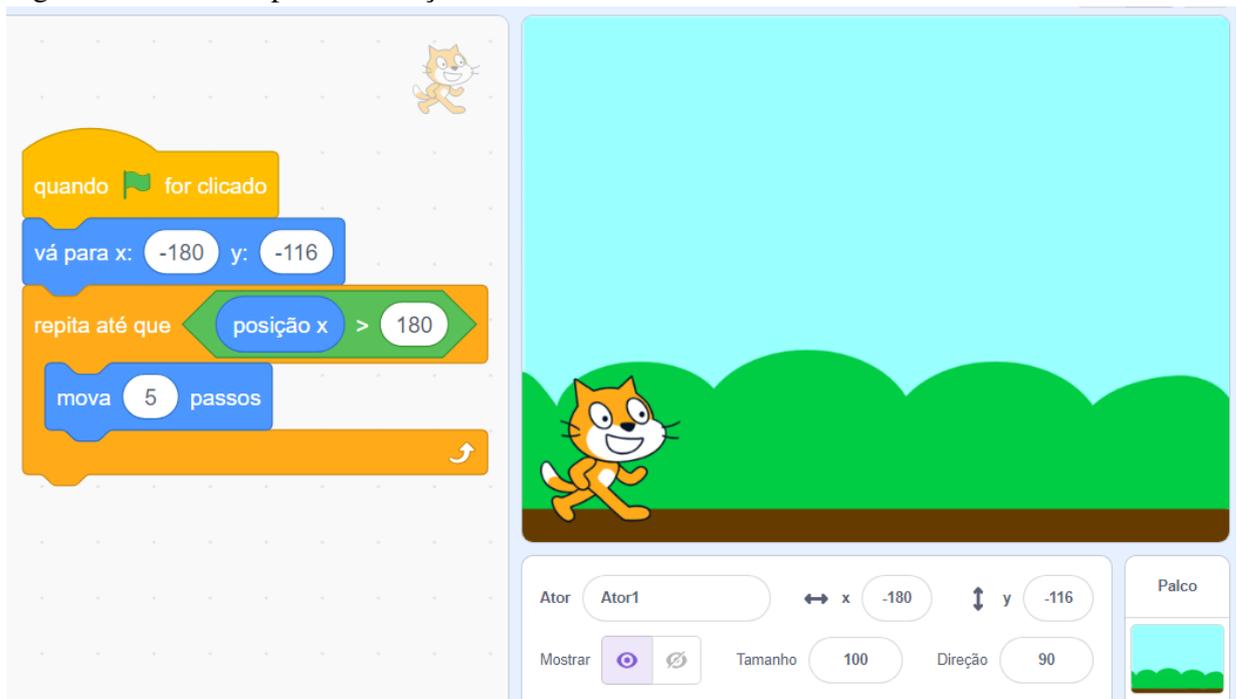


Fonte: Elaborado pelo autor (2024).

Os alunos foram então introduzidos ao comando de repetição "repita até que ...", que permite definir um teste de condição para interromper o laço de repetição. O desafio proposto era criar um programa no qual o gato se movesse de uma ponta à outra do palco, partindo do lado esquerdo até o lado direito.

Logo ao iniciar o desafio, foi percebido que poderiam ser utilizadas as coordenadas x e y do personagem para definir posições de início e fim e utilizar essas posições como condição de parada para o laço de repetição como demonstra na Figura 15. A aula foi encerrada após esse desafio, pois o tempo disponível já tinha se esgotado, e os conteúdos restantes seriam continuados na aula seguinte.

Figura 15 – Área de palco e de ações utilizados nos desafios.



Fonte: Elaborado pelo autor (2024).

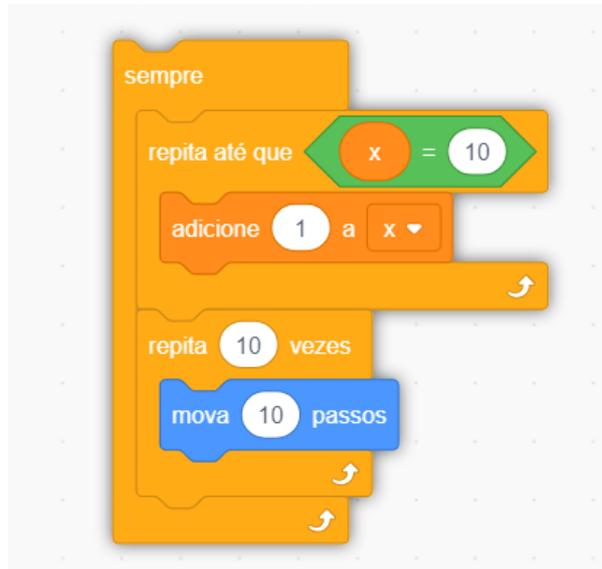
Quarta, 01 de Março de 2024

Não houve aula devido ao feriado do dia do trabalhador.

Quinta, 02 de Março de 2024

Na quinta aula, os conteúdos de estruturas de repetição foram continuados e os conteúdos previstos foram finalizados com a aula sobre funções. Os alunos foram lembrados do assunto da aula passada e foram apresentados à última estrutura de repetição, "repita sempre", e aos conceitos de comandos de repetição aninhados expostos na Figura 16.

Figura 16 – Bloco de repetição "sempre" aninhado com outros blocos de repetição.



Fonte: Elaborado pelo autor (2024).

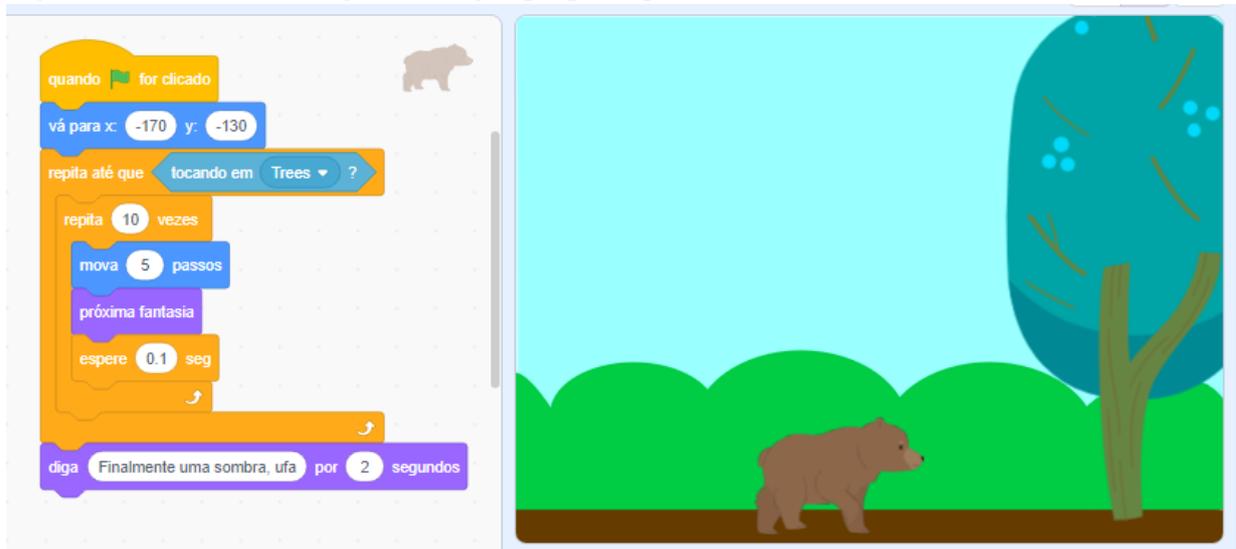
Dessa forma, foi proposto o último desafio envolvendo comandos de repetição. Os alunos deveriam criar uma cena em que o personagem urso se move do lado esquerdo para o lado direito do palco, onde encontraria o personagem árvore. O urso deveria andar até a árvore e parar quando estivesse debaixo dela, utilizando um comando de repetição. Além disso, enquanto se movia, o urso deveria trocar de fantasia, utilizando um segundo comando de repetição aninhado ao primeiro. A solução proposta para o exercício está ilustrada na Figura 17.

Para a surpresa do autor, a maioria dos alunos conseguiu reproduzir a cena com apenas um comando de repetição, fazendo com que o urso andasse até a árvore e, simultaneamente, trocasse de fantasia automaticamente através de um comando interno e resolvendo o desafio de uma forma mais simples e eficiente. Esta solução é apresentada na Figura 18.

Esta abordagem dos alunos demonstrou a aplicação prática dos conceitos de Pensamento Computacional, como a abstração para compreender o novo problema reconhecendo as novas necessidades e o reconhecimento de padrões para aplicar soluções desenvolvidas em

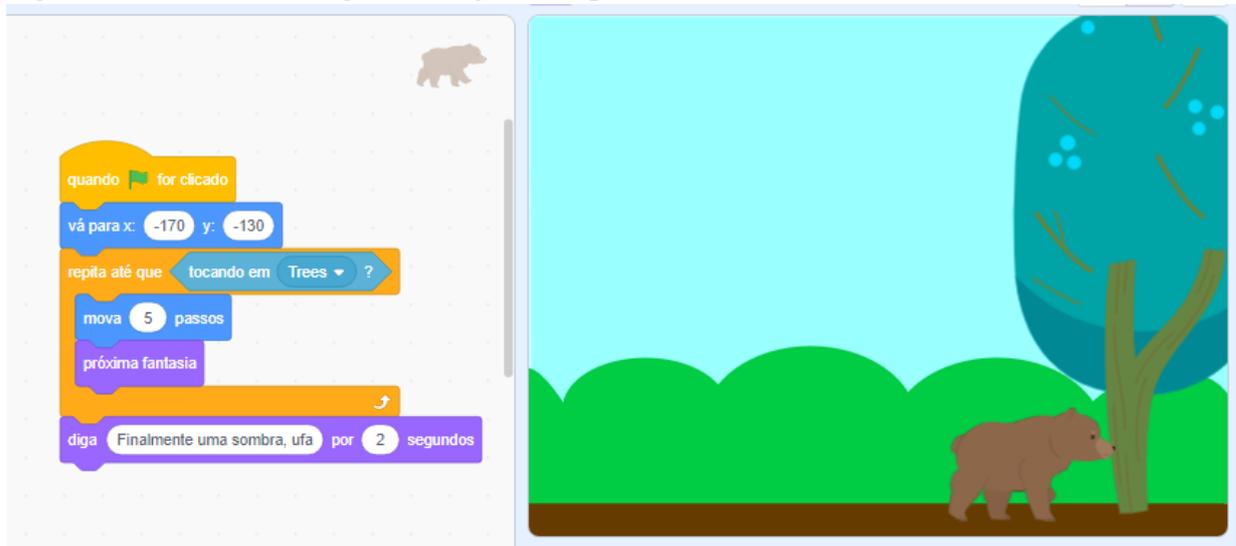
desafios anteriores.

Figura 17 – Bloco de código da solução proposta para o desafio.



Fonte: Elaborado pelo autor (2024).

Figura 18 – Bloco de código da solução feita pelos alunos.



Fonte: Elaborado pelo autor (2024).

A aula sobre funções teve início na segunda metade da aula. Os alunos foram introduzidos ao conteúdo por meio de uma analogia com as funções matemáticas que retornam um elemento y no conjunto imagem para cada elemento x no conjunto domínio. Após essa abordagem, os alunos foram introduzidos aos conceitos de funções na programação, de funções internas da linguagem, de criação e utilização de suas próprias funções no Scratch (Figura 19), e de sua importância para a construção de programas complexos.

Os desafios dessa etapa consistiram em criar uma função que calcula o Índice de

Figura 19 – Bloco de código de demonstração da criação e da execução de funções no Scratch.



Fonte: Elaborado pelo autor (2024).

Massa Corpórea (IMC) de uma pessoa baseada nos parâmetros de altura e peso passados para a função, bem como a criação de funções para encontrar as raízes de uma equação de segundo grau utilizando a fórmula de Bhaskara. Devido ao tempo de aula, só foi possível aplicar o desafio envolvendo a criação de função para o cálculo do IMC.

Quarta e quinta, 08 e 09 de Março de 2024

A sexta e a sétima aulas foram dedicadas à apresentação da dinâmica proposta, que serviria como atividade final para validar os conhecimentos adquiridos durante as aulas e tornar o desenvolvimento das habilidades de programação e do Pensamento Computacional mais dinâmicos, exercitando a autonomia e a criatividade dos alunos.

A dinâmica ocorreu conforme descrito na Seção 4.2.6.1 e foi dividida entre as apresentações do funcionamento do programa e dos blocos utilizados para criá-lo. Na sexta aula, a maioria dos projetos foi apresentada, enquanto na sétima e última aula foram mostrados os restantes, juntamente com a cerimônia de premiação dos três melhores projetos, os registros fotográficos dos alunos e o preenchimento do formulário de feedback como foi apresentado na Seção 4.2.6.2.2.

4.4 Considerações finais

Neste capítulo, foram descritas as atividades realizadas com o intuito de atingir os objetivos deste trabalho. O planejamento das ações foi conduzido de forma satisfatória e viável, incluindo a criação de material exclusivo para a ministração das aulas, revisão minuciosa de todo o conteúdo pelo orientador deste trabalho de Trabalho de Conclusão de Curso (TCC) e treinamentos internos fornecidos pelo Learninglab. A execução das aulas ocorreu de maneira

presencial em uma escola de ensino médio, com a totalidade das aulas ministrada pelo autor deste trabalho e com a participação de monitores voluntários do projeto Learninglab.

As aulas foram divididas em momentos teóricos e práticos, e cada aula foi seguida de desafios para reforçar o aprendizado. A condução das aulas foi satisfatória, oferecendo uma visão abrangente dos temas abordados e uma compreensão das suas complexidades. Por exemplo, na aula sobre repetição, alguns tópicos não previstos precisaram ser complementados durante a aula de funções, sendo esse um ponto de melhoria para próximas execuções deste experimento.

Nas últimas duas aulas, uma atividade dinâmica possibilitou aos alunos criar e apresentar seus próprios programas, aplicando todo o conhecimento adquirido. Isso não apenas promoveu o desenvolvimento da criatividade e do trabalho em grupo, mas também aprimorou suas habilidades em computação e Pensamento Computacional.

Foram utilizados dois formulários: um inicial para capturar informações demográficas e socioeconômicas e conhecimentos prévios dos alunos em tecnologia, e um de feedback para avaliar a metodologia e a eficácia das aulas. Essas avaliações forneceram dados valiosos para o experimento proposto neste trabalho.

Assim, as aulas desenvolvidas representaram uma experiência educacional enriquecedora para os alunos da turma de Desenvolvimento de Sistemas. Essa pesquisa demonstrou o potencial da tecnologia como uma ferramenta eficaz para o ensino de conceitos complexos de computação, ao mesmo tempo em que incentivou o engajamento dos alunos e promoveu o desenvolvimento de habilidades cruciais de Pensamento Computacional de maneira lúdica, eficaz e envolvente. Isso contribuiu para fortalecer o entendimento dos alunos sobre Pensamento Computacional e programação, enquanto promovia a colaboração e a criatividade. Esses e outros aspectos serão expostos detalhadamente no Capítulo 5.

5 RESULTADOS

5.1 Introdução

Este capítulo apresenta uma análise detalhada dos resultados obtidos a partir dos dados coletados nos questionários e dos resultados da avaliação do aprendizado, mencionados na Seção 4.2.6. O experimento, realizado com alunos do primeiro ano do ensino médio da escola profissionalizante Professor Walquer Cavalcante Maia, teve como objetivo promover o desenvolvimento do Pensamento Computacional, utilizando o Scratch como ferramenta pedagógica.

As análises seguintes buscam não apenas descrever os dados coletados durante o processo, mas também identificar as relações entre os conhecimentos adquiridos, o perfil dos alunos e suas percepções sobre programação. Além disso, será examinado como diferentes perfis dos estudantes, incluindo suas experiências prévias, acesso a recursos tecnológicos e preferências de aprendizagem, influenciaram seus resultados no curso. Por fim, essas análises contribuirão para responder às questões de pesquisa propostas e fornecerão informações valiosas sobre a eficácia do ensino de programação para promover habilidades de Pensamento Computacional no contexto do ensino médio. Os dados completos utilizados nesta análise podem ser encontrados no Apêndice C.

5.2 Questionário inicial

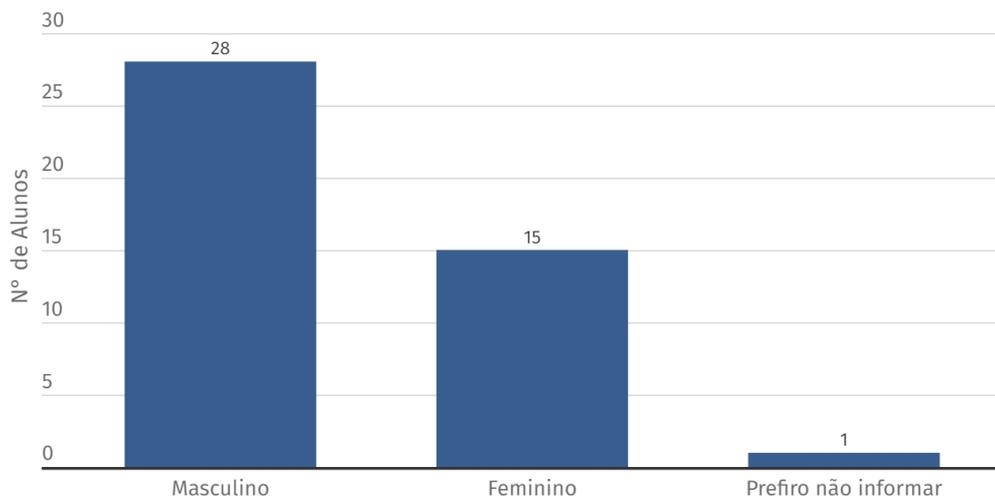
O questionário inicial, composto por 21 perguntas, foi aplicado no primeiro dia do experimento para entender o perfil sociodemográfico dos participantes, suas experiências anteriores com programação, suas condições de acesso a recursos tecnológicos e suas expectativas em relação ao curso de Pensamento Computacional utilizando Scratch. Os dados coletados foram analisados por meio de técnicas descritivas, o que permitiu identificar padrões e tendências que podem influenciar o aprendizado dos alunos. Ao todo, houve 45 respostas ao questionário; no entanto, um dos participantes desistiu durante o experimento, e suas respostas iniciais foram desconsideradas. Isso assegura uma análise precisa e contribui para a consistência dos resultados obtidos.

5.2.1 Questões sociodemográficas

A amostra do estudo foi composta por 44 alunos, sendo 28 do sexo masculino, 15 do sexo feminino e 1 que preferiu não informar (Figura 20). A faixa etária predominante foi de 15 anos (Figura 21).

Figura 20 – Distribuição de alunos por sexo.

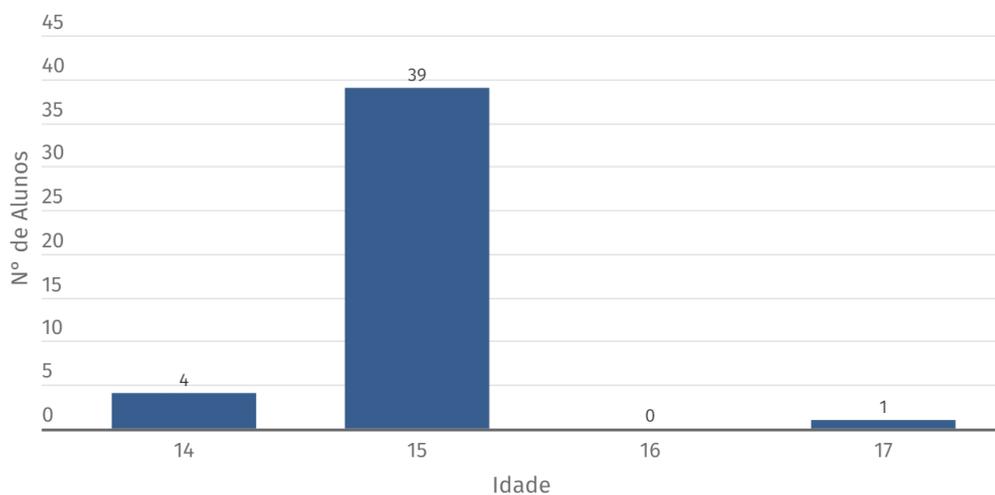
Quantidade de alunos por sexo



Fonte: Elaborado pelo autor (2024).

Figura 21 – Distribuição de alunos por idade.

Quantidade de alunos por idade

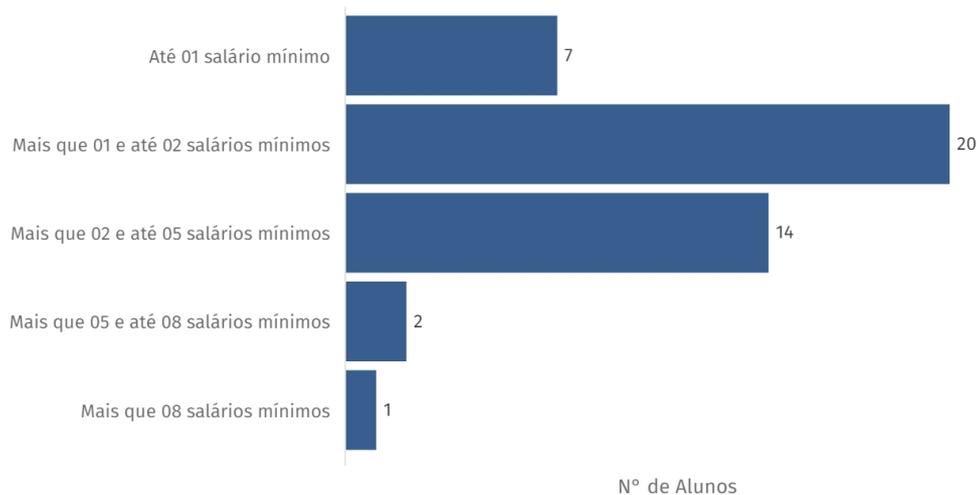


Fonte: Elaborado pelo autor (2024).

Em relação à situação socioeconômica, a maioria dos alunos respondeu que suas famílias possuem renda entre um e dois salários mínimos (45,4% das repostas), conforme demonstrado na Figura 22. O nível de escolaridade dos pais indica que 43,2% dos pais e 56,8% das mães possuem ensino médio completo (Figura 23).

Figura 22 – Distribuição de renda por faixa salarial.

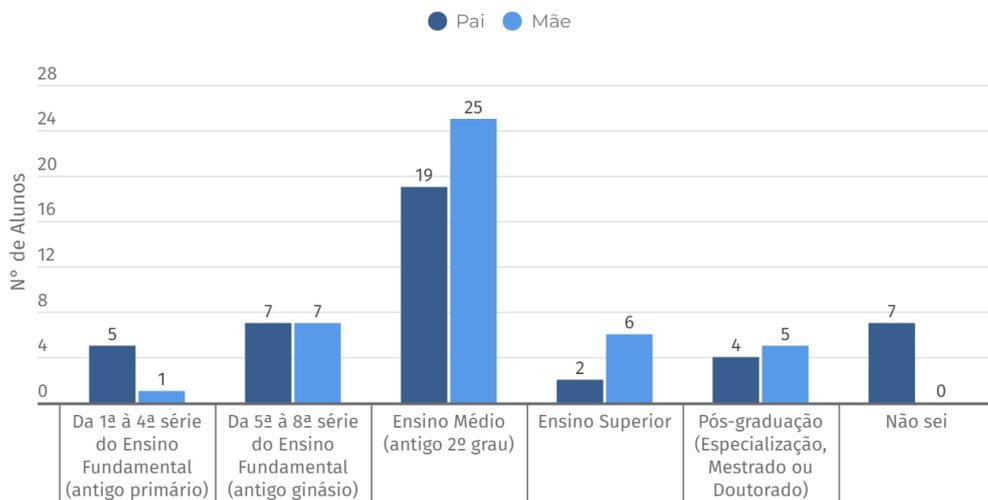
Quantidade de alunos por faixa salarial



Fonte: Elaborado pelo autor (2024).

Figura 23 – Distribuição de nível de escolaridade dos pais por aluno.

Qntd. de alunos por nível de escolaridade dos pais

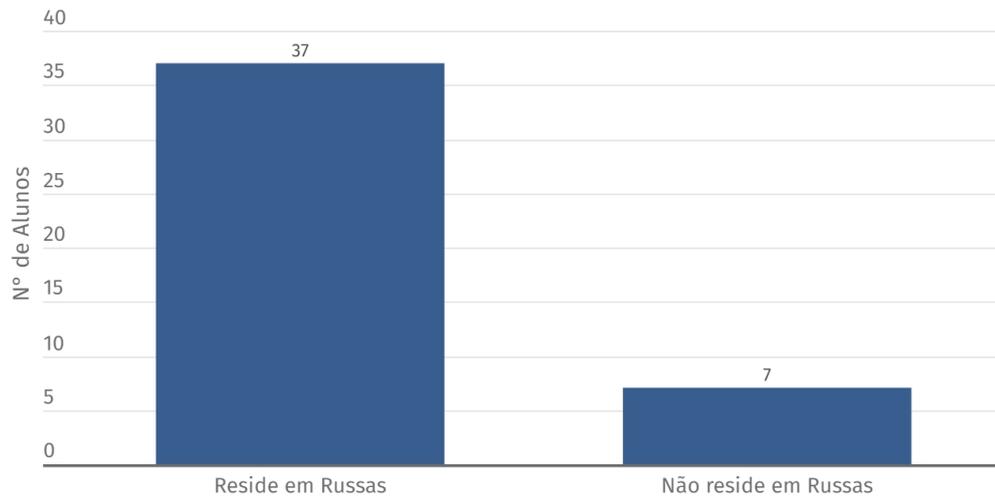


Fonte: Elaborado pelo autor (2024).

A maior parte dos estudantes, cerca de 84,1%, reside em Russas. O transporte mais utilizado para se deslocar até a escola é o ônibus escolar, com um tempo médio de trajeto entre 5 e 10 minutos (Figuras 24, 25 e 26).

Figura 24 – Distribuição de alunos que residem em russas.

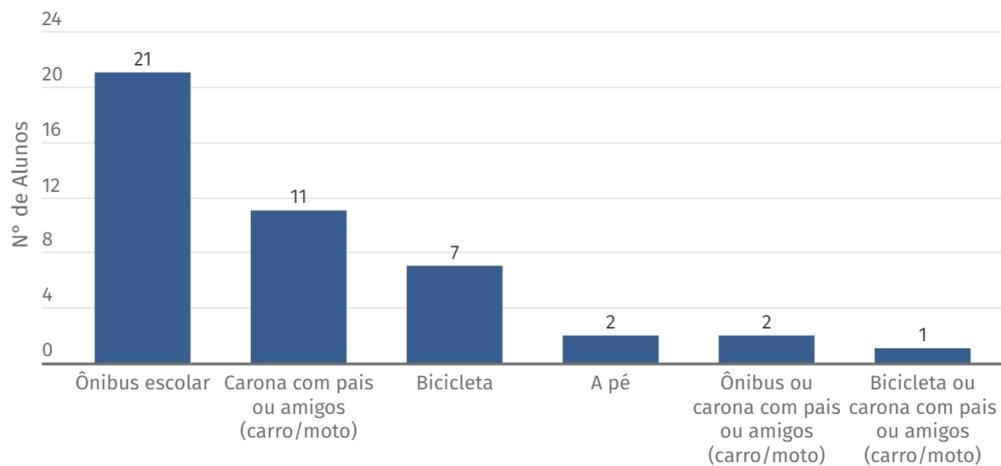
Quantidade de alunos que residem em Russas



Fonte: Elaborado pelo autor (2024).

Figura 25 – Distribuição de transporte utilizado para chegar à escola por aluno.

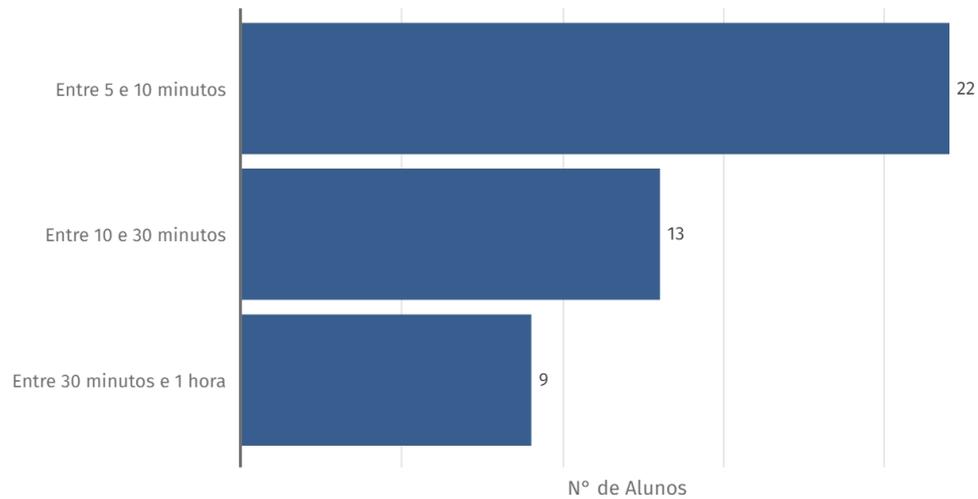
Qntd. de alunos por transporte utilizado para chegar à escola



Fonte: Elaborado pelo autor (2024).

Figura 26 – Distribuição de alunos por tempo médio para chegar à escola.

Tempo médio por aluno para chegar à escola

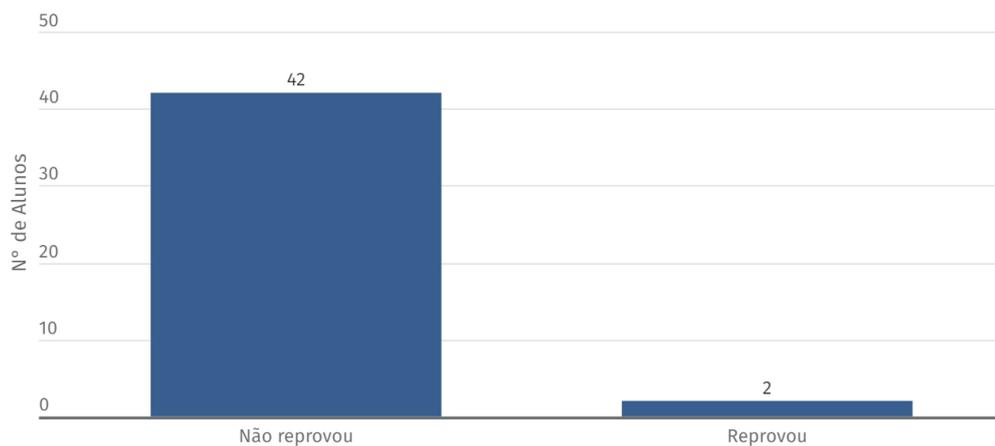


Fonte: Elaborado pelo autor (2024).

Quanto à experiência escolar, 95,4% dos alunos nunca reprovaram. Apenas 2 alunos relataram reprovação, sendo um em duas ocasiões e outro em uma única (Figura 27).

Figura 27 – Distribuição de alunos repetentes.

Quantidade de alunos que já reprovaram

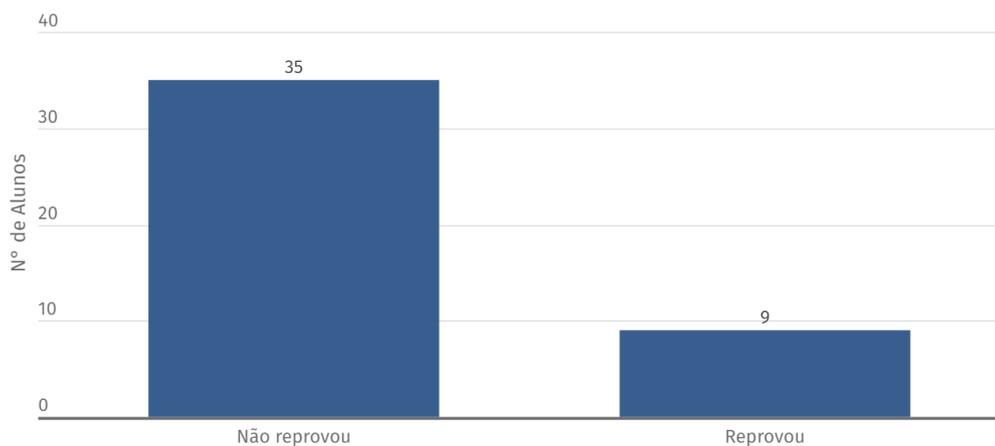


Fonte: Elaborado pelo autor (2024).

Um dado relevante é o alto interesse dos alunos em matemática. Cerca de 79,5% dos alunos afirmaram gostar da disciplina, destacando a resolução de problemas e o uso da lógica como os principais motivos. Por outro lado, 20,5% dos alunos declararam não gostar de matemática, citando dificuldades com números e cálculos ou apenas não se identificam com a matéria como justificativas (Figura 28).

Figura 28 – Contagem de alunos que gostam de matemática.

Quantidade de alunos que gostam de matemática



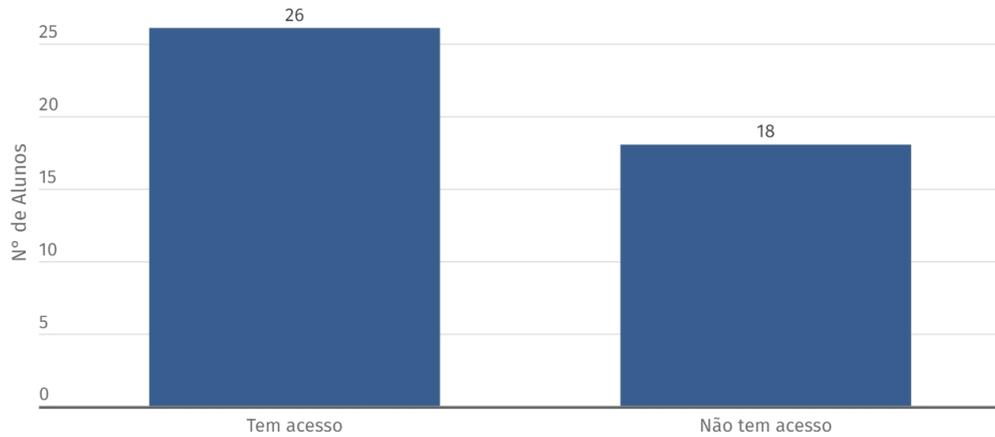
Fonte: Elaborado pelo autor (2024).

5.2.2 Questões de pesquisa

A pergunta sobre o acesso a computador em casa recebeu uma resposta positiva de 59,1% dos alunos e uma negativa de 40,9%, conforme descrito na Figura 29. Em relação ao acesso à internet, os resultados indicam que todos os alunos têm acesso, como mostrado na Figura 30.

Figura 29 – Contagem de alunos que tem acesso a computador em casa.

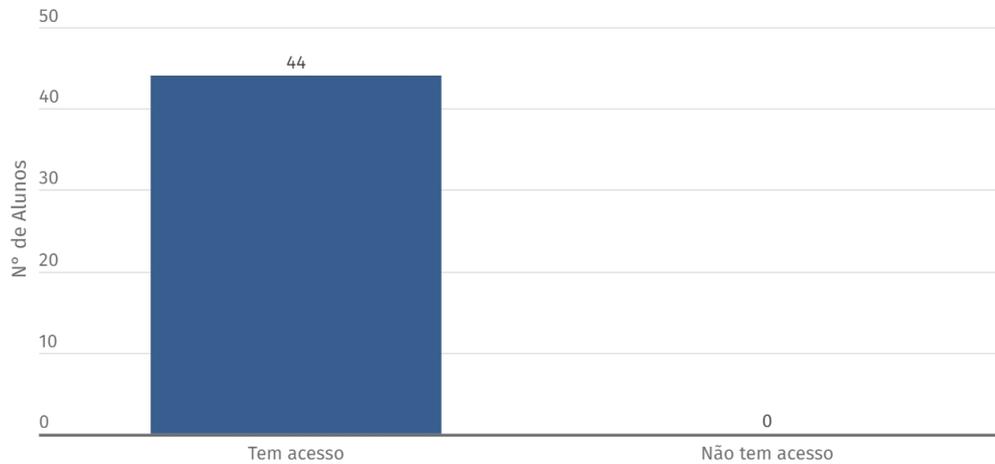
Quantidade de alunos que têm acesso a um computador em casa



Fonte: Elaborado pelo autor (2024).

Figura 30 – Contagem de alunos que tem acesso a internet em casa.

Quantidade de alunos que têm acesso a internet em casa

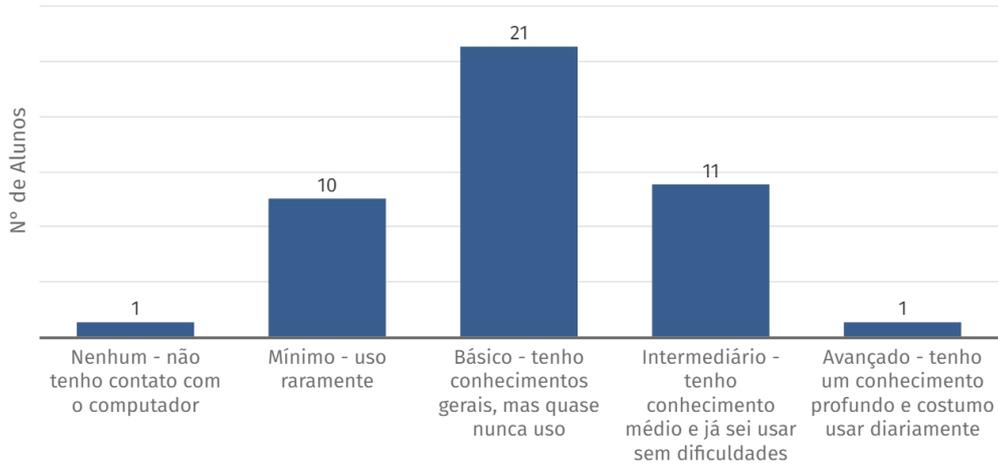


Fonte: Elaborado pelo autor (2024).

Quanto ao nível de conhecimento em informática básica, a maioria dos alunos (47,7%) declarou como tendo um nível básico. A distribuição dos resultados, conforme apresentado na Figura 31, indica uma homogeneidade considerável no grupo, com uma distribuição próxima à normal.

Figura 31 – Distribuição de alunos por nível de conhecimento em informática básica.

Qntd. de alunos por nível de conhecimento em informática básica

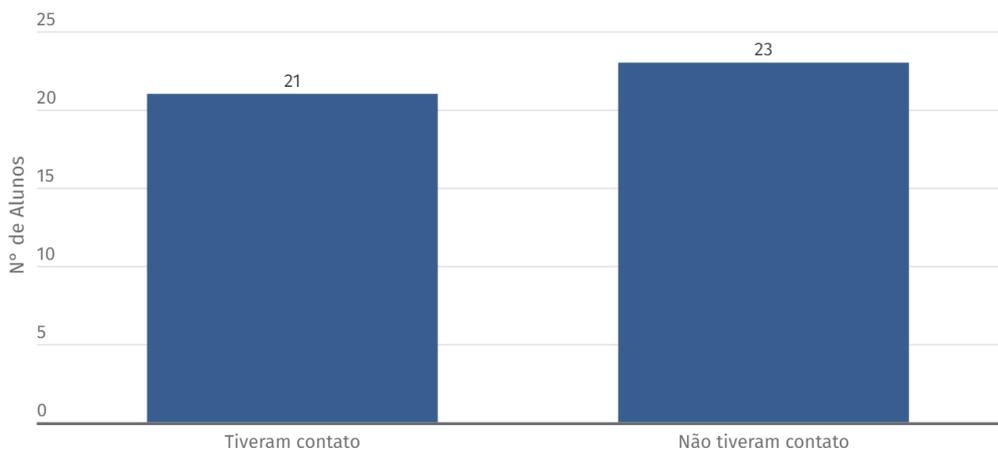


Fonte: Elaborado pelo autor (2024).

Em relação ao contato prévio dos alunos com programação, 21 alunos responderam de que já tiveram contato, o que representa 47,7% dos alunos e 23 dos alunos responderam que não tiveram nenhum contato com programação, representando 52,3% dos alunos como mostrado na Figura 32.

Figura 32 – Distribuição de alunos que já tiveram contato com programação.

Quantidade de alunos que já tiveram contato com programação



Fonte: Elaborado pelo autor (2024).

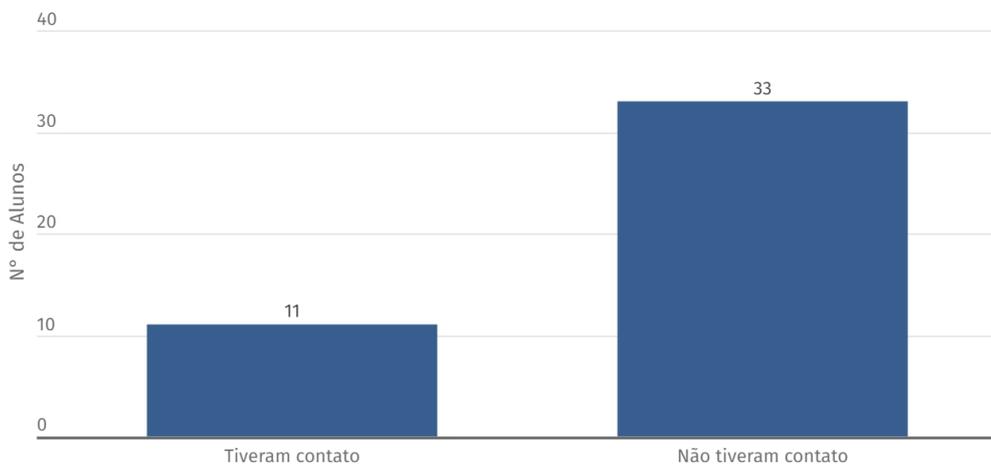
Dentre os alunos que tiveram experiências em programação, alguns as descreveram como:

- “Apenas uma aula prática do curso de Desenvolvimento de Sistemas, mas que foi bastante interessante e divertido aprender sobre, mesmo não sabendo muito como programar ou algo relacionado.”
- “Meu contato com a programação foi algo de interesse próprio, eu gostava bastante sobre desenvolvimento de jogos, e buscava aprender sobre isso, então aprendi sobre lógica de programação e a programar só o básico, e também comecei a ter interesse em desenvolver sistemas no geral, hoje eu sei um pouco de HTML e python.”
- “Apenas operações básicas no python.”
- “Já cheguei a iniciar um curso de JS pelo o Youtube, foi bem legal, foi isso que me trouxe até aqui.”
- “Python, bloco e java básico.”

A questão sobre o contato prévio dos alunos com programação visual revelou que apenas 11 alunos, o que corresponde a 25% do total, afirmaram ter tido alguma experiência com a programação visual. Em contraste, 33 alunos, representando 75%, informaram não ter tido nenhum contato com programação visual, conforme ilustrado na Figura 33.

Figura 33 – Distribuição de alunos que já tiveram contato com programação visual.

Quantidade de alunos que já tiveram contato com programação visual



Fonte: Elaborado pelo autor (2024).

Dentre os alunos que tiveram experiências em programação visual, alguns as descre-

veram como:

- “Quando eu estava aprendendo a desenvolver jogos uma das formas que pensei em fazer isso foi programar em bloco, pois eu não sabia sobre os códigos de programação.”
- “No 8º ano do ensino fundamental, houve um curso de robótica, durante algumas aulas foi utilizado um site parecido com o Scratch.”
- “Programação de controladores, atuadores e sensores no uso da robótica.”
- “Vi um pouco apenas de programação em blocos, usando Tinkercard [tinkercad], e fazendo alguns códigos para acender e apagar pequenas luzes em um arduíno.”

Por fim, também foi discutido, em uma questão aberta, quais as expectativas dos alunos em relação ao curso e algumas respostas incluíram:

- "Lógica de programação."
- "Obter uma visão do uso básico de programação."
- "Como programar, e onde aplicar o que aprendo."
- "O básico da programação e programar jogos."
- "Adquirir mais conhecimento sobre informática e programação."
- "Lógica de programação com Scratch ou pelo menos entender um pouco sobre."
- "Espero desenvolver a parte lógica, que é onde eu tenho mais dificuldade."
- "Aprender melhor como utilizar variáveis e outros recursos da programação, para ter um melhor desempenho durante o curso de Desenvolvimento de Sistemas."
- "Espero aprender mais sobre programação, de maneira a facilitar meu entendimento quando ver assuntos mais complexos da programação."
- "Aprender a como resolver problemas de programação."

5.2.3 *Análise sobre questionário inicial*

Os dados revelam um perfil de turma predominantemente masculino, com idade média de 15 anos e sem histórico de reprovação. A afinidade com a matemática é notável; no entanto, é importante destacar que cerca de 20% dos alunos relatam dificuldades em cálculos ou falta de identificação com a disciplina.

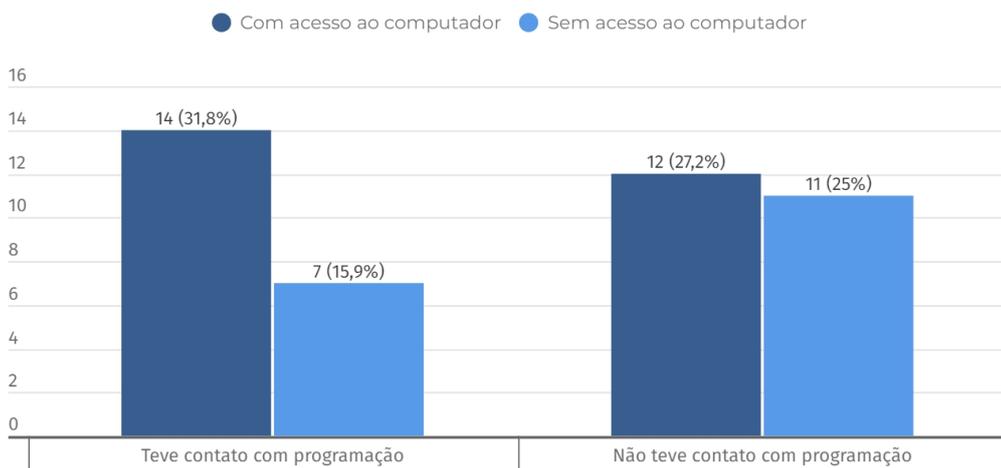
Quanto ao acesso a recursos tecnológicos, constata-se que quase a totalidade dos alunos possui acesso à internet em casa. No entanto, o acesso a computadores pessoais é mais restrito, com 59,7% dos estudantes relatando possuir esse equipamento em seus lares. Portanto, conclui-se que uma parte significativa dos alunos acessa a internet predominantemente por meio

de celulares ou outros dispositivos que não sejam computadores.

É interessante notar uma relação entre o acesso a computadores e a experiência prévia com programação. A maioria dos alunos que já tiveram contato com programação (31,8%) possui um computador em casa, enquanto apenas 15,9% daqueles que não têm esse equipamento já programaram anteriormente (Figura 34). Esse padrão se inverte quando se considera a programação visual, onde os alunos sem computador em casa demonstram uma taxa ligeiramente maior de contato com essa modalidade (13,6% contra 11,3% dos alunos com computador) (Figura 35).

Figura 34 – Relação entre alunos que tiveram contato com programação e acesso a computador.

Contato com programação X Acesso a um computador

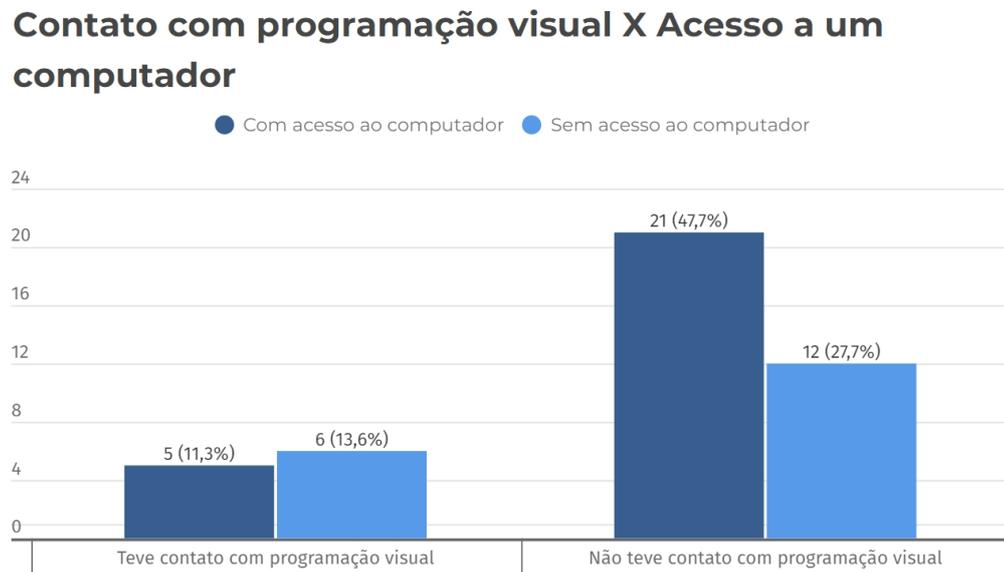


Fonte: Elaborado pelo autor (2024).

Essa relação entre acesso a computador e experiência com programação fica ainda mais evidente ao analisarmos os relatos dos alunos. Aqueles que não possuem computador em casa, em sua maioria, tiveram seu primeiro contato com a programação ou a programação visual durante atividades escolares, como cursos de robótica ou projetos específicos. Comentários como "No 8º ano do ensino fundamental, houve um curso de robótica, durante algumas aulas foi utilizado um site parecido com o Scratch" e "Tive contato em um curso de programação visual no Raimundo de Deus [Escola Municipal de Russas], em geral, consegui ajudar as pessoas com app que eu fiz e foi bom" ilustram essa realidade.

Analisando também a relação entre o acesso a computadores e o nível de conhecimento em informática declarado pelos alunos, percebemos uma correlação interessante. O

Figura 35 – Relação entre alunos que tiveram contato com programação visual e acesso a computador.



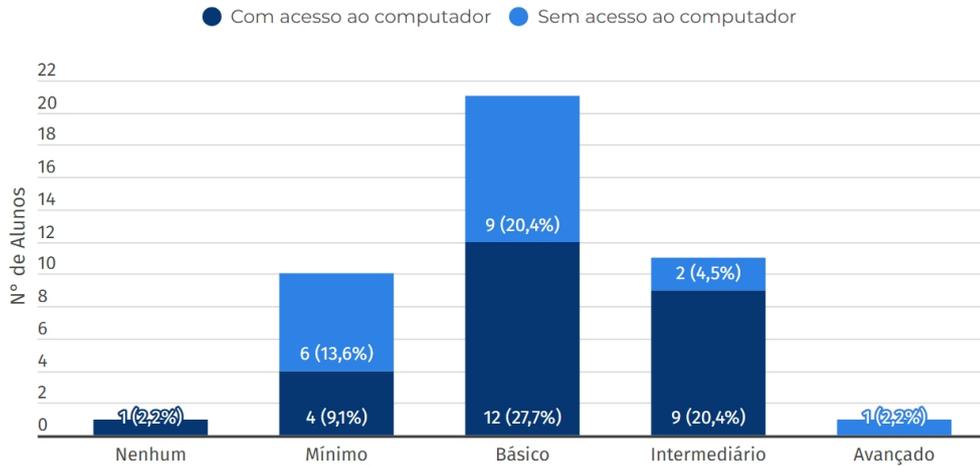
Fonte: Elaborado pelo autor (2024).

gráfico da Figura 36 revela que a maioria dos alunos com acesso a computador em casa declarou ter um nível de conhecimento em informática básico ou intermediário (48,1%), enquanto aqueles sem esse acesso predominantemente se autoavaliaram como tendo um nível mínimo ou básico (34%). Esses dados sugerem uma relação direta entre o acesso a computadores e o desenvolvimento de habilidades em informática, indicando que a familiaridade com dispositivos e *softwares* pode influenciar significativamente o aprendizado de programação.

Assim, os resultados apresentados até agora mostram que a turma é heterogênea em relação ao acesso a recursos tecnológicos, com uma parcela significativa de alunos sem acesso a computadores em casa. Isso evidencia que a disponibilidade de um computador pessoal é um fator importante para a exploração de conceitos de informática e programação. Além disso, destaca-se a relevância de projetos escolares que proporcionem oportunidades de contato com a programação para aqueles que não possuem computadores em casa, promovendo a democratização do acesso a esse conhecimento. A partir desses dados, é possível traçar um perfil geral da turma, onde a maioria dos alunos tem noções de informática, mas pode haver a necessidade de reforçar conceitos básicos.

Figura 36 – Relação entre os níveis de conhecimento em informática dos alunos e acesso a computador.

Nível de conhecimento em informática básica X Acesso a um computador



Fonte: Elaborado pelo autor (2024).

5.3 Questionário final

O questionário final foi aplicado ao término das aulas de programação e consistiu em 23 perguntas. Destas, 13 perguntas foram adaptadas do modelo de avaliação dTECT, as quais foram projetadas para avaliar a percepção dos alunos sobre a qualidade do curso, suas experiências com a computação e sua percepção de aprendizagem em relação às aulas de programação. As 10 perguntas restantes, tanto abertas quanto fechadas, foram elaboradas para explorar a percepção dos alunos sobre o conteúdo, a metodologia e a experiência geral das aulas. O questionário foi aplicado à mesma turma que recebeu o questionário inicial, resultando em 44 respostas.

5.3.1 Questões do modelo dTECT

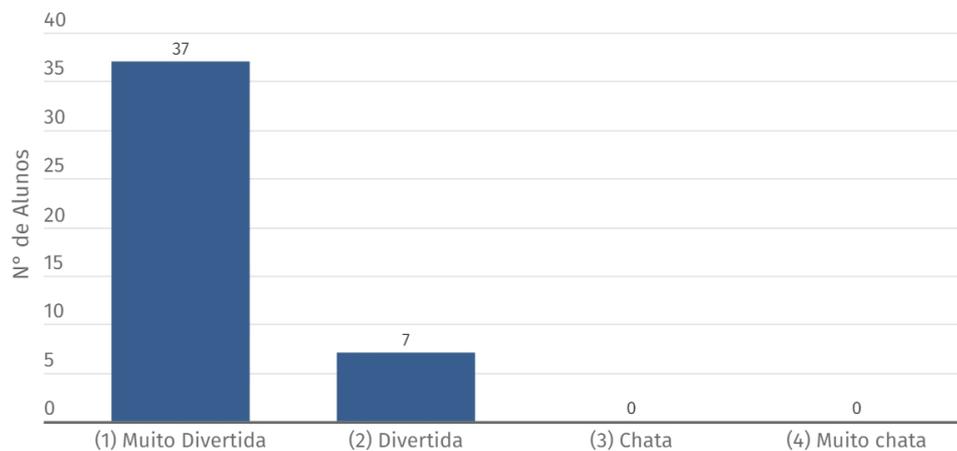
Os dados coletados a partir do questionário final revelam as percepções dos alunos sobre três diferentes fatores de qualidade. Primeiro, a qualidade da unidade instrucional, representada pelo curso desenvolvido, foi avaliada nas três primeiras perguntas dessa etapa do questionário. Em seguida, a experiência dos alunos com computação foi abordada nas seis perguntas subsequentes. Por fim, a percepção de aprendizado foi explorada nas últimas quatro perguntas, completando um total de 13 questões baseadas no modelo dTECT.

A totalidade dos participantes descreveu o curso como uma experiência muito divertida ou divertida, conforme demonstrado na Figura 37. Em relação à percepção do tempo, a maioria dos alunos (61,4%) relatou que as aulas passaram rápido, enquanto 31,8% consideraram que o tempo passou muito rápido. Uma pequena parcela (6,8%) relatou que o tempo passou devagar (Figura 38). Por fim, a avaliação geral do curso foi bastante positiva, com 86,4% dos alunos classificando-o como excelente e 13,6% como bom (Figura 39).

Figura 37 – Questão 1 do modelo dTECT em relação à qualidade institucional.

A oficina foi

Em relação à qualidade da Unidade Instrucional

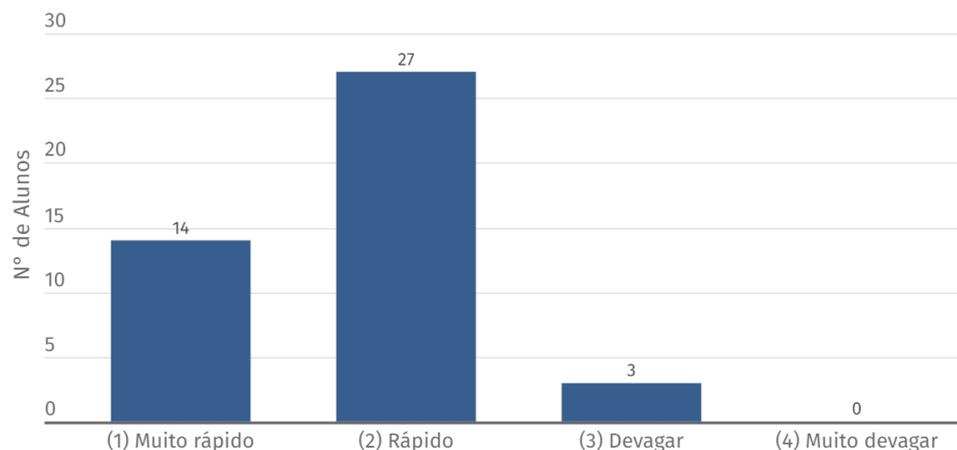


Fonte: Elaborado pelo autor (2024).

Figura 38 – Questão 2 do modelo dTECT em relação à qualidade institucional.

O tempo das aulas passou:

Em relação à qualidade da Unidade Instrucional

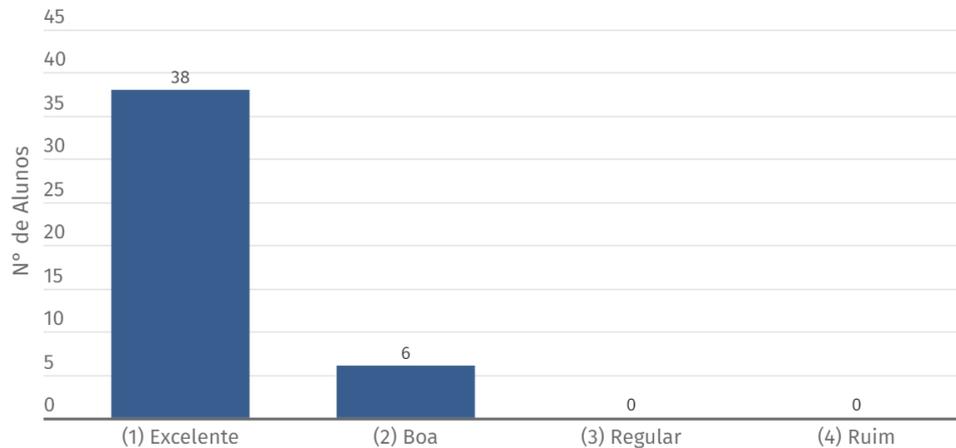


Fonte: Elaborado pelo autor (2024).

Figura 39 – Questão 3 do modelo dTECT em relação à qualidade institucional.

A oficina foi:

Em relação à qualidade da Unidade Instrucional



Fonte: Elaborado pelo autor (2024).

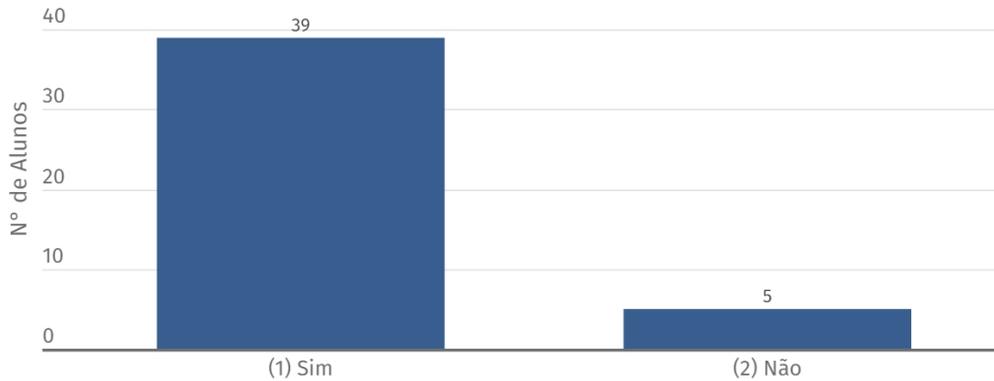
A avaliação altamente positiva do curso, tanto em termos de diversão quanto de conteúdo, indica que a abordagem metodológica utilizada foi eficaz em engajar os alunos e promover o aprendizado. A percepção de que o tempo passou rápido sugere um alto nível de envolvimento e interesse dos alunos nas atividades propostas. Além disso, a classificação geral do curso como excelente ou boa por uma grande maioria dos participantes corrobora a boa receptividade da metodologia empregada.

A grande maioria (88,6%) demonstrou interesse em compartilhar seus projetos com outras pessoas, e todos os participantes expressaram o desejo de aprofundar seus conhecimentos na área (Figuras 40 e 41). Além disso, a atividade de programar foi considerada muito divertida por 54,4% dos alunos e divertida por 45,5% (Figura 42), evidenciando o prazer que encontraram durante as atividades do curso.

Figura 40 – Questão 4 do modelo dTECT em relação à experiência em computação dos alunos.

Vou mostrar meu programa de computador para outras pessoas:

Em relação à experiência de computação

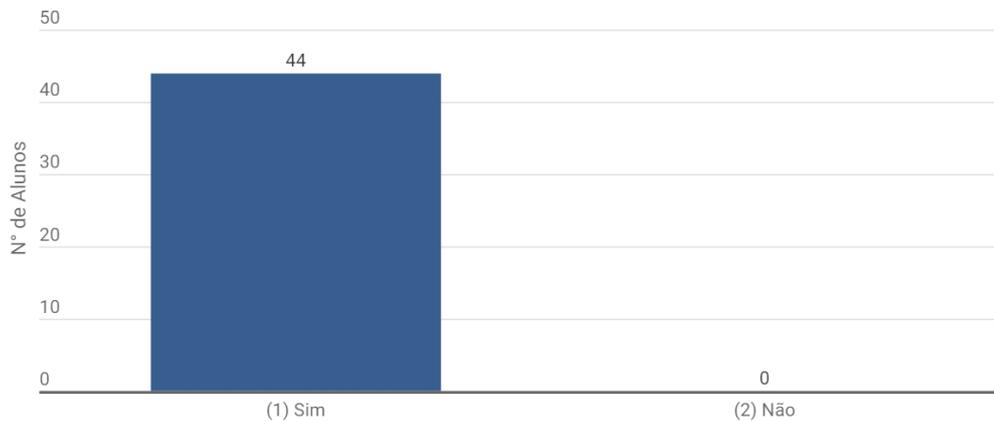


Fonte: Elaborado pelo autor (2024).

Figura 41 – Questão 5 do modelo dTECT em relação à experiência em computação dos alunos.

Quero aprender mais sobre como fazer programas de computador:

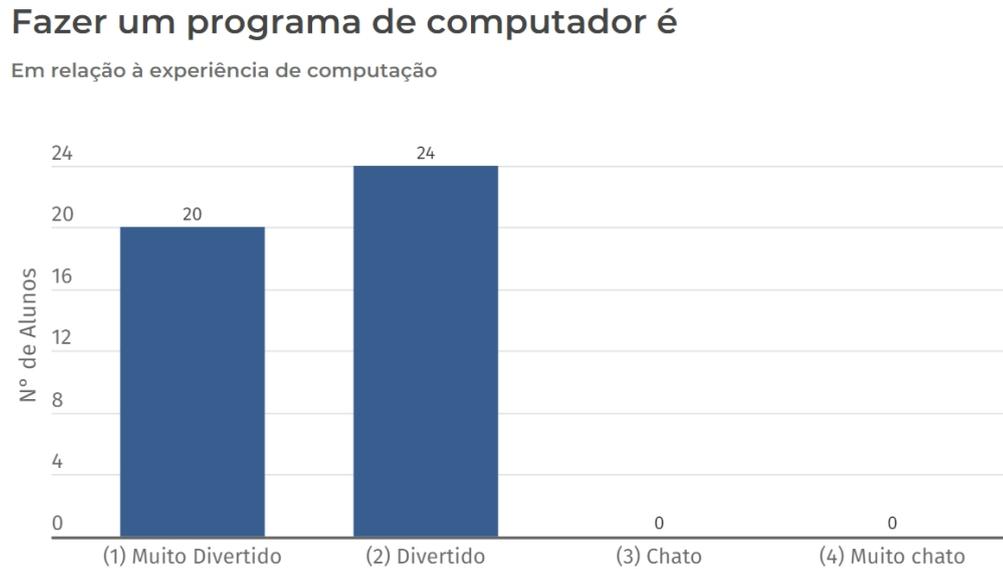
Em relação à experiência de computação



Fonte: Elaborado pelo autor (2024).

A afinidade dos alunos com a programação foi ainda mais evidente nas respostas às questões sobre o gosto pela atividade e sua utilidade no dia a dia. Todos os participantes (100%) afirmaram gostar de fazer programas de computador e reconhecer a utilidade da computação no dia a dia (Figuras 43 e 44). Além disso, 86,4% dos alunos manifestaram interesse em seguir carreira na área da computação (Figura 45), demonstrando um grande potencial para a formação de novos profissionais da tecnologia. É importante mencionar que, como a turma da escola

Figura 42 – Questão 6 do modelo dTECT em relação à experiência em computação dos alunos.



Fonte: Elaborado pelo autor (2024).

profissionalizante já era da área de tecnologia, esse número alto já era esperado.

Figura 43 – Questão 7 do modelo dTECT em relação à experiência em computação dos alunos.



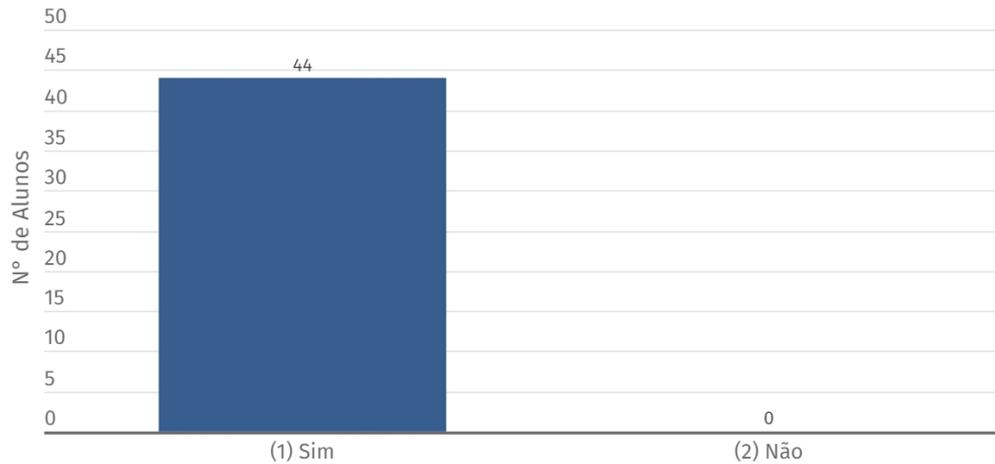
Fonte: Elaborado pelo autor (2024).

Os resultados apresentados destacam uma experiência positiva em relação às aulas de computação desenvolvidas. A alta adesão dos alunos às atividades do curso, evidenciada pelo interesse em compartilhar seus projetos e em aprofundar seus conhecimentos em programação, reflete uma percepção favorável dessa turma no estudo da programação. A avaliação positiva

Figura 44 – Questão 8 do modelo dTECT em relação à experiência em computação dos alunos.

A computação é útil no dia a dia

Em relação à experiência de computação

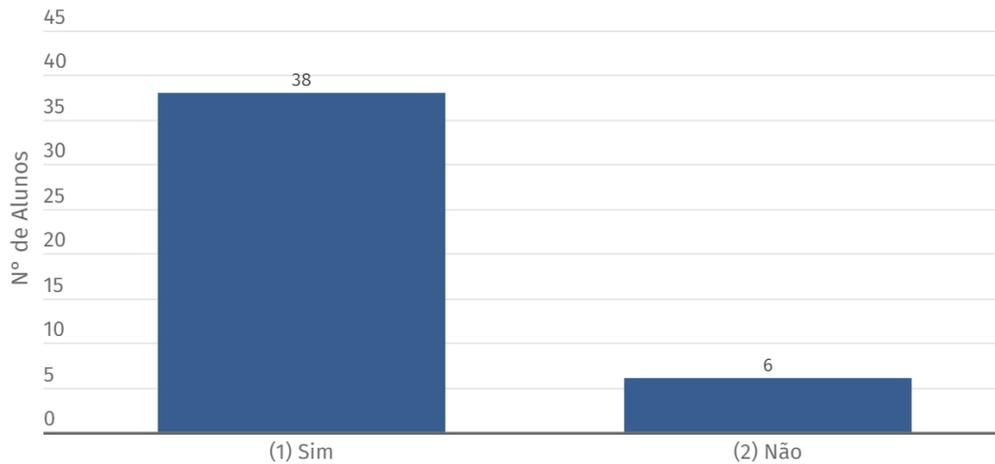


Fonte: Elaborado pelo autor (2024).

Figura 45 – Questão 9 do modelo dTECT em relação à experiência em computação dos alunos.

Eu penso em trabalhar com computação

Em relação à experiência de computação



Fonte: Elaborado pelo autor (2024).

do curso como divertido reforça a aceitação da abordagem lúdica no ensino de programação com Scratch, evidenciando seu potencial para desenvolver habilidades essenciais do século XXI, como Pensamento Computacional, criatividade e resolução de problemas. Além disso, o reconhecimento da importância da computação no cotidiano e a intenção de 86,4% dos alunos de seguir carreira na área sugerem a necessidade de ampliar as oportunidades de formação e desenvolvimento de projetos, contribuindo para a formação de uma nova geração de profissionais

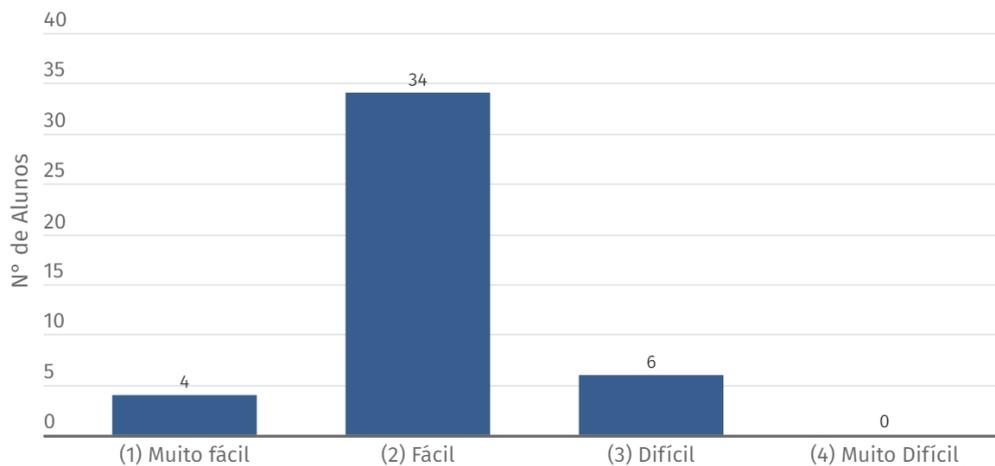
em tecnologia.

Sobre a dificuldade do curso, mais de dois terços dos alunos, cerca de 77,3%, classificaram como fácil, outros 9,1% dos alunos como muito fácil e para 13,6% como difícil como demonstra a Figura 46. A maioria dos estudantes também se mostrou confiante em criar seus próprios programas de computador (84,1%), enquanto uma pequena parte de 15,9% sinalizou que não conseguem (Figura 47). No entanto, a tarefa de explicar um programa para um colega foi considerada mais desafiadora, com apenas 70,5% dos participantes se sentindo seguros para fazê-lo (Figura 48). Por fim, ao serem questionados sobre a facilidade em criar programas de computador, cerca de metade dos alunos (54,5%) marcou como difícil, enquanto outros 43,2% marcou como fácil e apenas 2,3% marcaram como muito difícil como exposto na Figura 49.

Figura 46 – Questão 10 do modelo dTECT em relação à percepção de aprendizado dos alunos.

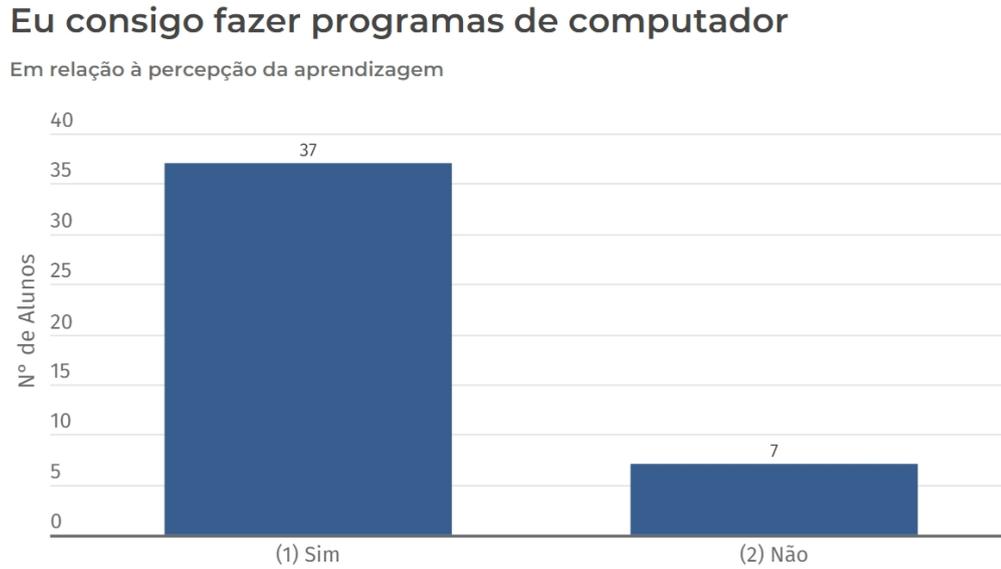
A oficina foi

Em relação à percepção da aprendizagem



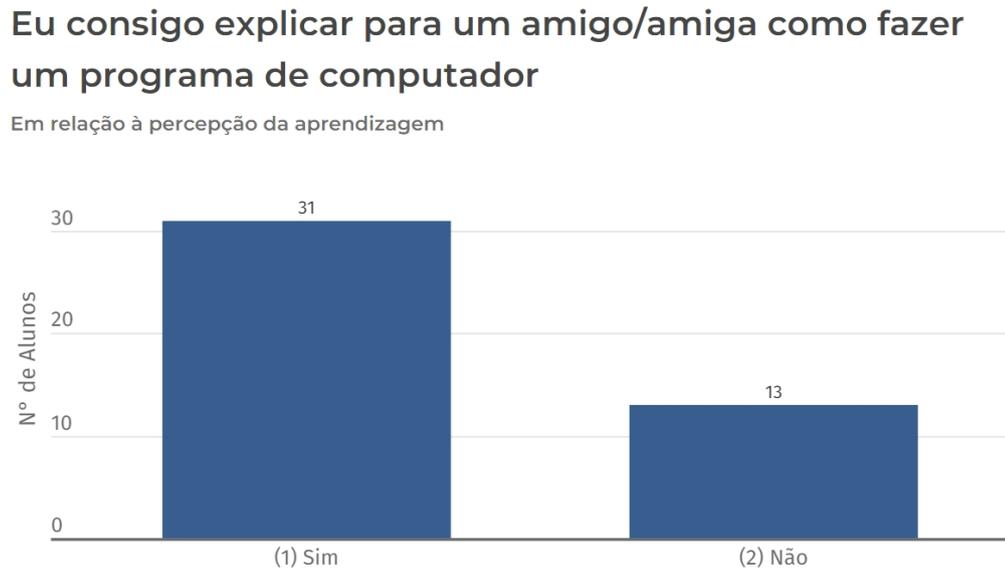
Fonte: Elaborado pelo autor (2024).

Figura 47 – Questão 11 do modelo dTECT em relação à percepção de aprendizado dos alunos.



Fonte: Elaborado pelo autor (2024).

Figura 48 – Questão 12 do modelo dTECT em relação à percepção de aprendizado dos alunos.



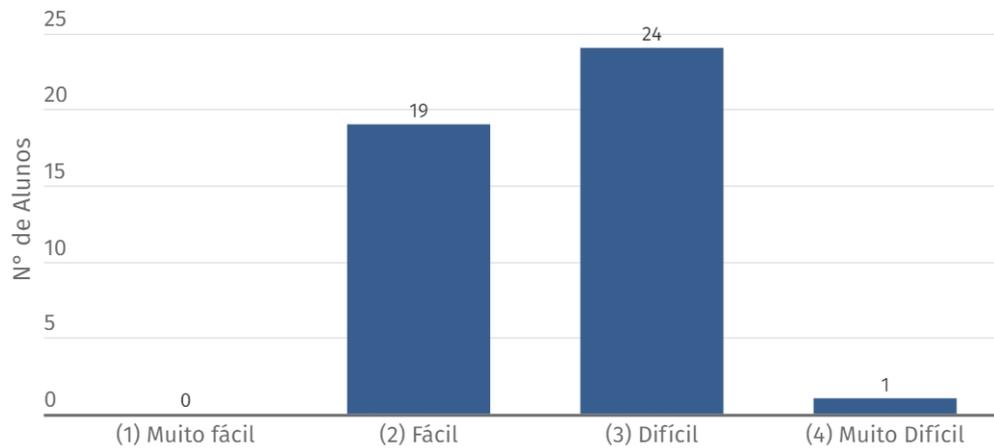
Fonte: Elaborado pelo autor (2024).

A maioria dos alunos percebeu o curso como fácil e se mostrou confiante em criar seus próprios programas, o que sugere que a plataforma Scratch é acessível e intuitiva para estudantes do ensino médio. No entanto, é interessante notar a aparente contradição entre a percepção de facilidade do curso e a dificuldade relatada na criação de programas. Enquanto a maioria dos alunos considerou o curso fácil e se sentiu capaz de criar seus próprios programas, mais da metade classificou a tarefa de criar um programa como difícil. Esses dados indicam

Figura 49 – Questão 13 do modelo dTECT em relação à percepção de aprendizado dos alunos.

Fazer um programa de computador é

Em relação à percepção da aprendizagem



Fonte: Elaborado pelo autor (2024).

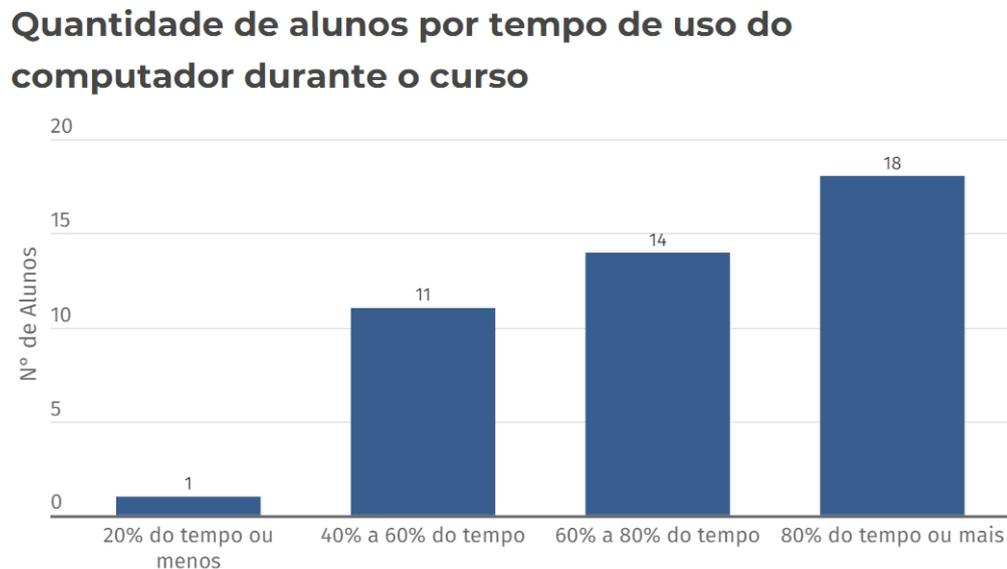
que, embora os alunos tenham compreendido os conceitos básicos e se sintam confiantes em fazer um programa de computador, sua criação ainda é uma tarefa desafiadora para a maioria, evidenciando a inexperiência da turma e a necessidade de atividades, principalmente atividades práticas, que aprofundem o conhecimento dos alunos e os preparem para a criação de programas mais complexos.

5.3.2 Questões de pesquisa

A fim de compreender melhor a percepção dos alunos sobre o curso, o questionário final incluiu um espaço para que eles expressassem suas opiniões de forma livre e aberta. Essa seção foi fundamental para identificar tanto os aspectos do curso que foram bem recebidos pelos alunos quanto aqueles que podem ser aprimorados, permitindo assim uma análise mais qualitativa da experiência dos participantes.

Para entender o papel do computador no processo de aprendizagem, a pesquisa investigou a frequência com que os alunos o utilizaram durante as aulas (Figura 50). Os resultados demonstram que o computador foi uma ferramenta central para a maioria dos alunos, com 72,7% relatando utilizá-lo por pelo menos 60% do tempo total das aulas. Essa alta taxa de utilização indica a importância do computador como recurso pedagógico do curso.

Figura 50 – Quantidade de alunos por tempo de uso do computador.



Fonte: Elaborado pelo autor (2024).

A análise das respostas dos alunos sobre o que aprenderam durante o curso indicou um aprendizado amplo e positivo. A maioria dos participantes (35 respostas) mencionou a consolidação de conceitos fundamentais de programação, como o uso de variáveis, condicionais e laços de repetição. Além disso, 12 alunos destacaram melhorias em seu raciocínio lógico e na capacidade de resolver problemas de maneira criativa, habilidades essenciais não apenas para a programação, mas para diversas outras áreas do conhecimento. A ferramenta Scratch foi apontada como um recurso valioso, tornando o aprendizado mais divertido e envolvente, especialmente na criação de jogos, conforme destacado em 13 respostas. Por fim, 9 respostas ressaltaram o desenvolvimento do Pensamento Computacional ao longo das aulas. Os relatos a seguir mostram como esses aprendizados foram aplicados e apreciados pelos alunos.

- "Durante o curso eu aprendi a lógica básica da programação, como por exemplo, condicional, operadores, variáveis, loop, etc."
- "Boa parte da lógica da programação, principalmente dentro da plataforma SCRATCH, onde era divertido construir códigos."
- "Programação nem sempre é só programar, é resolver erros e problemas de maneira lógica e criativa. Além de toda a teoria e práticas das aulas."
- "Aprendi a usar o scratch e fazer joguinhos fáceis/histórinhas usando a programação."
- "Eu aprendi a fazer um programa utilizando a ferramenta Scratch de uma maneira divertida, utilizamos praticamente todos os blocos durante as aulas entre eles variáveis, comando

se, senão, diga, pergunte, até a criar o próprio bloco. Aprendi também o que é algoritmo, Pensamento Computacional e muitas outras coisas que vou levar como ensinamento."

- "Foram aulas muito boas que puderam me ensinar sobre algoritmo, Pensamento Computacional, como funciona algumas linguagens de programação e também boa parte da lógica por trás de toda a programação."

A próxima pergunta explora a opinião dos alunos sobre o Pensamento Computacional. Esse conceito é amplamente reconhecido por sua utilidade em diversas disciplinas e na vida cotidiana. Aproximadamente 24 respostas ressaltaram sua importância na resolução de problemas e no desenvolvimento de habilidades de lógica e interpretação, que são aplicáveis em diferentes áreas de estudo. Além disso, 15 alunos expressaram interesse pelo tema, considerando o conteúdo sobre Pensamento Computacional interessante. Embora alguns tenham apontado dificuldades iniciais (cerca de 3 respostas), esses mesmos alunos acreditam que essas dificuldades podem ser superadas e enfatizam que, com prática, o Pensamento Computacional se torna mais acessível e proveitoso. Também houve respostas destacando seus benefícios para o desenvolvimento em programação e para o estímulo da criatividade. Os relatos a seguir ilustram esses pontos.

- "É um pensamento um pouco complicado, mas se bem interpretado pode ajudar muito a compreender outras disciplinas."
- "Achei bastante interessante. O Pensamento Computacional pode ajudar na outras disciplinas fazendo com que o estudante pense fora da caixa para resolver um problema, além de ensiná-lo a interpretar situações para depois conseguir pensar em uma solução."
- "É algo um pouco complicado de entender no começo, mas com um pouco de esforço e prática é possível aprender, e após isso fica muito legal de se trabalhar. Isso irá ajudar durante as aulas técnicas do curso de Desenvolvimento de Sistemas."
- "É uma ótima maneira de resolver problemas. Ajuda tanto na escola quanto na vida cotidiana, pois pode nos ajudar a resolver problemas de maneira lógica, objetiva e funcional, sem muito esforço."
- "Eu achei muito bom, é algo que desenvolve muito a lógica e te possibilita pensar fora da caixa, por isso, pode ser facilmente utilizado em outras disciplinas, ajudando a seguir um caminho mais fácil e prático."

A análise das respostas dos alunos sobre as dificuldades enfrentadas na manipulação do Scratch revela alguns pontos principais. A maioria dos participantes, 19 ao todo, relatou não ter tido dúvidas significativas, indicando que consideraram a ferramenta fácil de usar. No

entanto, 17 respostas destacaram desafios relacionados à lógica de programação, sugerindo que o entendimento dos conceitos lógicos necessários para programar foi uma dificuldade comum no início do curso, o que é um resultado esperado em uma turma com pouca experiência em programação. Em seguida, 7 alunos mencionaram que tiveram dificuldade ao utilizar a interface do Scratch, como localizar os blocos corretos para executar determinadas tarefas, enquanto 6 relataram dificuldades com o manuseio geral da ferramenta. Algumas respostas também citaram as limitações do Scratch, apontando restrições na ferramenta que dificultaram a implementação de ideias mais complexas. Outros relataram a falta de tempo, ressaltando a importância de um ambiente de aprendizagem que permita tempo suficiente para a prática e ofereça suporte adequado. Os relatos a seguir ilustram as dificuldades discutidas.

- "Não, o programa em si é bem fácil de utilizar, além de que todas as dúvidas que eu tive foram tiradas durante as aulas."
- "Não houve dificuldade sobre o entendimento da lógica do programa, no entanto, ele possui limitações que acabam dificultando no desenvolvimento de algumas ideias."
- "Um pouco, a parte da lógica foi difícil de entender no começo, mas após um tempo e um pouco de prática ficou divertido de se trabalhar. As dificuldades em geral foi apenas o tempo."
- "Não tive tantas dificuldades. Scratch é uma ferramenta visual bem simples, mas intuitiva o bastante para aprender, então dificuldades mesmo talvez só na teoria, mas acho que consegui pegar a maior parte da base."
- "As vezes tinha dificuldade de qual bloco usar para executar algo, mas aí fui testando e foi dando certo."
- "As dificuldades foram mais voltadas para como aprender a utilizar a ferramenta, aprender a usar adequadamente os blocos e formar blocos maiores que fossem funcionais."

A análise das respostas dos alunos sobre as dificuldades enfrentadas em conteúdos específicos ou durante alguma aula revelou que a maioria, totalizando 23 respostas, não encontrou dificuldades significativas, indicando que o material foi bem compreendido pela maioria dos participantes. No entanto, houve 13 respostas que destacaram desafios relacionados à lógica de programação, sugerindo que a complexidade do pensamento lógico foi um obstáculo para alguns alunos principalmente em relação aos conteúdos de funções (6 respostas), de variáveis (3 respostas), e de condicionais e laços de repetição (2 respostas), apontando alguma dificuldade em entender e aplicar estes conceitos. Outros pontos mencionados incluem o manuseio do Scratch

como desafiador desafios para 5 estudantes. Assim, embora a maioria dos participantes tenha se sentido confortável, ainda há áreas que exigem mais atenção e prática para alcançar a plena compreensão. A seguir são expostos algumas dos relatos de dificuldades dos alunos.

- "Não senti dificuldades nas aulas."
- "Senti mais dificuldade em funções."
- "Função. Acho que entendi na teoria, mas na prática do Scratch, senti um pouco de dificuldade, já que não consegui executar tão bem quanto esperado."
- "Variáveis. É um conteúdo simples, mas que eu demorei um pouco para entender a mecânica."
- "Nada em específico."

Ao responderem a questão sobre o que mais gostaram de trabalhar durante o curso, as respostas revelaram que a lógica de programação foi o aspecto mais apreciado, com 16 menções, indicando que os conceitos de programação despertaram grande interesse entre os participantes. O uso dos blocos do Scratch também se destacou, sendo mencionado por 15 alunos, o que demonstra uma forte inclinação dos alunos para a prática da programação de forma visual e interativa. Além disso, criar projetos próprios foi outro ponto de destaque, com 11 respostas, mostrando que a liberdade criativa e a oportunidade de desenvolver suas próprias ideias foram valorizadas. Outras áreas mencionadas incluem a manipulação de personagens no Scratch, com 6 respostas, e o ambiente colaborativo, que foi apreciado por 5 alunos, indicando que a interação entre os colegas também contribuiu positivamente para a experiência de aprendizado. Em suma, os aspectos técnicos e práticos, aliados à liberdade criativa, foram os elementos mais apreciados pelos alunos durante o curso. A seguir, são apresentados alguns dos relatos dos alunos sobre o que os alunos mais gostaram de trabalhar.

- "Foi legal entender como funciona as lógicas de programação."
- "Aprender a criar jogos."
- "A construção do trabalho final do curso, onde todas as duplas/trios precisaram criar um projeto para apresentar."
- "Gostei de fazer as tarefas, porque era algo meio livre, a gente podia adicionar mais alguma coisa, mas claro sem perder o foco principal da atividade."
- "Quando era necessário o uso de lógica para a solução de problemas."

A análise das respostas dos alunos sobre o que menos gostaram de trabalhar durante o curso revelou que a maioria, um total de 19 respostas, indicou que não houve reclamação,

sugerindo que grande parte dos alunos não encontrou dificuldades significativas durante as atividades propostas em sala de aula. No entanto, manipular funções foi o aspecto mais mencionado entre os que geraram algum desconforto, com 8 menções, destacando que alguns alunos tiveram dificuldades para trabalhar nesse conceito. As operações aritméticas também foram mencionadas como menos apreciadas, com 5 respostas, indicando desafios na compreensão ou aplicação desses conceitos. Outros pontos, como utilizar repetição e operações lógicas, foram mencionados em menor escala, com 3 e 2 respostas, respectivamente. Embora a maioria dos alunos tenha se sentido confortável com o conteúdo, as áreas de funções e operações matemáticas parecem ter gerado mais desafios. Essa análise pode ser percebida pelas respostas abaixo retiradas do questionário final.

- "Não recordo de algo que não tenha gostado."
- "Sinceramente eu gostei de tudo."
- "Eu gostei menos da parte de funções."
- "A parte de função, foi um pouco chato de aprender."
- "Talvez as operações aritméticas."
- "Quando o desafio era para fazer cálculos."

A análise das respostas dos alunos sobre a metodologia das aulas revelou respostas positivas por quase todos os alunos, com 40 menções, destacando a abordagem de combinação de teoria e prática como divertida e interessante. No entanto, surgiram sugestões para melhorias, principalmente em relação a aumentar do tempo das aulas, com 5 menções, e fornecer explicações mais detalhadas sobre os blocos do Scratch, com 2 menções. Além disso, foram apontadas a necessidade de melhorar o auxílio individual, aumentar o número de exemplos e ajustar o tamanho dos slides, com 2 respostas para cada aspecto. Apesar dessas sugestões, a abordagem dinâmica e interativa das aulas foi amplamente valorizada pelos alunos. Esses aspectos analisados podem ser vistos nos relatos abaixo.

- "Perfeita. Sempre alternando entre lógica e prática, deixando as aulas mais interessantes."
- "Achei bem didático. Em geral foi muito boa, não tenho reclamações."
- "Adorei bastante das aulas. Os monitores eram muito simpáticos e prestativos e as aulas eram bem precisas e interessantes, então não mudaria nada na metodologia, só aumentaria o tempo das aulas."
- "Achei muito boa, só acho que deveria ter tido mais aulas para poder abranger mais assuntos."

- "Foi bom, mas eu melhoraria na especificação de cada ferramenta do scratch."
- "Legal e divertida. Trazer mais aulas assim."
- "Achei muito incrível e dinâmico, houve muita interação com os alunos e pra mim está perfeito, só colocaria uma aula apenas de prática, para facilitar os alunos a se habituarem com o programa."
- "Foi muito boa. Uma boa mistura de prática e teoria o que tornou as aulas bem dinâmicas. Não sei ao certo o que poderia ser melhorado."

Por fim, a última pergunta do questionário final abordou o interesse dos alunos em seguir carreira na área de programação. Os resultados mostraram que 50% dos alunos já tinham a intenção de seguir essa carreira antes do curso, enquanto 20,5% relataram que o interesse foi despertado durante o curso. Além disso, 27,2% dos alunos ainda não têm certeza sobre sua futura carreira, e apenas um participante afirmou que não pretende seguir na área de programação (Figura 51).

Figura 51 – Quantidade de alunos que pretendem seguir carreira na programação.

Quantidade de alunos por interesse em seguir carreira na programação



Fonte: Elaborado pelo autor (2024).

Os resultados indicam que o curso teve um impacto positivo na motivação dos alunos para a programação, com muitos descobrindo ou confirmando seu interesse na área por causa do curso. A alta porcentagem de alunos com intenção de seguir carreira mostra que o curso atende bem às expectativas, enquanto a quantidade de indecisos sugere a necessidade de mais orientação para explorar opções de carreira.

5.3.3 *Análise sobre questionário final*

As respostas dessa etapa são primordiais para responder às questões de pesquisa propostas. De forma geral, o questionário final forneceu informações valiosas sobre a experiência dos alunos durante o curso, especialmente em relação ao uso da ferramenta Scratch para as atividades, ao aprendizado do Pensamento Computacional e ao impacto do curso no desenvolvimento de suas habilidades em lógica de programação. Além disso, serviu como um guia importante para identificar tanto os aspectos positivos quanto as áreas que precisam de melhorias, possibilitando ajustes para futuras aplicações.

A primeira etapa analisou as respostas do modelo de avaliação dTECT, que organiza suas perguntas em três grupos principais de interesse: qualidade da unidade institucional, experiência de computação e percepção da aprendizagem do ponto de vista dos alunos. Os resultados foram amplamente positivos em todas essas áreas.

Os alunos expressaram grande satisfação com o curso, considerando as aulas não apenas divertidas, mas também de excelente qualidade. Eles sentiram que o tempo passou rapidamente, o que indica que o curso conseguiu manter os alunos engajados e interessados durante as aulas.

Além disso, a experiência dos alunos com computação foi satisfatória. Todos destacaram a importância da computação no cotidiano e mostraram entusiasmo ao relatar que acharam a criação de programas de computador divertida.

A percepção geral da aprendizagem também foi positiva, destacando a facilidade que alunos sentiram em relação ao conteúdo proposto e também que eles demonstraram confiança em sua capacidade de criar programas de computador, apesar de que, em sua maioria, eles consideraram essa atividade desafiadora.

Em relação às respostas abertas, houve muitos feedbacks positivos sobre os benefícios que o curso proporcionou aos alunos, porém, também houve relatos de pontos que poderiam ser melhorados.

O Scratch se mostrou uma ferramenta eficaz para engajar os alunos, tornando o aprendizado da programação divertido e motivador. A alta taxa de participação e os feedbacks positivos dos alunos indicam que a plataforma é adequada para despertar o interesse pela computação.

É notável que muitos alunos relataram aprimoramento não apenas em relação aos conceitos de programação, mas também na resolução de problemas, na lógica e na criativi-

dade, habilidades fundamentais promovidas pelo estímulo ao Pensamento Computacional. Isso demonstra que o Scratch é uma ferramenta adequada para desenvolver essas habilidades essenciais, uma vez que é de fácil acesso e utilização, contribuindo para democratizar o ensino de programação.

Muitos alunos destacaram a abordagem prática e lúdica do Scratch, com a criação de projetos próprios durante as aulas e a liberdade para desenvolver o projeto final, incentivando a aprendizagem ativa e a construção autônoma do conhecimento ao mesmo tempo que estimulam o aprendizado coletivo ao desenvolverem seus projetos em duplas ou trios.

Apesar do sucesso geral, a lógica de programação ainda representou um desafio para muitos alunos, especialmente no início do aprendizado. É fundamental que as atividades sejam aplicadas de forma gradual e progressiva, com recursos que auxiliem os alunos a desenvolverem essa habilidade, como desafios para cada novo conceito e o apoio de monitores.

A falta de tempo foi citada como uma dificuldade por alguns alunos. Isso ocorreu principalmente devido ao tempo utilizado nas aulas sobre estruturas de repetição, que, por ser um conteúdo mais complexo, demandou mais tempo do que o planejado, impactando o desenvolvimento de outras atividades e conceitos, como a aplicação da aula de funções.

5.4 Dinâmica final

A dinâmica final ocorreu no encerramento do curso, com o objetivo de incentivar os alunos a exercitarem a criatividade através de projetos próprios, a promoverem a colaboratividade ao desenvolverem esses projetos em equipe e a aplicarem ao máximo os conceitos aprendidos ao longo do curso. No total, 43 alunos participaram, criando e apresentando seus projetos. Apenas um aluno não participou desta etapa, e por isso, seu resultado não foi contabilizado.

5.4.1 Resultados da dinâmica final

As notas, pertencentes a uma escala que varia de 0 a 10, podem ser consultadas na tabela de estatística descritiva da Figura 52. A partir dessa tabela, é possível identificar aspectos importantes relacionados ao desempenho dos alunos nesta atividade, como suas competências técnicas e a capacidade de inovação e criatividade demonstradas em seus projetos.

A análise dos resultados da dinâmica final revela um desempenho satisfatório dos alunos, com uma nota final média de 7,91 e desvio padrão de 0,89, indicando um bom domínio

Figura 52 – Tabela de estatística descritiva sobre as notas do alunos participantes da dinâmica final

Tabela de estatística descritiva sobre as notas dos alunos participantes da dinâmica final

Estatística	Nota CT	Nota IC	Nota Final
Mínimo	5.00	7.00	6.38
Máximo	8.75	9.88	9.31
Média	7.41	8.40	7.91
Moda	8.75	8.25	7.25
Mediana	7.50	8.25	8.06
Desvio Padrão	1.39	0.62	0.89
Variação Interquartil	2.50	0.75	1.44

Fonte: Elaborado pelo autor (2024).

tanto da competência técnica quanto da inovação e criatividade. A menor nota final ficou em 6,38, demonstrando que todos os alunos que participaram da dinâmica atingiram o nível mínimo para aprovação.

A avaliação da Competência Técnica (CT) apresentou uma média de 7,41, evidenciando a boa aplicação dos conteúdos teóricos. No entanto, o desvio padrão de 1,39 sugere uma variabilidade considerável no desempenho dos alunos nessa área. Os valores baixos para a nota mínima (5,00) e máxima (8,75) foram ocasionados, principalmente, pela ausência do conteúdo de funções nos programas apresentados pelos alunos. Provavelmente, isso aconteceu pelo pouco tempo dedicado a esse assunto durante as aulas.

Já a avaliação da Inovação e Criatividade (IC) apresentou resultados mais homogêneos, com média de 8,40 e desvio padrão de 0,62. A moda e a mediana, ambas iguais a 8.25, reforçam a consistência das notas nessa dimensão, indicando que a maioria dos alunos demonstrou um alto nível de originalidade e criatividade em seus projetos.

5.4.2 *Análise comparativa dos resultados da dinâmica*

Ao analisar os resultados da dinâmica final e cruzá-los com os dados obtidos na pergunta "Você tem acesso a computador em casa" do questionário inicial, é possível dividir os dados em dois conjuntos: os resultados dos alunos com acesso a computadores em casa e os resultados dos alunos sem acesso. As comparações entre esses conjuntos podem ser visualizadas na tabela da Figura 53.

Figura 53 – Tabela de estatística descritiva sobre as notas do alunos em relação ao acesso a computador

Tabela de estatística descritiva sobre a nota final dos alunos em relação ao acesso a computador em casa

Estatística	Com acesso a computador	Sem acesso a computador
Mínimo	6.38	6.38
Máximo	9.31	8.81
Média	8.08	7.65
Mediana	8.21	7.75
Moda	8.94	7.25
Desvio Padrão	0.94	0.73
Variação Interquartil	1.59	0.87

Fonte: Elaborado pelo autor (2024).

Os resultados da análise comparativa das notas dos alunos com e sem acesso a um computador em casa revelaram diferenças significativas no desempenho da dinâmica. Os alunos que possuíam um computador em casa demonstraram um desempenho superior, com médias e medianas de notas finais mais elevadas. Essa diferença sugere que o acesso a um computador pessoal pode ser um fator determinante para o sucesso na aprendizagem de programação nesse contexto.

Ao analisar os dados com mais profundidade, observou-se que os alunos com computador apresentaram uma maior variabilidade nas notas, indicando que o acesso à tecnologia pode ter permitido uma maior exploração de diferentes níveis de complexidade no projeto.

Os relatos a seguir, de várias equipes, sobre as dificuldades enfrentadas durante a dinâmica ilustram essas implicações, pois alguns membros tinham acesso a um computador em casa, enquanto outros não.

- "Organização em relação as variáveis, pois eram tantas que as vezes a variável ia pro lugar errado. Acessibilidade, nem todos os participantes tinham computador ou notebook em casa, o que dificultou um pouco o entendimento de todos os participantes."
- "Pouco tempo para realizar o trabalho, a falta de um computador/notebook na casa de um dos membros da dupla e pensar em uma forma de elaborar e construir o código."
- "As dificuldades encontradas durante a organização do nosso projeto foram as seguintes: falta de acesso a computadores, dificuldade em compreender os códigos, complexidade da concepção e desafios na implementação das funções."

Esses resultados corroboram a hipótese de que o acesso a um computador pessoal

pode proporcionar mais oportunidades de aprendizagem e desenvolvimento de habilidades de programação. Assim, os resultados encontrados neste estudo possuem importantes implicações para a educação em programação. A desigualdade no acesso a recursos tecnológicos, como computadores, pode gerar disparidades no desempenho dos alunos e, conseqüentemente, limitar as oportunidades de desenvolvimento de habilidades digitais.

5.5 Considerações finais

Os resultados da pesquisa revelaram um perfil heterogêneo da turma em relação ao acesso a recursos tecnológicos, o que impactou significativamente o aprendizado de programação. A análise dos dados destacou pontos fortes e fracos da intervenção pedagógica e traçou um panorama dos desafios e oportunidades no ensino de programação com Scratch.

A turma, predominantemente masculina, demonstrou interesse em matemática e um bom nível de conhecimento em informática, características esperadas para um curso de Desenvolvimento de Sistemas. No entanto, observou-se uma variação significativa no acesso a computadores pessoais, o que influenciou a experiência dos alunos com a programação.

Os dados do questionário DETECT mostraram alta satisfação com o curso, elogiando a metodologia e a relevância dos conteúdos. A percepção dos alunos quanto à sua aprendizagem foi positiva, refletindo confiança nas habilidades de programação adquiridas.

As respostas abertas corroboraram esses resultados, destacando a importância do Scratch como ferramenta de ensino para promover o desenvolvimento do Pensamento Computacional, com a maioria dos participantes relatando melhorias em habilidades de resolução de problemas, lógica e criatividade. A abordagem prática e a liberdade para criar projetos próprios foram especialmente valorizadas pelos alunos, o que estimulou a aplicação prática dos conceitos e o desenvolvimento de um raciocínio computacional mais elaborado.

A oportunidade de desenvolver projetos próprios durante a dinâmica final proporcionou um ambiente propício para a exploração da criatividade e a consolidação dos conhecimentos adquiridos ao longo do curso. No entanto, os resultados obtidos evidenciaram que o acesso a um computador pessoal pode ser um fator determinante para o desempenho dos alunos, uma vez que aqueles que possuíam essa ferramenta apresentaram, em média, melhores resultados.

Portanto, o Scratch se mostrou uma excelente ferramenta para introduzir conceitos básicos de programação de forma lúdica e intuitiva, aumentando a motivação e o engajamento dos alunos. No entanto, a pesquisa identificou desafios, como a dificuldade inicial em compreender

conceitos de programação e a necessidade de mais tempo para explorar profundamente alguns tópicos. Recomenda-se oferecer um suporte individualizado com material especializado e desenvolver ainda mais atividades cooperativas, estimulando o compartilhamento de conhecimento e a colaboração entre os alunos.

Os resultados abrem oportunidades para investigar o impacto do Scratch em diferentes contextos educativos, como escolas com recursos limitados ou com alunos de diversas idades, além de explorar o impacto a longo prazo no desenvolvimento profissional dos alunos.

Esses resultados podem orientar professores que utilizam o Scratch em suas aulas, destacando a importância de um ambiente colaborativo e desafiador que permita aos alunos explorar ideias e construir conhecimento. É essencial que os professores estejam preparados para oferecer suporte adequado e adaptar as atividades às necessidades dos alunos.

6 CONCLUSÃO

6.1 Principais contribuições

Este trabalho abordou os conceitos fundamentais do ensino do Pensamento Computacional e sua relevância para a educação. Foram apresentadas diversas ferramentas que facilitam o ensino da lógica de programação por meio de gamificação e ambientes de programação lúdicos e simplificados, com destaque principal para o *software* Scratch.

A pesquisa desenvolvida desempenhou um papel fundamental para o levantamento de estratégias utilizadas para o estímulo do Pensamento Computacional em diferentes etapas do ensino. A metodologia apresentada teve sua concentração no ensino médio, visto seu potencial para aprendizado do raciocínio lógico e dos benefícios do estímulo dessa habilidade nesta fase da educação.

Neste contexto, esta pesquisa objetivou analisar o uso do Scratch como ferramenta para ensino de Pensamento Computacional, identificar os desafios e as dificuldades no aprendizado de programação visual com Scratch e investigar a relação entre o acesso a computadores e o desempenho em programação.

Para atingir esses objetivos, foi conduzido um experimento com a turma do primeiro ano do curso de Desenvolvimento de Sistemas de uma escola de ensino médio profissionalizante, em parceria com o projeto LearningLab da Universidade Federal do Ceará, o orientador deste trabalho e o autor deste estudo. O experimento foi realizado na forma de um curso de programação com o objetivo de ensinar lógica de programação por meio da ferramenta Scratch e promover o Pensamento Computacional.

Essa abordagem é fundamentada em materiais validados do Programe_CE e nos conhecimentos adquiridos durante a pesquisa para aprimorar a experiência do ensino do Pensamento Computacional, permitindo metrificar os resultados e benefícios obtidos de sua aplicação.

Os dados foram coletados por meio das atividades realizadas pelos alunos durante o curso e de questionários que exploravam aspectos socioeconômicos, experiências anteriores com programação, percepções sobre o aprendizado durante o curso, entre outras questões.

Foi possível observar a evolução do Pensamento Computacional dos alunos ao longo do curso. A capacidade de criar projetos cada vez mais complexos e de solucionar problemas de forma lógica e criativa demonstra o potencial do Scratch como ferramenta para desenvolver essas habilidades.

Os resultados também destacaram como o Scratch contribuiu para despertar o interesse dos alunos pela programação, através de seu ambiente visual de programação por blocos. A abordagem lúdica para questões fundamentais em programação possibilitou o desenvolvimento de atividades colaborativas e desafiadoras, tornando o aprendizado mais envolvente e eficaz.

No entanto, a pesquisa revelou áreas que precisam de melhorias. A duração do curso foi considerada insuficiente para o desenvolvimento completo dos conceitos de repetição e funções, além da necessidade de um acompanhamento mais próximo dos alunos, especialmente daqueles com menor familiaridade com a tecnologia.

Além disso, a pesquisa evidenciou que a desigualdade no acesso a computadores pessoais afetou significativamente o desempenho dos alunos. Aqueles com acesso a um computador em casa demonstraram maior familiaridade com a ferramenta e obtiveram melhores resultados, enquanto a falta de acesso a computadores fora do ambiente escolar limitou a prática da programação e pode ter comprometido a compreensão aprofundada dos conceitos abordados no curso. Essa disparidade ressalta a importância de políticas públicas e iniciativas educacionais que visem democratizar o acesso à tecnologia, garantindo que todos os alunos tenham oportunidades iguais de desenvolver habilidades digitais.

6.2 Trabalhos futuros

Os resultados desta pesquisa abrem portas para novas investigações sobre o uso do Scratch em diferentes níveis de ensino e em diversas áreas do conhecimento assim como a investigação do impacto do Scratch no desenvolvimento de outras habilidades, como a comunicação e o trabalho em equipe. Além disso, estudos futuros podem explorar a utilização do Scratch em conjunto com outras ferramentas e metodologias de ensino, visando otimizar o processo de aprendizagem.

Além disso, é fundamental aprofundar o estudo sobre o impacto da desigualdade digital no processo de ensino-aprendizagem de programação. Pesquisas futuras devem investigar as diversas formas de desigualdade digital e seus efeitos na educação, bem como o desenvolvimento de metodologias e ferramentas pedagógicas que promovam a inclusão de todos os alunos, independentemente de suas condições socioeconômicas.

Portanto, este trabalho visa não apenas compartilhar os resultados deste estudo dedicado a implementação da ferramenta Scratch para a promoção do Pensamento Computacional, mas também apresenta um modelo de metodologia aplicável, proporcionando uma base

para análise e reprodução futura. Assim, considera-se que foram atingidos os objetivos desse estudo, de contribuir para o desenvolvimento contínuo dessa área de pesquisa, incentivando a disseminação de práticas eficazes no ensino do Pensamento Computacional.

REFERÊNCIAS

- BARR, D.; HARRISON, J.; CONERY, L. Computational thinking: A digital age skill for everyone. **Learning and leading with technology**, Association for Computing Machinery, v. 38, p. 20–23, 2011. ISSN ISSN-1082-5754.
- BRACKMANN, C. P. **Desenvolvimento do pensamento computacional através de atividades desplugadas na educação básica**. Tese (Doutorado) – Universidade Federal do Rio Grande do Sul - UFRGS, Rio Grande do Sul - Brasil, 2017.
- BRENNAN, K.; BALCH, C.; CHUNG, M. **Creative Computing - Learner Workbook**. CreateSpace Independent Publishing Platform, 2014. ISBN 9781503388079. Disponível em: https://books.google.com.br/books?id=Y7_VrQEACAAJ.
- CAVALCANTE NETO, J. B.; BONATES, M. F.; BONATES, T. de Oliveira e; FILHO, J. M. da S. M.; FARIAS, E. J. P.; LIRA, J. H. S. de; OLIVEIRA, A. G. L. S. **Programa_Ce: Fundamentos de Programação Convencional - Módulo 2**. Fortaleza - Ceará - Brasil: Secretaria da Educação do Estado do Ceará - SEDUC/CE and Fundação Cearense de Apoio ao Desenvolvimento Científico - FUNCAP, 2021. ISBN 978-65-00-36596-2. Disponível em: <https://www.seduc.ce.gov.br/wp-content/uploads/sites/37/2022/01/Modulo2.pdf>.
- Code.org. **Code.org**. 2023. Disponível em: <https://code.org/>.
- CSIZMADIA, A.; CURZON, P.; DORLING, M.; HUMPHREYS, S.; NG, T.; SELBY, C.; WOOLLARD, J. Computational thinking - a guide for teachers. **BCS - The Chartered Institute for TI**, Computing At School, v. 1, 2015.
- FISHER, D. E.; FISHER, M. **Tube: The Invention of Television**. Harcourt Brace, 1997. (A Harvest book). ISBN 9780156005364. Disponível em: <https://books.google.com.br/books?id=Pbt2QgAACAAJ>.
- GROVER, S.; PEA, R. Computational thinking in k–12: A review of the state of the field. **Educational Researcher**, v. 42, n. 1, p. 38–43, 2013.
- LAVE, J.; WENGER, E. **Situated Learning: Legitimate Peripheral Participation**. [S. l.]: Cambridge University Press, 1991. (Learning in Doing: Social, Cognitive and Computational Perspectives). ISBN 9780511815355.
- LEARNING, B. **What is computational thinking?** 2023. Disponível em: <http://www.bbc.co.uk/education/guides/zp92mp3/revision>.
- LEARNINGLAB. **LearningLab - Quem Somos**. 2023. Disponível em: <https://learninglab.com.br/sobre/>.
- LINS, B. F. E. A evolução da internet : uma perspectiva histórica. **Cadernos Aslegis**, Associação dos Consultores Legislativos e de Orçamento e Fiscalização Financeira da Câmara dos Deputados, n. 48, p. 11–46, 2013. ISSN 1677-9010. Disponível em: <http://bd.camara.leg.br/bd/handle/bdcamara/33179>.
- LIUKAS, L. **Hello Ruby: Adventures in Coding**. Helsinki - Finlândia: Feiweil & Friends, 2015. (Hello Ruby). ISBN 9781250065001. Disponível em: <https://books.google.com.br/books?id=I2yACgAAQBAJ>.

Ministério da Educação; BNCC. **Computação na Educação Básica - Complemento à BNCC**. 2022. Disponível em: http://basenacionalcomum.mec.gov.br/images//historico/anexo_parecer_cneceb_n_2_2022_bncc_computacao.pdf.

Ministério da Educação; CAPES. **Portal de Periódicos da CAPES**. 2020. Disponível em: <https://www.periodicos.capes.gov.br/>.

MOTA, F. P.; RIBEIRO, N. F. A.; EMMENDORFER, L.; BUTZEN, P.; MACHADO, K. S.; ADAMATTI, D. F. Desenvolvendo o raciocínio lógico no ensino médio: uma proposta utilizando a ferramenta scratch. **Anais do XXV Simpósio Brasileiro de Informática na Educação (SBIE 2014)**, Centro de Ciências Computacionais – Universidade Federal do Rio Grande (FURG), p. 377–381, 2014.

PETZOLD, C. **Code: The Hidden Language of Computer Hardware and Software**. Microsoft Press, 1999. (DV-MPS General Series). ISBN 9780735605053. Disponível em: <https://books.google.com.br/books?id=pHcfAQAIAAJ>.

PRODANOV, C. C.; FREITAS, E. C. de. **Metodologia do Trabalho Científico: Métodos e Técnicas da Pesquisa e do Trabalho Acadêmico**. Novo Hamburgo - Rio Grande do Sul - Brasil: EDITORA FEEVALE, 2013. v. 2. 49-54 p. ISBN 978-85-7717-158-3. Disponível em: <https://www.feevale.br/institucional/editora-feevale/metodologia-do-trabalho-cientifico---2-edicao>.

RIBAS, E.; BIANCO, G. D.; LAHM, R. A. Programação visual para introdução ao ensino de programação na educação superior: uma análise prática. **Revista Novas Tecnologias na Educação**, Portal de Periódicos Científicos UFRGS, v. 14, n. 2, p. –, 2016.

RODRIGUES, E. V.; SOUZA, L. D. de. Instituto de hackers: o pensamento computacional aplicado ao ensino médio integrado profissionalizante. **INFORMÁTICA NA EDUCAÇÃO teoria prática.**, Portal de Periódicos Científicos UFRGS, v. 24, n. 1, p. 115–128, 2021.

SCRATCH. **Página de criação do Scratch**. 2023. Disponível em: <https://scratch.mit.edu/projects/editor/>.

SCRATCH, F. **Sobre o Scratch**. 2023. Disponível em: <https://scratch.mit.edu/about>.

SEDUC/CE, S. da Educação do Estado do C.; FUNCAP, F. C. de Apoio ao D. C. **Programa_Ce**. 2021. Disponível em: https://www.seduc.ce.gov.br/2021/08/20/programe_ce/.

SHIMASAKI, R.; PRADO, M. E. B. B. O ensino da programação e o desenvolvimento do pensamento lógico: uma revisão sistemática de literatura. **Revista de Ensino, Educação e Ciências Humanas**, Portal de Periódicos Científicos da Cogna, v. 22, n. 2, p. 197–205, 2021.

VIANA, G. A.; PORTELA, C. dos S. O uso de softwares educativos para introdução de lógica de programação no ensino de base e superior. **Revista Softwares Educativos, Ensino de Ciências, Metodologias Ativas (2019)**, Informática na Educação: teoria & prática, v. 22, n. 1, p. 10–22, 2019.

VICARI, R. M.; MOREIRA Álvaro; MENEZES, P. B. **Pensamento computacional : revisão bibliográfica**. Rio Grande do Sul - Brasil: Universidade Federal do Rio Grande do Sul - UFRGS, 2018. Disponível em: <http://hdl.handle.net/10183/197566>.

WANGENHEIM, C. Gresse von; PETRI, G.; ZIBETTI, A. W.; HAUCK, J.; PACHECO, F. S.; FILHO, R. detect: Um modelo para a avaliação de unidades instrucionais para o ensino de computação na educação básica. **Informatics in Education**, Vilnius University Institute of Data Science and Digital Technologies, v. 16, n. 2, p. 301–318, 2017. ISSN 1648-5831.

WING, J. M. Computational thinking. **Communications of the ACM**, Association for Computing Machinery, v. 49, p. 33–35, 2006.

APÊNDICE A – SLIDES UTILIZADOS NAS AULAS

Neste apêndice, são apresentadas as informações referentes às apresentações de slides desenvolvidas para auxiliar na ministração das aulas do projeto. Foram elaboradas cinco apresentações de slides, elaboradas utilizando o software Canva, incorporando imagens obtidas do Google Imagens e capturas de tela da ferramenta Scratch para ilustrar diversos conceitos abordados.

Antes de serem utilizadas em sala de aula, as apresentações de slides passaram por um processo de validação, sendo revisadas e aprovadas pelo professor orientador e pela coordenadora do projeto Learninglab. Esse procedimento garantia a qualidade e a adequação do conteúdo apresentado aos alunos durante as atividades pedagógicas.

A inclusão deste apêndice visa fornecer um registro adicional do material didático utilizado no desenvolvimento das aulas, contribuindo para a compreensão e contextualização do método de ensino adotado no projeto.



AVLA 1

INTRODUÇÃO

CURSO DE INTRODUÇÃO À PROGRAMAÇÃO UTILIZANDO SCRATCH!

LEARNINGLAB

MARÇO 2024

MEU NOME É ROBÉRIO.



Sou estudante da UFC Campus de Russas e estou no último semestre do curso de **Ciência da Computação.**



Atualmente faço parte de um projeto chamado **LearningLab** atuando no setor de extensão.

LEARNINGLAB

MARÇO 2024

POR QUE FAZER O CURSO?

EXPLORAR O
SCRATCH

APRENDER
LÓGICA DE
PROGRAMAÇÃO

DESENVOLVER O
PENSAMENTO
COMPUTACIONAL

CRIAR NOSSOS
PRÓPRIOS
PROGRAMAS E JOGOS

CONTRIBUIR COM
UMA PESQUISA
CIENTÍFICA

GARANTIR UM
CERTIFICADO DE
PARTICIPAÇÃO

LEARNINGLAB

MARÇO 2024

FACULDADE E MERCADO DE TRABALHO

O QUE É
PROGRAMAÇÃO?

O QUE É
PENSAMENTO
COMPUTACIONAL?

ONDE EU POSSO
UTILIZAR ISSO?

POR ONDE EU
POSSO COMEÇAR
A APRENDER?

COMO VOCÊ ACHA
QUE É TRABALHAR
COM PROGRAMAÇÃO?

O QUE VOCÊ ACHA
SOBRE A FACULDADE
DE COMPUTAÇÃO?

LEARNINGLAB

MARÇO 2024

COMO VÃO FUNCIONAR NOSSAS AULAS?

2 MOMENTOS



AULA TEÓRICA:

abordaremos o conteúdo em forma de slide apresentando o que será trabalhado em sala



AULA PRÁTICA:

realizaremos alguma atividade relacionada com o conteúdo visto no primeiro momento da aula



LEARNINGLAB

MARÇO 2024

AVALIAÇÕES



1. ATIVIDADES REALIZADAS DURANTE AS AULAS

Todas as atividades desenvolvidas em cada aula contam como nota para a nota final

2. ATIVIDADE FINAL

A atividade final será um desafio utilizando todos os conhecimentos que foram adquiridos durante as aulas

3. FORMULÁRIOS

Todos os formulários preenchidos contribuem para a pesquisa científica e para a melhora dos cursos

LEARNINGLAB

PENSAMENTO COMPUTACIONAL



"uma forma de solucionar problemas, projetar sistemas e compreender comportamentos humanos, aproveitando conceitos fundamentais de ciência da computação"

Jeannette M. Wing (2006)

LEARNINGLAB

PENSAMENTO COMPUTACIONAL



DECOMPOSIÇÃO

Dividir o desafio em problemas menores para facilitar a compreensão.

ABSTRAÇÃO

Reconhecer o que é mais importante na situação-problema e deixar de lado o que não for essencial.

4 PILARES



RECONHECIMENTO DE PADRÕES

Identificar as repetições e similaridades dos problemas para auxiliar na resolução.

ALGORITMO

Propor uma ordem ou uma sequência de passos para resolver o problema.

LEARNINGLAB

DECOMPOSIÇÃO

Para exemplificar podemos tomar como referência uma bicicleta. Ela é **uma peça** criada para cumprir um determinada função mas **contém diversas peças** que separadamente cumprem uma função específica.



LEARNINGLAB

RECONHECIMENTO DE PADRÕES

Podemos observar os cachorros. Existem diversas raças, de diferentes tamanhos e cores mas todos são cachorros. Todos têm uma **função em comum**, agem de uma maneira em comum.



LEARNINGLAB

ABSTRAÇÃO

Um manequim tem a função de **representar** o corpo humano mas, não necessariamente todas as suas características, **abstraído os detalhes** e focando apenas no essencial para sua função.



LEARNINGLAB

ALGORITMO

Uma receita de bolo é como um guia. Seu modo de preparo é um **passo a passo** detalhado do que se deve fazer para chegar no objetivo final: fazer um bolo.

Ingredientes:

- * 3 ovos
- * 3 colheres de sopa de margarina
- * 1 e 1/2 xícaras de açúcar (xícara de 240 ml)
- * 2 xícaras de farinha de trigo
- * 2 colheres de sopa de leite em pó integral instantâneo
- * 1 xícara de água
- * 1 colher de sopa de fermento em pó

Modo de preparo:

1. Bater na batedeira os ovos, margarina e açúcar até ficar um creme mais claro
2. Acrescentar metade da farinha de trigo, leite em pó e a água, bater muito bem
3. Adicionar o restante da farinha e bater bem a massa
4. Acrescentar o fermento e misturar
5. Colocar em forma untada com margarina
6. Assar em forno médio até dourar, e colocando um palito ele saia seco
7. Espere esfriar para desenformar



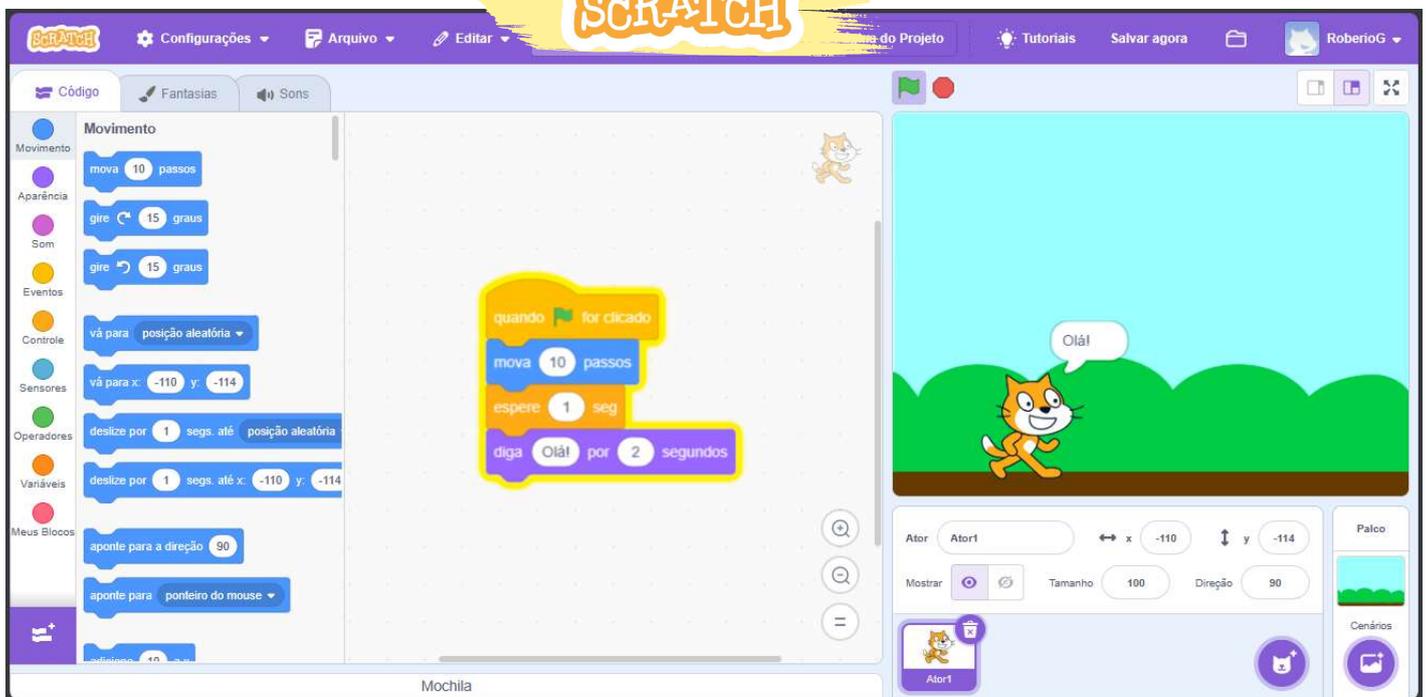
LEARNINGLAB

SCRATCH

Scratch é uma linguagem de programação com uma interface visual simples que permitem que as pessoas criem histórias, jogos e animações digitais..

O Scratch promove o pensamento computacional e habilidades de resolução de problemas; ensino e aprendizagem criativos; autoexpressão e colaboração; e equidade em computação.

LEARNINGLAB



LEARNINGLAB



LEARNINGLAB

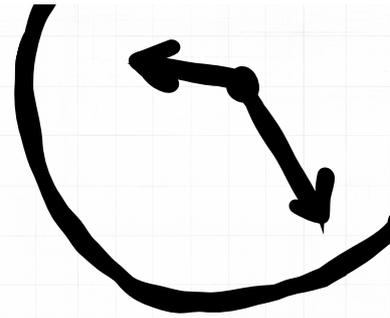


CURSO DE INTRODUÇÃO À PROGRAMAÇÃO UTILIZANDO SCRATCH!

LEARNINGLAB

CRONOGRAMA

O QUE VAMOS FAZER HOJE?

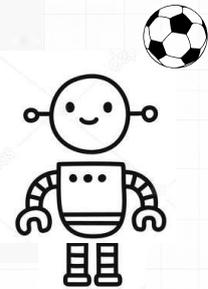
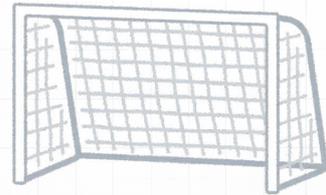


FASE 1	FASE 2	FASE 3	FASE 4	FASE 5
Programação sequencial	Comandos de entrada e saída	Variáveis	Expressões Aritméticas e Relacionais	Desafio de programação

LEARNINGLAB

PROGRAMAÇÃO SEQUENCIAL

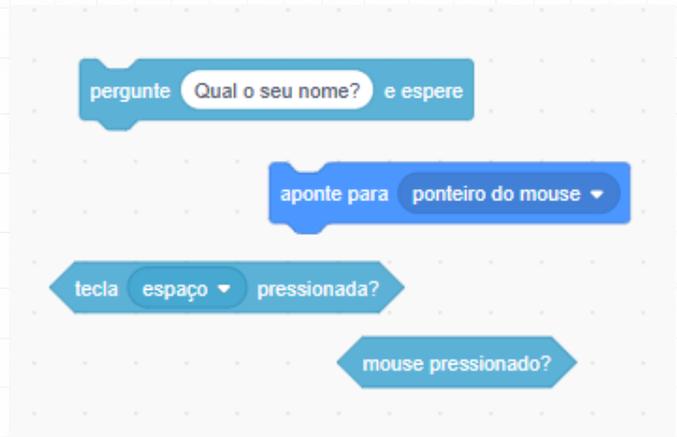
É um paradigma de programação que envolve a **execução de instruções em uma ordem linear**, uma após a outra. Como numa receita, onde se deve seguir uma sequência de passos para fazer um bolo.



LEARNINGLAB

COMANDOS DE ENTRADA

Para fazer um bolo, você precisa de ingredientes, certo? A entrada de dados é toda informação que o usuário fornece ao computador, tal como a receita precisa dos ingredientes e qual sua quantidade para fazer a receita.



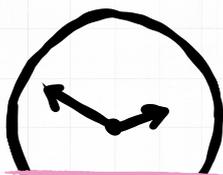
LEARNINGLAB

COMANDOS DE SAÍDA

Assim como temos entrada de dados, temos as saídas de dados. Estas são todas as informações que o programa retorna para o usuário de alguma forma, geralmente por uma tela ou um dispositivo de áudio.



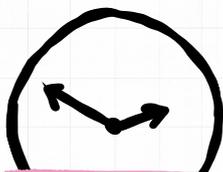
LEARNINGLAB



HORA DE PRATICAR

Faça um programa em que o gato pergunte "Qual o seu nome?" utilizando comandos de entrada e responda com "É um prazer te conhecer" utilizando comandos de saída.

LEARNINGLAB



HORA DE PRATICAR

quando for clicado

pergunte Qual o seu nome? e espere

diga É um prazer te conhecer por 5 segundos

LEARNINGLAB

VARIÁVEIS

Variáveis em um programa de computador são **espaços de memória** designados para armazenar dados. Elas são utilizadas para armazenar valores como números, texto, booleanos (verdadeiro ou falso), entre outros tipos de dados.



LEARNINGLAB

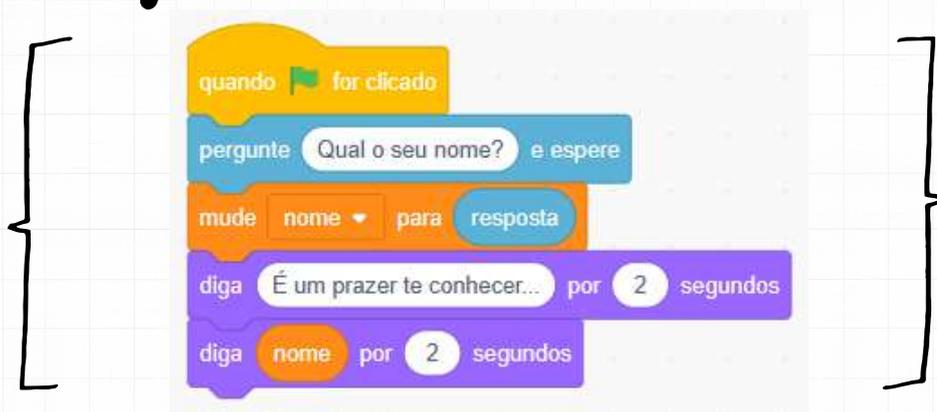
HORA DE PRATICAR

Faça um programa em que o gato pergunte "Qual o seu nome?" utilizando comandos de entrada, **salve sua resposta** em uma variável e responda com "É um prazer te conhecer..." e em seguida **diga o nome salvo** utilizando comandos de saída.

LEARNINGLAB



HORA DE PRATICAR



LEARNINGLAB

EXPRESSÕES

1. ARITMÉTICAS

Envolvem operações matemáticas

2. RELACIONAIS

Envolvem operadores de comparação para determinar a relação entre dois valores

3. LÓGICAS

Também chamados de booleanos, envolvem operadores lógicos, como AND, OR e NOT.

LEARNINGLAB

EXPRESSÕES

1. ARITMÉTICAS

$3 + 5;$
 $2 * 4 - 3;$
 $6 / 3 + (2 - 2)$
(resultam em um valor numérico)

2. RELACIONAIS

$10 > 20;$
 $x \leq y;$
 $idade = 18$
(produzem um resultado verdadeiro ou falso)

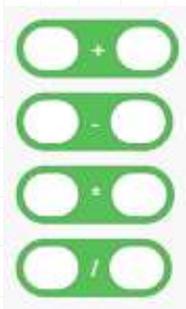
3. LÓGICAS

$altura > 170 \text{ AND } peso < 70;$
 $idade > 18 \text{ OR } autorizacao = True;$
 $NOT (choveu)$
(produzem um resultado verdadeiro ou falso)

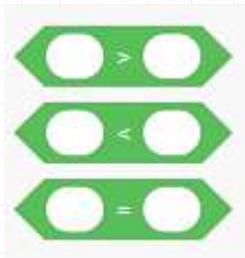
LEARNINGLAB

EXPRESSÕES

1. ARITMÉTICAS



2. RELACIONAIS



3. LÓGICAS



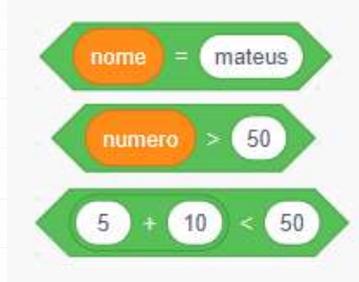
LEARNINGLAB

EXPRESSÕES

1. ARITMÉTICAS



2. RELACIONAIS



3. LÓGICAS



LEARNINGLAB

DESAFIO 1

Faça um programa em que o gato resolva uma conta simples de matemática seguindo esse algoritmos:

- O gato pergunta um número;
- O número é salvo
- O gato pergunta outro número;
- O número é salvo
- O gato responde o resultado da soma dos dois números;



LEARNINGLAB

DESAFIO 1

```
quando for clicado
pergunte Me diga um número e espere
mude numero1 para resposta
pergunte Me diga outro número e espere
mude numero2 para resposta
mude resultado para numero1 + numero2
diga A soma dos dois números é... por 2 segundos
diga resultado por 2 segundos
```

LEARNINGLAB

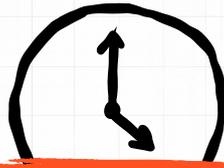
DESAFIO EXTRA

Faça um programa em que o gato calcule a média de suas notas seguindo esse algoritmos:

- O gato pergunta a primeira nota;
- A nota é salva;
- O gato pergunta a segunda nota;
- A nota é salva
- O gato pergunta a terceira nota;
- A nota é salva
- O gato responde o resultado da média das três notas;



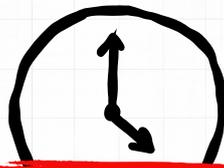
LEARNINGLAB



DESAFIO EXTRA

```
quando for clicado
pergunte Diga qual a primeira nota e espere
mude Nota1 para resposta
pergunte Diga qual a segunda nota e espere
mude Nota2 para resposta
pergunte Diga qual a terceira nota e espere
mude Nota3 para resposta
mude resultado para Nota1 + Nota2 + Nota3 / 3
diga resultado por 5 segundos
```

LEARNINGLAB

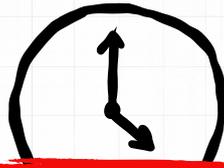


DESAFIO HARD

Faça um programa em que o gato converta a temperatura de celsius para fahrenheit seguindo esse algoritmos:

- O gato pergunta a temperatura em celsius;
- A temperatura é salva
- O gato calcula a conversão para fahrenheit;
- O gato responde o resultado do calculo;

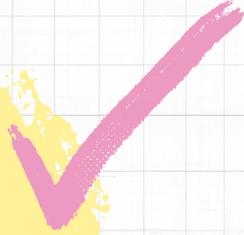
LEARNINGLAB



DESAFIO HARD

```
quando for clicado
pergunte Qual a temperatura? e espere
mude celsius para resposta
mude fahrenheit para  $celsius * 9 / 5 + 32$ 
diga fahrenheit por 2 segundos
```

LEARNINGLAB


OBRIGADO  

LEARNINGLAB



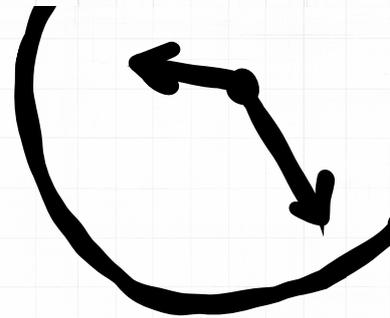
AVULA 3

ESTRUTURA DE DECISÃO

CURSO DE INTRODUÇÃO À PROGRAMAÇÃO UTILIZANDO SCRATCH!

LEARNINGLAB

CRONOGRAMA
O QUE VAMOS FAZER HOJE?



FASE 1

Estrutura
de decisão

FASE 2

Comandos de
decisão no
Scratch

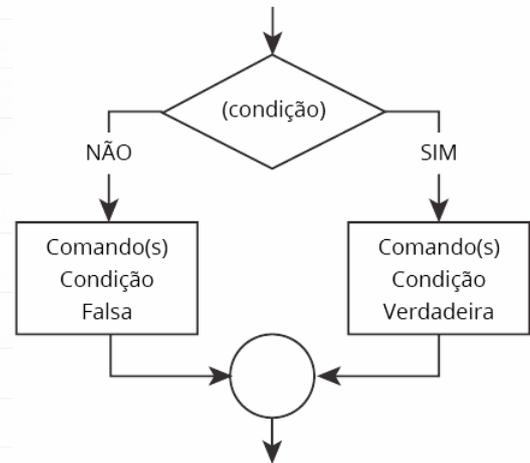
FASE 3

Desafio de
programação

LEARNINGLAB

COMANDOS DE DECISÃO

Comandos de decisão são instruções que permitem que um programa escolha entre diferentes caminhos de execução com base em uma **condição específica**. Eles são fundamentais para controlar o fluxo de um programa e **tomar decisões lógicas**.



LEARNINGLAB

TESTES DE DECISÃO

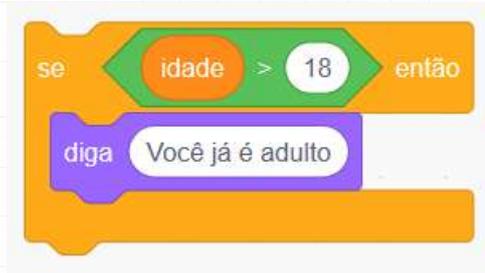
Esta condição geralmente envolve operadores relacionais (como igualdade, desigualdade, maior que, menor que, etc.) e/ou operadores lógicos (como AND, OR, NOT).



LEARNINGLAB

COMANDOS SE... ENTÃO

O comando **se... então** é usado para executar um bloco de código se uma condição específica for verdadeira.

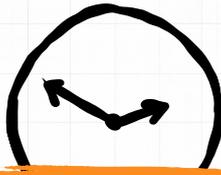


LEARNINGLAB

HORA DE PRATICAR

Faça um programa no qual se a tecla espaço estiver pressionada, o gato dá um pulinho.

LEARNINGLAB



HORA DE PRATICAR



LEARNINGLAB

COMANDO SE ... ENTÃO ... SENÃO

O comando **se... então... se não** é usado para executar diferentes blocos de código se uma condição específica for verdadeira.



LEARNINGLAB

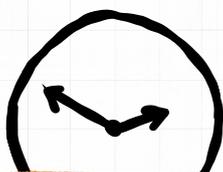


HORA DE PRATICAR

Faça um programa em que o gato responda se um número é par ou ímpar:

- O gato pergunta um número;
- O número é salvo;
- Se o número for par o gato responde "O número é par"
- Se o número for ímpar o gato responde "O número é ímpar"

LEARNINGLAB



HORA DE PRATICAR

```
quando for clicado
pergunte Diga um número e espere
mude numero para resposta
se resto de numero por 2 = 0 então
diga O número é par por 2 segundos
senão
diga O número é ímpar por 2 segundos
```

LEARNINGLAB

COMANDOS DE DECISÃO ANINHADOS

Nem sempre conseguimos representar todas as opções de decisão apenas com um bloco de comando de decisão, por isso podemos **aninhar blocos de decisão** para verificar várias condições.



LEARNINGLAB



DESAFIO 2

Faça um programa em que o gato responda qual número é maior:

- O gato pergunta o primeiro número;
- O gato pergunta o segundo número;
- Salva os dois números;
- **Se o primeiro número for maior** o gato responde "O primeiro número é maior"
- **Se o segundo número for maior** o gato responde "O segundo número é maior"
- **Se os dois números forem iguais** o gato responde "Os números são iguais"

LEARNINGLAB



DESAFIO 2

```
quando for clicado
pergunte Diga qual o primeiro número e espere
mude numero1 para resposta
pergunte Diga qual o segundo número e espere
mude numero2 para resposta
se numero1 > numero2 então
diga O primeiro número é maior por 2 segundos
senão
se numero1 < numero2 então
diga O segundo número é maior por 2 segundos
senão
diga Os números são iguais por 2 segundos
```

LEARNINGLAB



OBRIGADO

LEARNINGLAB



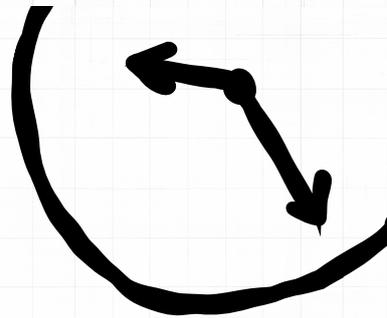
AVULA 4

ESTRUTURA DE REPETIÇÃO

CURSO DE INTRODUÇÃO À PROGRAMAÇÃO UTILIZANDO SCRATCH!

LEARNINGLAB

CRONOGRAMA
O QUE VAMOS FAZER HOJE?



FASE 1

Estrutura de
repetição

FASE 2

Comandos de
repetição
no Scratch

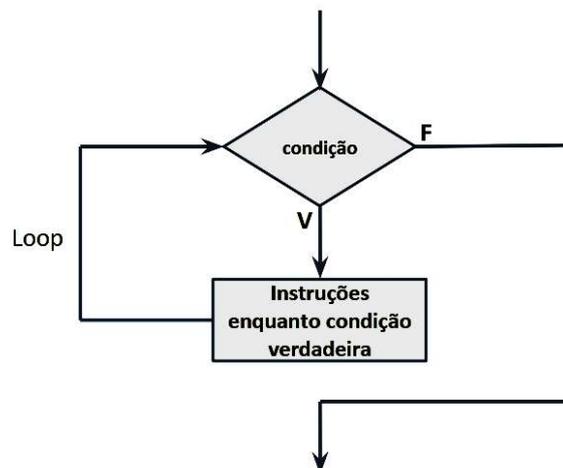
FASE 3

Desafio de
programação

LEARNINGLAB

COMANDOS DE REPETIÇÃO

Comandos de repetição são estruturas de controle que permitem executar um bloco de código repetidamente enquanto uma condição específica for verdadeira. Eles são fundamentais para automatizar tarefas que exigem repetição de operações semelhantes.



LEARNINGLAB

TESTES DE DECISÃO

Para lembrar, os testes de condição são os mesmos vistos nas últimas aulas, envolvendo operadores relacionais e/ou operadores lógicos.



LEARNINGLAB

COMANDO REPITA ... VEZES

O comando **repita ... vezes** é usado para executar um bloco de código quando se sabe exatamente o número de repetições que se precisa.



LEARNINGLAB

HORA DE PRATICAR

Faça um programa em que o gato conte de 1 até 100.

LEARNINGLAB



HORA DE PRATICAR

```
quando for clicado
  muda numero para 0
  repita 100 vezes
    muda numero para numero + 1
    diga numero por 0.5 segundos
```

LEARNINGLAB

COMANDO REPITA ATÉ QUE ...



O comando **repita até que ...** é usado quando o número de repetições não é conhecido. Assim, o bloco de repetição deve continuar até uma condição ser atendida.

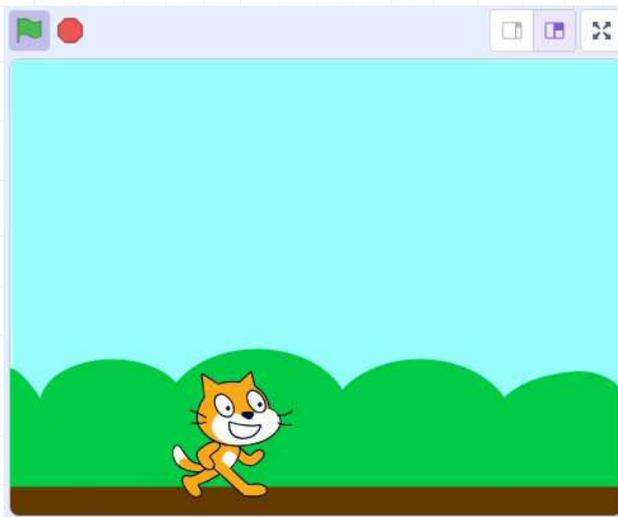
```
repita até que numero = 100
  muda numero para numero + 1
  diga numero por 0.5 segundos
```

LEARNINGLAB

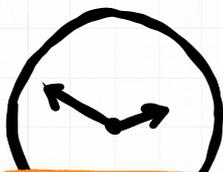


HORA DE PRATICAR

Faça um programa em que o gato ande de uma ponta a outra do cenário utilizando o comando de repetição "repita até que ..."



LEARNINGLAB



HORA DE PRATICAR



LEARNINGLAB

COMANDO SEMPRE

O comando **sempre** é usada para repetir um bloco de comando indefinidamente e só parar quando é pressionado o botão de parada do programa. Casos assim na programação convencional são chamados de Laço Infinito.



LEARNINGLAB

COMANDOS DE REPETIÇÃO ANINHADOS

Pode-se combinar vários tipos de repetição ao **aninhar blocos de repetição**. É usado para ter mais opções de repetição de comandos dentro de um mesmo bloco.

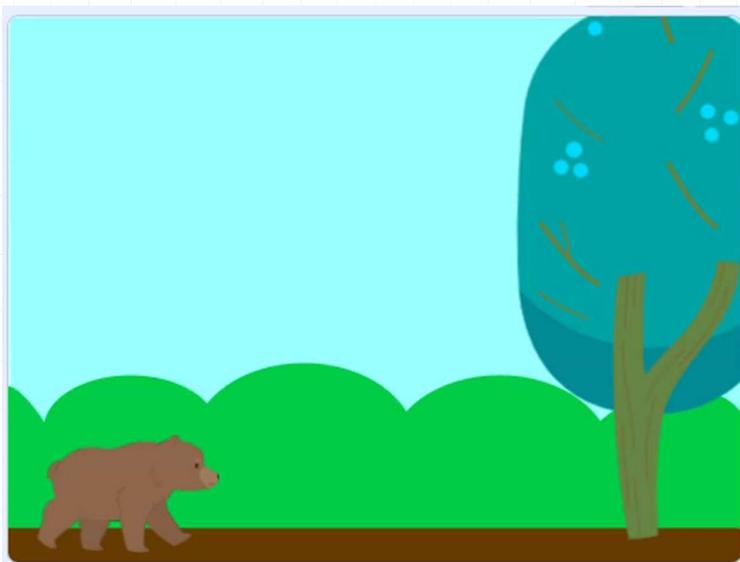


LEARNINGLAB



DESAFIO 3

Faça um programa utilizando o cenário "Blue Sky" e o personagem "Bear Walk" e uma árvore para reproduzir essa cena:



LEARNINGLAB



DESAFIO 3

```
quando for clicado
  vá para x: -170 y: -130
  repita até que tocando em Trees ?
  repita 10 vezes
    mova 5 passos
    próxima fantasia
    espere 0.1 seg
  diga Finalmente uma sombra, ufa por 2 segundos
```

LEARNINGLAB



LEARNINGLAB

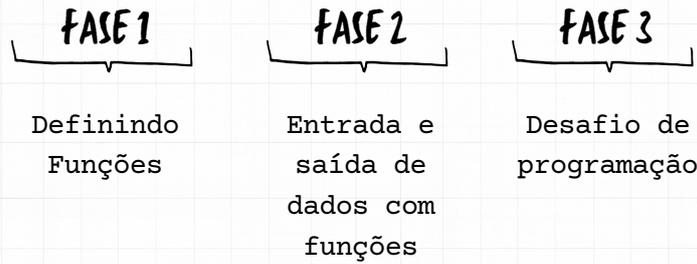
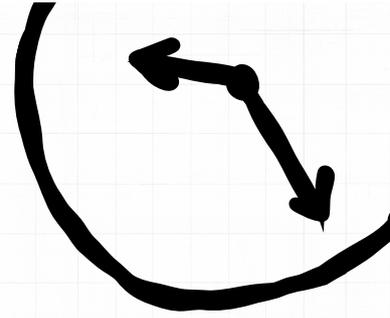


CURSO DE INTRODUÇÃO À PROGRAMAÇÃO UTILIZANDO SCRATCH!

LEARNINGLAB

CRONOGRAMA

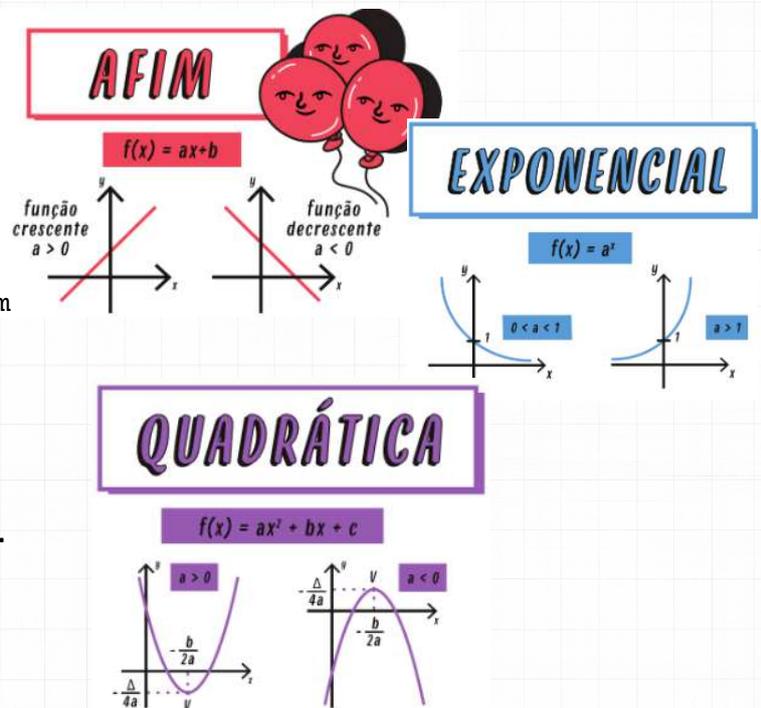
O QUE VAMOS FAZER HOJE?



LEARNINGLAB

FUNÇÕES

Funções são blocos de código que realizam uma tarefa específica e podem ser usadas em diferentes partes de um programa para executar essa tarefa. Elas permitem modularizar o código, o que facilita a compreensão, a manutenção e a reutilização do código.



LEARNINGLAB

FUNÇÕES NO SCRATCH

No Scratch, temos os blocos disponibilizados pelo próprio Scratch, que chamamos de **Funções Internas da Linguagem** (Built-in Functions).

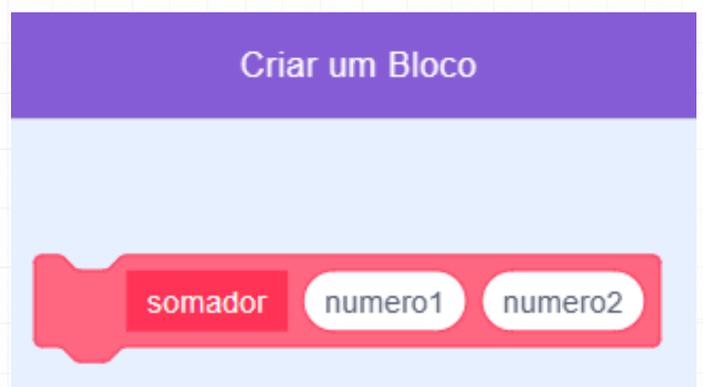
Também temos as **Funções Definidas pelo Usuário**, que são aquelas que criamos para executar tarefas específicas.



LEARNINGLAB

CRIANDO FUNÇÕES

Para criar funções no Scratch, utilizamos a área "Meus Blocos". Nela, criamos o cabeçalho da função com o nome da função e, caso necessário, entradas de dados que podem ser números ou texto.



LEARNINGLAB

CRIANDO FUNÇÕES

Ao criar o cabeçalho, um campo dentro da área de trabalho é disponibilizado para definir o que a função deve fazer. Após criar sua função, ela pode ser usada com o seu personagem do Scratch para executar sua tarefa, da mesma forma que os blocos nativos.



LEARNINGLAB



Crie uma função que calcula o IMC de uma pessoa e salva seu resultado em uma variável.

$$\text{IMC} = \frac{\text{peso}}{\text{altura}^2}$$

IMC	Classificações
Menor do que 18,8	Abaixo do peso normal
18,5 - 24,9	Peso normal
25,0 - 29,9	Excesso de Peso
30,0 - 34,9	Obesidade Classe I
35,0 a - 39,9	Obesidade Classe II
Maior ou Igual a 40,0	Obesidade Classe III

LEARNINGLAB



DESAFIO 4.1

Crie uma função que tem como entrada valores para A, B e C.
A função calcula o delta de uma equação do segundo grau,
guarda esse valor e mostra se a equação tem duas, uma ou
nenhuma solução no conjunto dos reais.

Delta
 $ax^2 + bx + c = 0$
 $\Delta = b^2 - 4 \cdot a \cdot c$

$\Delta > 0$	A equação terá duas soluções reais diferentes.
$\Delta = 0$	A equação terá duas soluções reais iguais.
$\Delta < 0$	A equação não terá solução nos reais.

$a = 2, b = 8 \text{ e } c = -24 \quad \Delta = 256$

LEARNINGLAB



DESAFIO 4.1

```
defina delta a b c
mude delta para b * b - 4 * a * c
se delta > 0 então
diga A equação possui duas raízes reais diferentes por 2 segundos
se delta = 0 então
diga A equação possui duas raízes reais iguais por 2 segundos
se delta < 0 então
diga A equação não possui solução no conjunto dos reais por 2 segundos
```

LEARNINGLAB



DESAFIO 4.2

Agora que você criou a função para calcular o delta, crie outra função que utilize a fórmula de Bhaskara para resolver uma equação do segundo grau, utilizando o resultado do delta para calcular e armazenar os valores de x1 e x2.

$$X = \frac{-b \pm \sqrt{b^2 - 4.a.c}}{2.a}$$

$$a = 2, b = 8 \text{ e } c = -24 \quad \Delta = 256$$

$$x' = \frac{-8 - 16}{4} = \frac{-24}{4} = -6$$

$$x'' = \frac{-8 + 16}{4} = \frac{8}{4} = 2$$

LEARNINGLAB



DESAFIO 4.2

```
defina bhaskara a b c
delta a b c
mude x1 para 0 - b + raiz quadrada de delta / 2 * a
mude x2 para 0 - b - raiz quadrada de delta / 2 * a
```

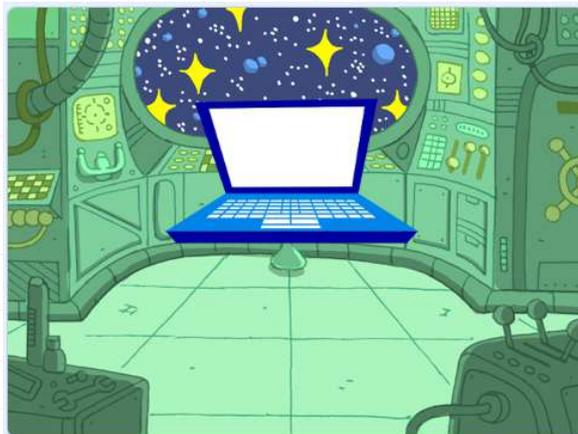
LEARNINGLAB



DESAFIO 4.3

Agora que criamos duas funções para calcular uma equação do segundo grau, vamos criar o programa que usa elas.

Crie um programa que pede ao usuário os valores de a , b e c de uma equação, utiliza eles nas funções para calcular seu resultado e depois mostra o resultado



LEARNINGLAB



DESAFIO 4.3

```
quando for clicado
diga Olá, eu sou um computador que calcula equação do segundo grau. por 2 segundos
pergunte Qual o valor de a? e espere
mude a para resposta
pergunte Qual o valor de b? e espere
mude b para resposta
pergunte Qual o valor de c? e espere
mude c para resposta
bhaskara a b c
diga junte O valor de x1 é com x1 por 2 segundos
diga junte O valor de x2 é com x2 por 2 segundos
```

LEARNINGLAB



APÊNDICE B – PROPOSTA DE DINÂMICA PARA ATIVIDADE FINAL

Neste apêndice, são apresentadas as informações referentes a proposta de dinâmica que foi utilizada como atividade final durante as aulas do curso de Introdução À Programação Utilizando Scratch.

Antes de serem utilizadas em sala de aula, a proposta de atividade passou pelo processo de validação, sendo revisadas e aprovadas pelo professor orientador e pela coordenadora do projeto Learninglab. Esse procedimento garantia a qualidade e a adequação do conteúdo apresentado aos alunos durante as atividades pedagógicas.

A inclusão deste apêndice visa fornecer um registro adicional do material didático utilizado no desenvolvimento das aulas, contribuindo para a compreensão e contextualização do método de ensino adotado no projeto.

Proposta de Dinâmica para a Última Aula:

Na última aula, propõe-se uma **dinâmica de concurso**, na qual os alunos serão organizados em **equipes de dois ou três** integrantes e desafiados a criar um **programa ou jogo que aplique os conceitos aprendidos ao longo do curso**.

Detalhes da Dinâmica:

O projeto:

- O projeto deve ser desenvolvido utilizando a ferramenta scratch.
- O projeto deve ter um objetivo claro e resolver uma problemática.
- O projeto deve englobar o máximo de conteúdos vistos em sala.
- O projeto precisa ser entregue juntamente com um pequeno relatório.
- O projeto deve ser compartilhado na área de projetos da sua conta do scratch
- Pode-se utilizar quaisquer materiais para consulta como chatgpt, youtube e apostilas de estudos para desenvolver o projeto, não se limitando ao que foi visto em sala de aula, contudo, não é permitido a cópia de outros trabalhos.

Elaboração do Relatório:

Os alunos devem produzir um relatório contemplando os seguintes pontos:

1. Ideias propostas para a criação do programa ou jogo.
2. Descrição do que foi efetivamente entregue (funcionalidades e funcionamento do programa/jogo).
3. Dificuldades encontradas durante a elaboração do projeto (como a falta de acesso a computadores, a complexidade da ideia, dificuldades na implementação com o Scratch, ou restrições de tempo).
4. Link para o projeto disponibilizado na plataforma Scratch

Exemplo relatório:

Equipe: Roberio e Oirebor.

Proposta: Jogo da Velha.

Funções: O projeto trata de criar um jogo da velha onde dois jogadores jogam contra. Ao clicar na bandeirinha os usuários colocam seus nomes e depois podem jogar clicando onde querem colocar o X ou O. As regras seguem as mesmas do jogo da velha convencional, ganha quem conseguir preencher uma coluna, uma linha ou uma diagonal com X ou O, e caso nenhum consiga isso, o jogo termina empatado.

Dificuldades: Não consegui adicionar um contador de vitórias e não consegui entender como criar uma função de menu.

Link: <https://scratch.mit.edu/projects/13969717>

Prazos:

A proposta deve ser apresentada na aula do dia 02 de maio (quinta-feira).

Os alunos terão uma semana para desenvolver e executar seus projetos, devendo entregá-los na aula seguinte, do dia 08 de maio (quarta-feira).

Apresentação:

- A apresentação deve durar no máximo 5 minutos.
- Os alunos devem apresentar o projeto funcionando.
- Os alunos devem descrever qual a problemática que o projeto resolve.
- Os alunos devem descrever quais os principais blocos que utilizaram para resolver a problemática.

Avaliação dos Projetos:

Os projetos serão submetidos à avaliação dos monitores, juntamente com o tutor, seguindo os seguintes critérios:

- O projeto contou com quais conceitos aprendidos nas aulas?
 - entrada de dados
 - saída de dados
 - variáveis
 - operações aritméticas
 - operações lógicas
 - estrutura de decisão
 - estrutura de repetição
 - funções
- Avaliação do cumprimento da proposta inicial do projeto.
- Atribuição de nota técnica dos monitores e do tutor.

Premiação:

Após a apresentação dos projetos pelos alunos, os monitores e o tutor se reunirão para atribuir notas e estabelecer um ranking dos três melhores projetos.

- 1º lugar: Uma caixa de chocolate para a equipe, um copo e um broche para cada integrante.
- 2º lugar: Um copo, um broche e um kit de participação para cada integrante.
- 3º lugar: Um broche e um kit de participação para cada integrante.
- Demais alunos participantes: Kit de participação.

Com esta dinâmica, busca-se promover a aplicação prática dos conhecimentos adquiridos, estimulando a criatividade, colaboração, o desenvolvimento do pensamento computacional e das habilidades em programação.

APÊNDICE C – RESULTADOS COLETADOS

O presente apêndice contém os dados completos e agrupados obtidos através dos questionários inicial e final, bem como da dinâmica final realizada. A fim de garantir o anonimato dos participantes e o cumprimento das normas éticas, os nomes foram removidos, porém a identificação única de cada participante foi mantida para permitir a correlação entre os diferentes conjuntos de dados.

A inclusão deste apêndice visa fornecer um registro detalhado e transparente dos dados utilizados nas análises, possibilitando sua reutilização em futuras pesquisas, análises complementares ou validações.

Sexo	Idade:	Você já reprovou alguma vez?	Se respondeu sim para a pergunta anterior, quantas vezes você reprovou?	Qual a renda mensal da sua família?	Qual o nível de escolaridade do seu pai?	Qual o nível de escolaridade da sua mãe?
Prefero não informar	15	Não		Até 01 salário mínimo	Da 1ª à 4ª série do Ensino Fundamental (antigo primário).	Ensino Médio (antigo 2º grau).
Feminino	15	Não		Mais que 02 e até 05 salários mínimos	Ensino Médio (antigo 2º grau).	Ensino Médio (antigo 2º grau).
Masculino	15	Não		Mais que 02 e até 05 salários mínimos	Ensino Superior.	Ensino Médio (antigo 2º grau).
Masculino	15	Não		Mais que 02 e até 05 salários mínimos	Não sei.	Pós-graduação (Especialização, Mestrado ou Doutorado).
Feminino	15	Não		Mais que 01 e até 02 salários mínimos	Não sei.	Da 5ª à 8ª série do Ensino Fundamental (antigo ginásio).
Masculino	15	Não		Mais que 01 e até 02 salários mínimos	Da 5ª à 8ª série do Ensino Fundamental (antigo ginásio).	Ensino Superior.
Masculino	15	Não		Mais que 01 e até 02 salários mínimos	Ensino Médio (antigo 2º grau).	Ensino Médio (antigo 2º grau).
Masculino	15	Não		Mais que 01 e até 02 salários mínimos	Ensino Médio (antigo 2º grau).	Ensino Médio (antigo 2º grau).
Masculino	15	Não		Mais que 01 e até 02 salários mínimos	Ensino Médio (antigo 2º grau).	Ensino Médio (antigo 2º grau).
Feminino	14	Não		Mais que 01 e até 02 salários mínimos	Ensino Médio (antigo 2º grau).	Ensino Médio (antigo 2º grau).
Masculino	17	Sim	2	Até 01 salário mínimo	Não sei.	Da 5ª à 8ª série do Ensino Fundamental (antigo ginásio).
Masculino	15	Não		Até 01 salário mínimo	Ensino Superior.	Ensino Médio (antigo 2º grau).
Masculino	15	Não		Mais que 01 e até 02 salários mínimos	Da 5ª à 8ª série do Ensino Fundamental (antigo ginásio).	Ensino Médio (antigo 2º grau).
Feminino	15	Não		Mais que 01 e até 02 salários mínimos	Ensino Médio (antigo 2º grau).	Ensino Superior.
Masculino	15	Não		Mais que 02 e até 05 salários mínimos	Ensino Médio (antigo 2º grau).	Ensino Médio (antigo 2º grau).
Masculino	15	Não		Mais que 01 e até 02 salários mínimos	Ensino Médio (antigo 2º grau).	Pós-graduação (Especialização, Mestrado ou Doutorado).
Masculino	15	Não		Mais que 01 e até 02 salários mínimos	Ensino Médio (antigo 2º grau).	Ensino Médio (antigo 2º grau).
Masculino	15	Não		Mais que 02 e até 05 salários mínimos	Da 5ª à 8ª série do Ensino Fundamental (antigo ginásio).	Ensino Médio (antigo 2º grau).
Feminino	15	Não		Até 01 salário mínimo	Da 1ª à 4ª série do Ensino Fundamental (antigo primário).	Da 5ª à 8ª série do Ensino Fundamental (antigo ginásio).
Feminino	15	Não		Mais que 05 e até 08 salários mínimos	Pós-graduação (Especialização, Mestrado ou Doutorado).	Ensino Superior.
Feminino	15	Não		Mais que 01 e até 02 salários mínimos	Não sei.	Da 1ª à 4ª série do Ensino Fundamental (antigo primário).
Masculino	15	Não		Mais que 05 e até 08 salários mínimos	Pós-graduação (Especialização, Mestrado ou Doutorado).	Ensino Médio (antigo 2º grau).
Feminino	15	Não		Até 01 salário mínimo	Da 5ª à 8ª série do Ensino Fundamental (antigo ginásio).	Ensino Médio (antigo 2º grau).
Masculino	14	Não		Mais que 01 e até 02 salários mínimos	Não sei.	Ensino Médio (antigo 2º grau).
Masculino	15	Não		Mais que 01 e até 02 salários mínimos	Ensino Médio (antigo 2º grau).	Ensino Médio (antigo 2º grau).
Masculino	15	Não		Mais que 02 e até 05 salários mínimos	Pós-graduação (Especialização, Mestrado ou Doutorado).	Pós-graduação (Especialização, Mestrado ou Doutorado).
Feminino	15	Não		Mais que 01 e até 02 salários mínimos	Ensino Médio (antigo 2º grau).	Da 5ª à 8ª série do Ensino Fundamental (antigo ginásio).
Feminino	15	Não		Mais que 02 e até 05 salários mínimos	Ensino Médio (antigo 2º grau).	Ensino Médio (antigo 2º grau).
Feminino	15	Não		Mais que 02 e até 05 salários mínimos	Não sei.	Ensino Médio (antigo 2º grau).
Feminino	15	Não		Mais que 01 e até 02 salários mínimos	Da 5ª à 8ª série do Ensino Fundamental (antigo ginásio).	Ensino Médio (antigo 2º grau).
Masculino	15	Não		Mais que 01 e até 02 salários mínimos	Da 1ª à 4ª série do Ensino Fundamental (antigo primário).	Da 5ª à 8ª série do Ensino Fundamental (antigo ginásio).
Masculino	15	Não		Até 01 salário mínimo	Ensino Médio (antigo 2º grau).	Ensino Médio (antigo 2º grau).
Feminino	14	Não		Mais que 08 salários mínimos	Ensino Médio (antigo 2º grau).	Ensino Médio (antigo 2º grau).
Feminino	15	Não		Até 01 salário mínimo	Da 5ª à 8ª série do Ensino Fundamental (antigo ginásio).	Ensino Médio (antigo 2º grau).
Masculino	15	Não		Mais que 01 e até 02 salários mínimos	Da 1ª à 4ª série do Ensino Fundamental (antigo primário).	Ensino Superior.
Masculino	15	Não		Mais que 02 e até 05 salários mínimos	Pós-graduação (Especialização, Mestrado ou Doutorado).	Ensino Superior.
Masculino	15	Não		Mais que 02 e até 05 salários mínimos	Ensino Médio (antigo 2º grau).	Ensino Médio (antigo 2º grau).
Feminino	15	Não		Mais que 02 e até 05 salários mínimos	Ensino Médio (antigo 2º grau).	Ensino Superior.
Masculino	15	Não		Mais que 02 e até 05 salários mínimos	Da 1ª à 4ª série do Ensino Fundamental (antigo primário).	Pós-graduação (Especialização, Mestrado ou Doutorado).
Masculino	15	Sim	1	Mais que 01 e até 02 salários mínimos	Da 5ª à 8ª série do Ensino Fundamental (antigo ginásio).	Da 5ª à 8ª série do Ensino Fundamental (antigo ginásio).
Masculino	15	Não		Mais que 02 e até 05 salários mínimos	Ensino Médio (antigo 2º grau).	Pós-graduação (Especialização, Mestrado ou Doutorado).
Masculino	15	Não		Mais que 01 e até 02 salários mínimos	Não sei.	Ensino Médio (antigo 2º grau).
Masculino	15	Não		Mais que 02 e até 05 salários mínimos	Ensino Médio (antigo 2º grau).	Da 5ª à 8ª série do Ensino Fundamental (antigo ginásio).
Masculino	14	Não		Mais que 01 e até 02 salários mínimos	Ensino Médio (antigo 2º grau).	Ensino Médio (antigo 2º grau).

Reside em Russas?	Qual principal transporte utiliza para chegar à escola?	Quanto tempo em média demora para chegar à escola?	Tempo médio chegar a escola	Tem acesso a computador em casa?	Tem acesso a internet em casa?	Qual seu nível de conhecimento em informática básica?	Você gosta de matemática?
Sim	Ônibus escolar.	20 minutos	Entre 10 min e 30 min	Não.	Sim.	Básico - tenho conhecimentos gerais, mas quase nunca uso;	Sim.
Sim	De carona com pais ou amigos (carro/moto)	5 minutos de moto ou 20 minutos a pé	Entre 5 min e 10 min	Não.	Sim.	Mínimo - uso raramente;	Sim.
Sim	Bicicleta.	6min	Entre 5 min e 10 min	Sim.	Sim.	Básico - tenho conhecimentos gerais, mas quase nunca uso;	Sim.
Sim	De carona com pais ou amigos (carro/moto)	10 minutos	Entre 5 min e 10 min	Sim.	Sim.	Nenhum - não tenho contato com o computador;	Não.
Sim	Ônibus escolar.	20 minutos	Entre 10 min e 30 min	Não.	Sim.	Básico - tenho conhecimentos gerais, mas quase nunca uso;	Não.
Sim	A pé.	10 minutos	Entre 5 min e 10 min	Não.	Sim.	Avançado - tenho um conhecimento profundo e costumo usar diariamente.	Não.
Sim	Ônibus escolar.	5 min	Entre 5 min e 10 min	Sim.	Sim.	Mínimo - uso raramente;	Sim.
Sim	De carona com pais ou amigos (carro/moto)	5 - 10 minutos	Entre 5 min e 10 min	Sim.	Sim.	Básico - tenho conhecimentos gerais, mas quase nunca uso;	Sim.
Não	Ônibus escolar.	1h	Entre 30 min e 1 hora	Sim.	Sim.	Básico - tenho conhecimentos gerais, mas quase nunca uso;	Sim.
Sim	Ônibus escolar.	10 a 20 minutos	Entre 10 min e 30 min	Não.	Sim.	Mínimo - uso raramente;	Sim.
Sim	Bicicleta.	9min	Entre 5 min e 10 min	Não.	Sim.	Mínimo - uso raramente;	Não.
Sim	De carona com pais ou amigos (carro/moto)	6 minutos.	Entre 5 min e 10 min	Não.	Sim.	Básico - tenho conhecimentos gerais, mas quase nunca uso;	Sim.
Sim	Bicicleta.	7 minutos	Entre 5 min e 10 min	Não.	Sim.	Básico - tenho conhecimentos gerais, mas quase nunca uso;	Sim.
Não	Ônibus escolar.	40 minutos de ida e volta.	Entre 30 min e 1 hora	Sim.	Sim.	Básico - tenho conhecimentos gerais, mas quase nunca uso;	Sim.
Não	Ônibus escolar.	40min	Entre 30 min e 1 hora	Sim.	Sim.	Intermediário - tenho conhecimento médio e já sei usar sem dificuldades;	Sim.
Sim	Bicicleta.	7 minutos	Entre 5 min e 10 min	Sim.	Sim.	Básico - tenho conhecimentos gerais, mas quase nunca uso;	Sim.
Sim	De carona com pais ou amigos (carro/moto)	4 minutos	Entre 5 min e 10 min	Não.	Sim.	Mínimo - uso raramente;	Sim.
Sim	Bicicleta.	7 minutos	Entre 5 min e 10 min	Sim.	Sim.	Intermediário - tenho conhecimento médio e já sei usar sem dificuldades;	Sim.
Sim	Ônibus escolar.	30 minutos	Entre 10 min e 30 min	Não.	Sim.	Mínimo - uso raramente;	Não.
Não	Ônibus escolar.	1 hora de vinda, e uma de volta	Entre 30 min e 1 hora	Sim.	Sim.	Intermediário - tenho conhecimento médio e já sei usar sem dificuldades;	Sim.
Sim	Ônibus escolar.	30 a 40 minutos	Entre 30 min e 1 hora	Sim.	Sim.	Mínimo - uso raramente;	Sim.
Sim	De carona com pais ou amigos (carro/moto)	10 minutos	Entre 5 min e 10 min	Sim.	Sim.	Básico - tenho conhecimentos gerais, mas quase nunca uso;	Sim.
Não	Ônibus escolar.	30 minutos	Entre 10 min e 30 min	Não.	Sim.	Mínimo - uso raramente;	Sim.
Sim	Bicicleta.	7 minutos	Entre 5 min e 10 min	Não.	Sim.	Básico - tenho conhecimentos gerais, mas quase nunca uso;	Sim.
Sim	De carona com pais ou amigos (carro/moto)	7 minutos	Entre 5 min e 10 min	Sim.	Sim.	Intermediário - tenho conhecimento médio e já sei usar sem dificuldades;	Sim.
Sim	Ônibus escolar.	20 Minutos.	Entre 10 min e 30 min	Sim.	Sim.	Mínimo - uso raramente;	Sim.
Sim	Bicicleta.	15/20 minutos	Entre 10 min e 30 min	Sim.	Sim.	Intermediário - tenho conhecimento médio e já sei usar sem dificuldades;	Sim.
Sim	Vou de ônibus e às vezes de moto.	30min	Entre 10 min e 30 min	Não.	Sim.	Básico - tenho conhecimentos gerais, mas quase nunca uso;	Sim.
Sim	A pé.	15 minutos	Entre 10 min e 30 min	Sim.	Sim.	Básico - tenho conhecimentos gerais, mas quase nunca uso;	Sim.
Não	Ônibus escolar.	15 minutos	Entre 10 min e 30 min	Sim.	Sim.	Básico - tenho conhecimentos gerais, mas quase nunca uso;	Sim.
Sim	De carona com pais ou amigos (carro/moto)	10 minutos	Entre 5 min e 10 min	Sim.	Sim.	Básico - tenho conhecimentos gerais, mas quase nunca uso;	Sim.
Sim	Ônibus e as vezes carro	50min	Entre 30 min e 1 hora	Sim.	Sim.	Intermediário - tenho conhecimento médio e já sei usar sem dificuldades;	Não.
Sim	De carona com pais ou amigos (carro/moto)	10 minutos	Entre 5 min e 10 min	Sim.	Sim.	Básico - tenho conhecimentos gerais, mas quase nunca uso;	Sim.
Sim	Ônibus escolar.	15 minutos	Entre 10 min e 30 min	Não.	Sim.	Básico - tenho conhecimentos gerais, mas quase nunca uso;	Não.
Sim	De carona com pais ou amigos (carro/moto)	10 Minutos	Entre 5 min e 10 min	Sim.	Sim.	Intermediário - tenho conhecimento médio e já sei usar sem dificuldades;	Não.
Sim	Ônibus escolar.	10 minutos	Entre 5 min e 10 min	Sim.	Sim.	Básico - tenho conhecimentos gerais, mas quase nunca uso;	Sim.
Sim	Ônibus escolar.	40 minutos	Entre 30 min e 1 hora	Não.	Sim.	Intermediário - tenho conhecimento médio e já sei usar sem dificuldades;	Sim.
Sim	De carona com pais ou amigos (carro/moto)	10 minutos	Entre 5 min e 10 min	Sim.	Sim.	Básico - tenho conhecimentos gerais, mas quase nunca uso;	Sim.
Sim	Ônibus escolar.	5 minutos	Entre 5 min e 10 min	Sim.	Sim.	Intermediário - tenho conhecimento médio e já sei usar sem dificuldades;	Sim.
Sim	Ônibus escolar.	15 minutos	Entre 10 min e 30 min	Sim.	Sim.	Intermediário - tenho conhecimento médio e já sei usar sem dificuldades;	Não.
Sim	Ônibus escolar.	40 minutos	Entre 30 min e 1 hora	Sim.	Sim.	Mínimo - uso raramente;	Sim.
Sim	moto e bike	6 minutos	Entre 5 min e 10 min	Não.	Sim.	Básico - tenho conhecimentos gerais, mas quase nunca uso;	Sim.
Não	Ônibus escolar.	40 minutos	Entre 30 min e 1 hora	Não.	Sim.	Intermediário - tenho conhecimento médio e já sei usar sem dificuldades;	Sim.
Sim	Ônibus escolar.	Meia hora	Entre 10 min e 30 min	Não.	Sim.	Básico - tenho conhecimentos gerais, mas quase nunca uso;	Sim.

Justifique a resposta anterior.	O que você espera aprender nesse curso?	Você já teve algum contato com programação?	Se respondeu sim para a pergunta anterior, qual tipo de contato, conte sobre sua experiência.
Por que eu amo exatas		Sim.	
é uma matéria que eu tenho muita afinidade	o básico sobre lógica de programação	Não.	
tirei 9,4 pai	pretendo aprender o básico sobre lógica de programação	Sim.	programando nos computadores da escola
não mais consigo passar	programação	Sim.	um pouco de python
Tenho uma grande dificuldade na matéria, pois os números e letras sempre estão se invertendo na minha mente, ou seja, eles ficam mudando de posição.	O básico sobre programação com o schart.	Não.	
Não me identifico com a matéria	Aprender um novo modo de programação	Sim.	tenho contato nas aulas do curso técnico na escola
Gosto bastante, acho interessante	Lógica de Programação	Não.	
Porque gosto de resolver problemas e chegar em resultados concretos.	Obter uma visão do uso básico de programação.	Sim.	Conheço o básico de programação com um pouco de conhecimento em Python, Html e Css.
Por dar um resultado exato	Como programar	Não.	
Acho interessante	Logica de programação	Não.	
não costumo gostar de calculos	programação de forma mais facil	Não.	
É uma área que me chama a atenção, principalmente por não gostar da área das humanas. Me dou bem em matemática e gosto de praticar.	Aprender melhor como utilizar variáveis e outros recursos da programação, para ter um melhor desempenho durante o curso de Desenvolvimento de Sistemas.	Sim.	Apenas uma aula prática do curso de Desenvolvimento de Sistemas, mas que foi bastante interessante e divertido aprender sobre, mesmo não sabendo muito como programar ou algo relacionado.
porque tenho facilidade com cálculos	ganhar experiencia	Não.	não
Desde o fundamental gosto da matéria, me interesse em conhecer mais sobre.	A base da lógica de programação.	Não.	
Prefiro a parte das exatas porque são exatas não tem 'se'	Programar	Sim.	com um amigo que já estudou na UFC e me ensinou poucas coisas
Eu gosto de fazer cálculos	O básico da programação e programar jogos	Não.	
acho interessante	aprende o que ensina	Não.	
gosto da matemática pela forma que ela é	conceitos básicos sobre programação	Sim.	eu faço desenvolvimento de sistemas
Porque eu não tenho afinidade com a matéria	as linguagens de programação	Não.	
tenho afinidade com a materia desde de criança	lógica de programação	Não.	
Gosto de resolver e aprender fórmulas da matéria	Lógica de programação com Scratch ou pelo menos entender um pouco sobre.	Não.	
Gosto de operações lógicas	Adquirir experiência	Não.	
porém não sou muito boa na matéria	lógica de programação	Não.	
porque eu gosto de cálculos	espero aprender a programar	Sim.	em notebook
Adoro toda a logica envolvida com a matemática, e a forma como se resolve os problemas	Aprender a como resolver problemas de programação	Sim.	Meu contato com a programação foi algo de interesse próprio, eu gostava bastante sobre desenvolvimento de jogos, e buscava aprender sobre isso, então aprendi sobre lógica de programação e a programar só o básico, e também comecei a ter interesse em desenvolver sistemas no geral,hoje eu sei um pouco de HTML e python
Gosto de matemática pois é uma matéria a qual eu me identifico e me estimula a minha vontade de estudar.	Aprender a utilizar as ferramentas e o programa.	Sim.	Tive uma apresentação através de um professor.
É uma matéria que eu aprecio por ser algo mais preciso, e gosto de fazer cálculos.	Espero aprender mais sobre programação, de maneira a facilitar meu entendimento quando ver assuntos mais complexos da programação.	Não.	
Eu gosto mas tenho dificuldade na parte da lógica e interpretação de questões.	Espero desenvolver a parte lógica, que é onde eu tenho mais dificuldade.	Não.	
Pois apresenta conhecimentos necessários para todas as áreas da vida.	Lógica de programação e como manipular a mesma em diversos sistemas	Sim.	Apenas operações básicas no python
Eu gosto de fazer cálculos, mas nem sempre sou boa.	Espero saber programar, e desenvolver sistemas.	Sim.	Já tive experiencia com html, css, e phyton. Mas nada muito aprofundado.
Gosto de fazer cálculos, (cálculos simples)	Aprender mais sobre lógica de programação	Não.	
É não me identifico	Aprender a Programar	Sim.	Já cheguei a iniciar um curso de JS pelo o Youtube, Foi bem legal, foi isso que me trouxe até aqui
Gosto de fazer operações	Quero aprender sobre a programação lógica, pois irá me ajudar no futuro	Não.	
Só não me identifico com a matéria mesmo	aprender a programar	Não.	
Não me identifico com a matematica, a area de exatas no caso	tudo possivel,para que eu possa fazer qualquer coisa sem dificuldade nenhuma	Não.	
Pois eu enxergo a matemática como um jogo/desafio, além de fazer eu me sentir mais inteligente e que é com a matemática que os segredos do universo são descobertos.	Eaepo aprender a programar e compreender a lógica da programação.	Sim.	Tive contato em um curso de programação visual no Raimundo de Deus, em geral, consegui ajudar as pessoas com app que eu fiz e foi bom.
Me dou bem com a matéria	Lógica de programação	Não.	
Gosto de resolver desafios (alguns)	O básico da programação	Sim.	Curso na ufc
Gosto de matemática, acho uma matéria desafiadora e importante	Um pouco de programação	Não.	
Não sou muito bom em matematica	conhecimentos gerais de logica de programação	Sim.	Não foi exatamente com progamação e sim com html
Tenho certa dificuldades, mas gosto muito da materiaeria.	Como programar ,e onde aplicar o que aprendo	Sim.	Foi pouco até agora,pois estou fazendo um curso de Java paralelamente.
gosto só não sou bom.	pensamento computacional	Sim.	python, bloco e java baico
eu me considero uma pessoa até que boa em matemática	ter uma noção maior sobre a programação e sua linguagens.	Sim.	ja vi outras pessoas programando e até ja tentei.
Gosto de fazer contas e usar a lógica	Conhecer um pouco mais sobre programação e sua lógica	Sim.	Vendo outras pessoas fazendo comandos simples, e tentando fazer uma tela de introdução escrita usando a linguagem python

Já teve contato com programação visual?	Se respondeu sim para a pergunta anterior, qual tipo de contato, conte sobre sua experiência.	A oficina foi:	O tempo das aulas passou:	A oficina foi:	Mostrei ou vou mostrar meu programa de computador para outras pessoas:	Quero aprender mais sobre como fazer programas de computador.
Não.		(1) Muito Divertida	(1) Muito rápido	(2) Boa	(1) Sim	(1) Sim
Não.		(1) Muito Divertida	(2) Rápido	(1) Excelente	(1) Sim	(1) Sim
Não.		(2) Divertida	(2) Rápido	(1) Excelente	(2) Não	(1) Sim
Não.		(2) Divertida	(1) Muito rápido	(1) Excelente	(2) Não	(1) Sim
Não.		(1) Muito Divertida	(2) Rápido	(1) Excelente	(1) Sim	(1) Sim
Sim.		(1) Muito Divertida	(2) Rápido	(1) Excelente	(1) Sim	(1) Sim
Não.		(1) Muito Divertida	(1) Muito rápido	(1) Excelente	(1) Sim	(1) Sim
Sim.	Programação de controladores, atuadores e sensores no uso da robótica.	(1) Muito Divertida	(1) Muito rápido	(1) Excelente	(1) Sim	(1) Sim
Não.		(1) Muito Divertida	(2) Rápido	(2) Boa	(1) Sim	(1) Sim
Sim.	Escola	(1) Muito Divertida	(2) Rápido	(1) Excelente	(1) Sim	(1) Sim
Não.		(2) Divertida	(2) Rápido	(2) Boa	(1) Sim	(1) Sim
Sim.	No 8º ano do ensino Fundamental, houve um curso de Robótica, durante algumas aulas foi utilizado um site parecido com o Scratch.	(1) Muito Divertida	(2) Rápido	(1) Excelente	(1) Sim	(1) Sim
Não.	não	(1) Muito Divertida	(2) Rápido	(2) Boa	(1) Sim	(1) Sim
Não.		(1) Muito Divertida	(2) Rápido	(1) Excelente	(1) Sim	(1) Sim
Não.		(1) Muito Divertida	(3) Devagar	(1) Excelente	(1) Sim	(1) Sim
Não.		(1) Muito Divertida	(1) Muito rápido	(1) Excelente	(1) Sim	(1) Sim
Sim.	na escola	(1) Muito Divertida	(2) Rápido	(1) Excelente	(1) Sim	(1) Sim
Não.		(1) Muito Divertida	(1) Muito rápido	(1) Excelente	(1) Sim	(1) Sim
Não.		(2) Divertida	(1) Muito rápido	(1) Excelente	(1) Sim	(1) Sim
Não.		(2) Divertida	(2) Rápido	(1) Excelente	(1) Sim	(1) Sim
Não.		(1) Muito Divertida	(2) Rápido	(1) Excelente	(1) Sim	(1) Sim
Não.		(1) Muito Divertida	(1) Muito rápido	(1) Excelente	(1) Sim	(1) Sim
Não.		(1) Muito Divertida	(1) Muito rápido	(1) Excelente	(1) Sim	(1) Sim
Sim.	em notebook	(1) Muito Divertida	(2) Rápido	(1) Excelente	(1) Sim	(1) Sim
Sim.	Quando eu estava aprendendo a desenvolver jogos uma das formas que pensei em fazer isso foi programar em bloco, pois eu não sabia sobre os códigos de programação	(1) Muito Divertida	(3) Devagar	(1) Excelente	(1) Sim	(1) Sim
Não.		(1) Muito Divertida	(2) Rápido	(1) Excelente	(1) Sim	(1) Sim
Sim.	Vi um pouco apenas de programação em blocos, usando Tinkercard, e fazendo alguns códigos para acender e apagar pequenas luzes em um arduino.	(1) Muito Divertida	(2) Rápido	(1) Excelente	(1) Sim	(1) Sim
Não.		(1) Muito Divertida	(2) Rápido	(1) Excelente	(1) Sim	(1) Sim
Não.		(2) Divertida	(2) Rápido	(2) Boa	(1) Sim	(1) Sim
Sim.	Ja tentei fazer um coração em css.	(2) Divertida	(2) Rápido	(1) Excelente	(2) Não	(1) Sim
Não.		(1) Muito Divertida	(1) Muito rápido	(1) Excelente	(1) Sim	(1) Sim
Não.		(1) Muito Divertida	(1) Muito rápido	(1) Excelente	(1) Sim	(1) Sim
Não.		(1) Muito Divertida	(1) Muito rápido	(1) Excelente	(1) Sim	(1) Sim
Não.		(1) Muito Divertida	(1) Muito rápido	(1) Excelente	(1) Sim	(1) Sim
Não.		(1) Muito Divertida	(2) Rápido	(1) Excelente	(1) Sim	(1) Sim
Sim.	Tive contato em um curso de programação visual no Raimundo de Deus, em geral, consegui ajudar as pessoas com app que eu fiz e foi bom.	(1) Muito Divertida	(3) Devagar	(1) Excelente	(1) Sim	(1) Sim
Não.		(1) Muito Divertida	(2) Rápido	(1) Excelente	(1) Sim	(1) Sim
Não.		(1) Muito Divertida	(2) Rápido	(1) Excelente	(1) Sim	(1) Sim
Não.		(1) Muito Divertida	(2) Rápido	(1) Excelente	(1) Sim	(1) Sim
Não.		(1) Muito Divertida	(2) Rápido	(1) Excelente	(1) Sim	(1) Sim
Não.		(1) Muito Divertida	(2) Rápido	(1) Excelente	(1) Sim	(1) Sim
Não.		(1) Muito Divertida	(2) Rápido	(2) Boa	(2) Não	(1) Sim
Não.		(1) Muito Divertida	(2) Rápido	(1) Excelente	(1) Sim	(1) Sim
Sim.	vendo o meu professor fazer operações no python	(1) Muito Divertida	(1) Muito rápido	(1) Excelente	(2) Não	(1) Sim

Fazer um programa de computador é:	Eu gosto de fazer programas de computador:	A computação é útil no dia a dia:	Eu penso em trabalhar com computação:	A oficina foi:	Eu consigo fazer programas de computador:	Eu consigo explicar para um amigo/amiga como fazer um programa de computador:	Fazer um programa de computador é:	Qual a percentagem de tempo estimada você acha que usou o computador durante o curso?
(1) Muito Divertido	(1) Sim	(1) Sim	(1) Sim	(1) Muito fácil	(2) Não	(2) Não	(4) Muito Difícil	80% do tempo ou mais
(2) Divertido	(1) Sim	(1) Sim	(2) Não	(2) Fácil	(1) Sim	(1) Sim	(2) Fácil	80% do tempo ou mais
(1) Muito Divertido	(1) Sim	(1) Sim	(1) Sim	(2) Fácil	(1) Sim	(1) Sim	(3) Difícil	60% a 80% do tempo
(2) Divertido	(1) Sim	(1) Sim	(1) Sim	(2) Fácil	(2) Não	(2) Não	(3) Difícil	20% do tempo ou menos
(1) Muito Divertido	(1) Sim	(1) Sim	(2) Não	(2) Fácil	(1) Sim	(1) Sim	(3) Difícil	80% do tempo ou mais
(2) Divertido	(1) Sim	(1) Sim	(1) Sim	(2) Fácil	(1) Sim	(2) Não	(3) Difícil	80% do tempo ou mais
(1) Muito Divertido	(1) Sim	(1) Sim	(1) Sim	(2) Fácil	(1) Sim	(1) Sim	(2) Fácil	40% a 60% do tempo
(1) Muito Divertido	(1) Sim	(1) Sim	(1) Sim	(1) Muito fácil	(1) Sim	(1) Sim	(2) Fácil	60% a 80% do tempo
(2) Divertido	(1) Sim	(1) Sim	(2) Não	(2) Fácil	(1) Sim	(1) Sim	(3) Difícil	80% do tempo ou mais
(1) Muito Divertido	(1) Sim	(1) Sim	(1) Sim	(2) Fácil	(1) Sim	(1) Sim	(3) Difícil	40% a 60% do tempo
(2) Divertido	(1) Sim	(1) Sim	(1) Sim	(3) Difícil	(2) Não	(1) Sim	(2) Fácil	40% a 60% do tempo
(2) Divertido	(1) Sim	(1) Sim	(1) Sim	(2) Fácil	(1) Sim	(2) Não	(3) Difícil	80% do tempo ou mais
(2) Divertido	(1) Sim	(1) Sim	(2) Não	(2) Fácil	(2) Não	(2) Não	(3) Difícil	60% a 80% do tempo
(2) Divertido	(1) Sim	(1) Sim	(1) Sim	(2) Fácil	(1) Sim	(2) Não	(3) Difícil	80% do tempo ou mais
(1) Muito Divertido	(1) Sim	(1) Sim	(1) Sim	(2) Fácil	(1) Sim	(1) Sim	(3) Difícil	60% a 80% do tempo
(1) Muito Divertido	(1) Sim	(1) Sim	(1) Sim	(2) Fácil	(1) Sim	(1) Sim	(2) Fácil	80% do tempo ou mais
(1) Muito Divertido	(1) Sim	(1) Sim	(2) Não	(2) Fácil	(1) Sim	(1) Sim	(3) Difícil	40% a 60% do tempo
(1) Muito Divertido	(1) Sim	(1) Sim	(1) Sim	(1) Muito fácil	(1) Sim	(1) Sim	(2) Fácil	80% do tempo ou mais
(2) Divertido	(1) Sim	(1) Sim	(1) Sim	(2) Fácil	(1) Sim	(1) Sim	(2) Fácil	60% a 80% do tempo
(2) Divertido	(1) Sim	(1) Sim	(1) Sim	(3) Difícil	(2) Não	(2) Não	(3) Difícil	40% a 60% do tempo
(2) Divertido	(1) Sim	(1) Sim	(2) Não	(3) Difícil	(2) Não	(2) Não	(3) Difícil	80% do tempo ou mais
(1) Muito Divertido	(1) Sim	(1) Sim	(1) Sim	(2) Fácil	(1) Sim	(1) Sim	(2) Fácil	40% a 60% do tempo
(2) Divertido	(1) Sim	(1) Sim	(1) Sim	(3) Difícil	(1) Sim	(1) Sim	(3) Difícil	80% do tempo ou mais
(1) Muito Divertido	(1) Sim	(1) Sim	(1) Sim	(2) Fácil	(1) Sim	(2) Não	(3) Difícil	60% a 80% do tempo
(1) Muito Divertido	(1) Sim	(1) Sim	(1) Sim	(1) Muito fácil	(1) Sim	(1) Sim	(3) Difícil	60% a 80% do tempo
(1) Muito Divertido	(1) Sim	(1) Sim	(1) Sim	(2) Fácil	(1) Sim	(1) Sim	(2) Fácil	40% a 60% do tempo
(1) Muito Divertido	(1) Sim	(1) Sim	(1) Sim	(2) Fácil	(1) Sim	(1) Sim	(2) Fácil	60% a 80% do tempo
(1) Muito Divertido	(1) Sim	(1) Sim	(1) Sim	(2) Fácil	(1) Sim	(1) Sim	(3) Difícil	60% a 80% do tempo
(2) Divertido	(1) Sim	(1) Sim	(1) Sim	(3) Difícil	(1) Sim	(2) Não	(3) Difícil	80% do tempo ou mais
(2) Divertido	(1) Sim	(1) Sim	(1) Sim	(2) Fácil	(1) Sim	(1) Sim	(2) Fácil	80% do tempo ou mais
(1) Muito Divertido	(1) Sim	(1) Sim	(1) Sim	(2) Fácil	(1) Sim	(1) Sim	(2) Fácil	60% a 80% do tempo
(2) Divertido	(1) Sim	(1) Sim	(1) Sim	(3) Difícil	(1) Sim	(1) Sim	(3) Difícil	40% a 60% do tempo
(1) Muito Divertido	(1) Sim	(1) Sim	(1) Sim	(2) Fácil	(1) Sim	(1) Sim	(2) Fácil	80% do tempo ou mais
(2) Divertido	(1) Sim	(1) Sim	(1) Sim	(2) Fácil	(1) Sim	(1) Sim	(2) Fácil	80% do tempo ou mais
(1) Muito Divertido	(1) Sim	(1) Sim	(1) Sim	(2) Fácil	(1) Sim	(2) Não	(3) Difícil	80% do tempo ou mais
(2) Divertido	(1) Sim	(1) Sim	(1) Sim	(2) Fácil	(1) Sim	(1) Sim	(2) Fácil	80% do tempo ou mais
(1) Muito Divertido	(1) Sim	(1) Sim	(1) Sim	(2) Fácil	(1) Sim	(1) Sim	(2) Fácil	60% a 80% do tempo
(1) Muito Divertido	(1) Sim	(1) Sim	(1) Sim	(2) Fácil	(1) Sim	(1) Sim	(2) Fácil	80% do tempo ou mais
(2) Divertido	(1) Sim	(1) Sim	(1) Sim	(2) Fácil	(1) Sim	(1) Sim	(3) Difícil	60% a 80% do tempo
(1) Muito Divertido	(1) Sim	(1) Sim	(1) Sim	(2) Fácil	(1) Sim	(2) Não	(3) Difícil	40% a 60% do tempo
(2) Divertido	(1) Sim	(1) Sim	(1) Sim	(2) Fácil	(1) Sim	(1) Sim	(3) Difícil	40% a 60% do tempo
(1) Muito Divertido	(1) Sim	(1) Sim	(1) Sim	(2) Fácil	(1) Sim	(1) Sim	(2) Fácil	40% a 60% do tempo
(2) Divertido	(1) Sim	(1) Sim	(1) Sim	(2) Fácil	(2) Não	(2) Não	(3) Difícil	60% a 80% do tempo
(1) Muito Divertido	(1) Sim	(1) Sim	(1) Sim	(2) Fácil	(1) Sim	(1) Sim	(3) Difícil	60% a 80% do tempo

Diga com as suas palavras, o que você aprendeu durante as aulas?
A fazer um jogo no Scratch
Aprendi a usar o scratch e fazer joguinhos fáceis/histórinhas usando a programação.
Aprendi muito sobre a lógica de programação
Aprendi a teoria da programação
Eu consegui aprender à como passar informações pro computador e usá-las para programar. Além de entender a diferença entre o nosso entendimento, para o do computador.
Muita coisa
A usar mais a cabeça e ser dinâmico
Durante o curso eu aprendi a lógica básica da programação, como por exemplo, condicional, operadores, variáveis, loop, etc.
Aprendi um pouco da programação que irá me ajudar futuramente
Aprendi algumas lógicas que vão servir muito para mim no uso de computadores.
como usar as ferramentas de scratch como variáveis, sensores e diga.
Boa parte da lógica da programação, principalmente dentro da plataforma SCRATCH, onde era divertido construir códigos.
Aprendi sobre como o computador entende e como posso organiza-lo, e um pouco do algoritmo
Lógica de programação, como eu posso passar o que eu quero que seja feito para computador.
muito sobre algumas regras
Aprendi a programar no Scratch
Aprendi a programa no Scratch
Aprendi uma série de coisas muito importantes como o que é uma variável, que ajuda demais nesse mundo da programação e o curso proporcionou uma introdução nessa área.
-lógica de programação -pensamento computacional -programação no scratch -fazer jogos utilizando scratch
A lógica por trás das estruturas do scrhatch.
Aprendi como programar usando funções, sensores, variáveis e um pocou como criar blocos .
Ter ideias e ser criativo
a programação com scratch
A saber criar um programa no Scratch
Principalmente, sobre lógica de programação e abstração
Aprendi a usar o Scratch e suas utilidades.
Programação nem sempre é só programar, é resolver erros e problemas de maneira lógica e criativa. Além de toda a teoria e práticas das aulas.
Eu aprendi que podemos achar um caminho para tudo, que qualquer coisa pode ser feita na programação e que com processos finitos podemos criar infinitas soluções.
Como fazer operações, movimentar, mexer com variáveis e fazer o personagem falar, além de conseguir mudar fantasias e cenários no projeto.
Aprendi sobre lógica de programação, assim acredito que facilitará muito quando eu for mexer com as Linguagens de programação
Aprendi sobre lógica de programação
Aprendi a fazer programas/jogos na ferramenta Scratch e aprendi mais sobre lógica de programação.
aprendi que realmente a computação é muito essencial para todos
Eu aprendi a fazer um programa utilizando a ferramenta Scratch de uma maneira divertida, utilizamos praticamente todos os blocos durante as aulas entre eles variáveis, comando se, então, diga, pergunte, até a criar o próprio bloco. Aprendi também o que é algoritmo, pensamento computacional e muitas outras coisas que vou levar como ensinamento.
Aprendi muita coisa que vou levar para as aulas do curso de desenvolvimento de sistemas
Conceitos importantes na área de programação, os quais podem me ajudar a programar de maneira mais eficiente
Lógica de programação
Aprendi a questão do pensamento computacional, como ir criando uma lógica q desse certo e etc...
Lógica da computação, e conceitos básicos de programação.
Tudo que foi proposto
Aprendi como funciona a lógica da programação, algoritmos, funções, variáveis entre outros conceitos
Aprender ater mais pensamento computacional
Lógica da programação, o que é algoritmo, variáveis (etc).
Foram aulas muito boas que puderam me ensinar sobre algoritmo, pensamento computacional, como funciona algumas linguagens de programação e também boa parte da lógica por trás de toda a programação.

O que você achou sobre Pensamento Computacional? Como isso pode ajudar em outras disciplinas?
Achei complicado
Com certeza vai ajudar muito em informática básica, já que agora vamos começar a estudar sobre programação e tal, então a oficina talvez já sirva de base para o que vamos estudar.
É um pensamento um pouco complicado, mas se bem interpretado pode ajudar muito a compreender outras disciplinas
Pode me ajudar a entender um pouco mais sobre o curso de desenvolvimento de sistemas
Achei bastante interessante. O pensamento computacional pode ajudar na outras disciplinas fazendo com que o estudante pense fora da caixa para resolver um problema, além de ensiná-lo a interpretar situações para depois conseguir pensar em uma solução.
Achei muito legal, e pode ajudar de várias formas
Bem legal em atividades bem mais dinâmicas
É uma ótima maneira de resolver problemas; pode me ajudar quando eu encontrar empecilhos no desenvolvimento dos meus estudos.
Bom basicamente na maioria das coisas
Acho muito interessante e necessário, ajuda no raciocínio lógico.
Ajudar a resolver os problemas de forma mais fácil e eficiente.
É algo um pouco complicado de entender no começo, mas com um pouco de esforço e prática é possível apreender, e após isso fica muito legal de se trabalhar. Isso irá ajudar durante as aulas técnicas do curso de Desenvolvimento de Sistemas.
Acho interessante. Pode ajudar a resolver problemas que se manifestam ao longo dos estudos
Muito legal, e útil para meu curso técnico e para que no futuro possa desenvolver bons trabalhos.
útil, a resolver problemas
Na matemática
Achei interessante
Pensamento computacional é muito legal de entender, porque ele pode ser definido como uma habilidade para resolver problemas e desafios de forma eficiente, assim como um computador faria.
Isso pode melhorar nossa capacidade de interpretação e consequentemente nosso entendimento é interpretação das disciplinas.
Pode ser muito útil na resolução de problemas.
O pensamento computacional pode ajudar em muitas questões, principalmente nas disciplinas pois a forma que ele pode ser usado te ajuda a desenvolver algo que vai te auxiliar no que você deseja.
Acho que ajuda muito na resolução de problemas, como fazer contas e outros tipos de problemas
gostei de aprender sobre. Na resolução de problemas
Muito legal gostei muito. Em estudo
É algo muito útil e de grande importância para a área da programação e outras áreas, acho que o pensamento computacional em si não é tão útil em outras disciplinas, mas acho que a lógica de "usar aquilo para resolver os meus problemas" é algo que dava pra ser aplicado em outras disciplinas, como usar conhecimentos de matemática para resolver problemas da sua vida, por exemplo a regra de 3 que pode ser aplicada em diversas situações.
É uma ótima maneira de resolver problemas. Ajuda tanto na escola quanto na vida cotidiana, pois pode nos ajudar a resolver problemas de maneira lógica, objetiva e funcional, sem muito esforço.
Eu achei muito bom, é algo que desenvolve muito a lógica e te possibilita pensar fora da caixa, por isso, pode ser facilmente utilizado em outras disciplinas, ajudando a seguir um caminho mais fácil e prático.
Algo muito útil atualmente. Nos auxiliando na compreensão da tecnologia atual, para nos mantermos atualizados do mundo exterior.
Achei muito proveitoso, pois essa lógica que aprendemos vai nos ajudar no nosso curso na escola
Achei muito interessante, construindo conhecimentos lógicos
Pode ajudar a resolver problemáticas de outras disciplinas
pode ajudar a entender a compreensão melhor
Legal, não sabia a definição de fato antes, acredito sim já que resolver problemas.
E muito importante ter esse pensamento, Você pode resolver vários problemas de outras disciplinas com isso dúvidas, perguntas e etc
Gostei, pois o Pensamento Computacional me mostrou que o computador segue uma lógica, a maneira que ele pensa. Se o começo tem um erro, o resto terá erro. Isso me ajudou a pensar que as outras disciplinas devem seguir uma ordem.
Achei bem interessante esse método
É bem interessante. Nos ajudando a descobrir uma forma "certa" de aprender/pensar quando se trata de diferentes conhecimentos. Por exemplo as vezes podemos tentar estudar história com o método q usamos pra matemática, o q não daria certo.
Interessante. Pode nos dar uma base para os outros conteúdos
Vai ajudar muito, claro
Pode me auxiliar nas matérias do curso, mas também pode ser aplicada em algumas áreas de outras matérias.
Legal e eficaz
Ajuda a ter uma noção de como resolver problemas.
O pensamento computacional ele ajuda bastante na organização das ideias e também dos serviços, que muitas vezes é preciso ser feito. Não só na computação em si, mas em outras disciplinas se torna bem útil essa organização.

Houve dificuldades com a manipulação do Scratch? Quais dificuldades você sentiu?
Só em algumas coisas q eu não entendi como variáveis
Não,o programa em si é bem fácil de utilizar,além de que todas as dúvidas que eu tive foram tiradas durantes as aulas.
Senti dificuldade em saber quais caixas existiam dentro do Scratch para que eu pudesse programar
Não consegui ficar muito tempo programado
Sim, na complexidades dos códigos.
Nenhuma
Não
Não houve dificuldade sobre o entendimento da lógica do programa, no entanto, ele possui limitações que acabam dificultando no desenvolvimento de algumas ideias.
Um pouco, mas não senti tanta dificuldade
No começo dos desafios eu sentia dificuldade, mas depois de fazer uma vez eu já aprendia e conseguia fazer só.
Houve, tive difícil de aprender algumas ferramentas dos scratch.
Um pouco, a parte da lógica foi difícil de entender no começo, mas após um tempo e um pouco de prática ficou divertido de se trabalhar. As dificuldades em geral foi apenas o tempo.
Um pouco. No começo eu tive dificuldades de entender como o computador entende, mas ao longo da oficina, fui aprimorando isso.
Poucas dificuldades, em alguns momento não consegui passar exatamente o que eu queria para o scratch.
não.
Sim, apenas no começo quando eu nao sabia o que cada bloco fazia
Não
Não, na minha opnião o Scratch não proporciona grandes problemas, uma dificuldade pode ser tentar resolver o problema o que as vezes por um mínimo detalhe nos atrapalha em questão de escrever o código correto por exemplo.
Não tive muita dificuldade, somente na resolução dos problemas apresentados e como resolver eles no começo do curso
Não muitas, depois de conhecer eu me acostumei com a linguagem.
Sim, em entender melhor como cada poderia ser usado as funções, variáveis etc, e até mesmo conseguir fazer o jogo ou outra coisa, darem certo no final.
um pouco de dificuldade com a organização dos blocos e da interface do Scratch
sim. Não conseguir encaixar os blocos nas funções que eu queria
Sim . As de usar o comando certo para utiliza-lo
Na minha experiência, acho que o scratch é um pouco limitado, é possível sim criar projetos bons, mas não é algo que genuinamente bem feito,por exemplo para a criação de jogos, que é possível fazer jogos no scratch, mas criar um jogo com diversas mecânicas ele acaba sendo limitado, mas o scrctch é muito útil para ensinar sobre lógica de programação
Houve em alguns momentos, senti dificuldade em sequenciar alguns comandos e deixar movimentos de forma natural.
Não tive tantas dificuldades. Scratch é uma ferramenta visual bem simples, mas intuitiva o bastante para aprender, então dificuldades mesmo talvez só na teoria, mas acho que consegui pegar a maior parte da base.
Dificuldades com certeza tive, mas facilmente minhas dúvidas foram tiradas, algumas vezes eu senti dificuldade em colocar na prática a teoria
Quais comandos utilizar, como deixar o programa menor e etc.
Sim, tive dificuldade de pensar em uma lógica pra deixar o código curto
Não
Sim, em fazer o objeto se mover
não
As vezes tinha dificuldade de qual bloco usar para executar algo, mas ai fui testando e foi dando certo
Não houve
Criar e utilizar blocos de funções
Não
Um pouco mais na questão de montar uma lógica e tals
Não
Nenhuma
Mais ou menos .tive dificuldades em baixar modelos da Internet, trocas de fantasias, e nos sons
Sim na parte de função
No geral é fácil e simples de entender, o mais complicado é pensar perfeitamente o passo a passo.
As dificuldades foram mais voltadas para como aprender a utilizar a ferramenta, aprender a usar adequadamente os blocos e formar blocos maiores que fossem funcionais.

Houve dificuldades em algum conteúdo específico ou em alguma aula? Quais dificuldades você sentiu?
Variável
Nada em específico
Senti mais dificuldade em funções
Não
Senti um pouco de dificuldade no conteúdo de repetição. Minha maior dificuldade foi montar o código corretamente
Nenhuma
Não
Não houve dificuldade.
Só que as vezes não dava certo
Não.
As variáveis.
Apenas na parte lógica, mas que foi fácil estudar sobre.
Variáveis. É um conteúdo simples, mas que eu demorei um pouco para entender a mecânica.
Não
criar sua própria função
Não
Não
Não.
Na aula em que era para transformar celsius em fahrenheit
Penso que minha maior dificuldade era na organização de ideias na hora de executar o programa.
Ao se utilizar os blocos, não consegui entender muito bem como funcionava.
com operações matemáticas e quais blocos eu deveria utilizar
alguns. Em entender como eu conseguiria colocar em prática aquilo que era ensinado
Sim . No começo para se adaptar aos comandos
Não senti dificuldades nas aulas
Não houve.
Função. Acho que entendi na teoria, mas na prática do Scratch, senti um pouco de dificuldade, já que não consegui executar tão bem quanto esperado.
Eu senti dificuldade no "repita até que" ou "faça tal coisa se"
Sim, funções e como manipular elas.
sim, tive dificuldade com a troca de fantasia do gato.
Não
Não
não
Acho que não
Não
Senti dificuldade em criar blocos de Função
Não
Não
Não
Nenhuma
Não
Função
Não.
Os conteúdos foram bem repassados e foi possível aprender bem cada um deles, para que fossem colocados em prática.

O que mais gostou de trabalhar durante o curso?
Fazer o boneco andar
Fazer os personagens responderem perguntas matemáticas e questões no geral.
Gostei de trabalhar com a parte de variáveis
Foi legal entender como funciona as lógicas de programação
Com variáveis
A programação
Conseguir fazer meus projetos
Acredito que tenha sido os desafios, além do fato de todo o processo ter sido ao lado de meus colegas.
A programação de um jeito diferente
A parte da lógica matemática.
Aprender a criar jogos.
A construção do trabalho final do curso, onde todas as duplas/trios precisaram criar um projeto para apresentar.
A liberdade de conseguir usar sua criatividade no scratch
A produção do jogo, foi muito maravilhoso poder colocar o que aprendi em um trabalho e mostrar o que consegui para meus colegas.
programar os jogos
Programar
Tudo
A parte da criação de jogos de tema livre, essa parte foi simplesmente maravilhosa porque me incentivou a apresentar o melhor jogo que eu podia fornecer e a sensação de missão cumprida é maravilhoso de sentir.
Gostei de trabalhar o pensamento computacional
Quando era necessário o uso de lógica para a solução de problemas.
Com os movimentos, diálogo, sensores, a aparência.
funções e criação de variáveis
gostei muito de ver os resultados do meu esforço na prática na programação
O joguinhos que noa criamos e compartilhamos uns com outros
Função
Lógica.
Comando Se, então, se não.
Da dinâmica e da forma de ensinar
Efetuar operações no scratch
Gostei de fazer as tarefas, porque era algo meio livre, a gente podia adicionar mais alguma coisa, mas claro sem perder o foco principal da atividade.
Montar programação envolvendo comunicação
Os comandos, se e senão, se, e as variáveis
fazer os joguinhos e montar os bonecos
A prática em si
Tudo
Variáveis e condições
Os desafios
Prática e criação dos nossos próprios projetos
A lógica
As variáveis, estruturas de se esse não, e operações lógicas e aritméticas aplicadas no código.
Pensamento computacional
A prática e a liberdade que o programa nos proporciona.
A questão de movimentar os personagens torna-los fluidos, adicionar variáveis e comparar os seus resultados, conhecer e aprender como montar novos blocos para poder aplicar funções.

O que menos gostou de trabalhar durante o curso?
Cenário
Fazer o personagem caminhar em linha reta enquanto mudava de fantasia (por mais simples que seja eu demorei muito para conseguir fazer isso,mas deu tudo certo)
Eu gostei menos da parte de funções
Nada
Com blocos de repetição
Gostei de tudo
Nada
Não recorde de algo que não tenha gostado.
Operações lógicas
Sinceramente eu gostei de tudo.
tempo de duração.
A parte de função, foi um pouco chato de aprender.
Talvez as operações aritméticas.
Ter que fazer o personagem andar.
nada
Eu gostei de tudo
Nada
A parte de criar Blocos, isso porque eu não encontrei grande finalidade para eles.
Transformação de celsius em fahrenheit
O pouco tempo para a desenvoltura das atividades.
Quando o desafio era para fazer cálculos.
com operações matemáticas
distinguir como eu deveria usar as ferramentas do scratch
Quando o comando dava errado
Sendo sincero eu gostei de tudo
Função.
Que não teve mais prática
Funções
Nada.
Comandos que utilizam variáveis
A parte aritmética
nada
Nada
Funções
A parte teórica
Nada
Nada
Nada
Função
A parte das funções.
Não tiveram conteúdos que fossem do meu desagrado, afinal todos me fizeram aprender bastante sobre uma gama de possibilidades da programação, em coisas que eu não conhecia ou não sabia muito bem como usar.

O que achou da metodologia das aulas? O que melhoraria?
Eu gostei bastante, eu só melhoraria o tamanho do slide (deixaria um pouco maior) por que do lugar que eu estava não conseguia enxergar nadinha.
Gostei bastante da metodologia, acho que poderia melhorar na explicação das caixas que tem dentro do programa, para ajudar o aluno a encontrar o que precisa pra realizar a atividade
Foi boa
Adorei bastante das aulas. Os monitores eram muito simpáticos e prestativos e as aulas eram bem precisas e interessantes, então não mudaria nada na metodologia, só aumentaria o tempo das aulas.
Muito boa
Muito boas, não sei.
Em relação de trabalhar com essa área para pessoas que estavam interessadas no assunto e explicar sobre uma plataforma simples, eu acredito que foi uma boa metologia, principalmente, pelo fato de ser voltado para os jovens e possuir diversas dinâmicas (desafios, competições, prêmios, etc). As aulas foram boas e curtas, mas não acredito que possa ser usado para qualquer público ou assunto.
Bom, não mudaria nada
Achei muito boa, só acho que deveria ter tido mais aulas para poder abranger mais assuntos.
Foi bom, mas eu melhoria na especificação de cada ferramenta do scartch.
Perfeita. Sempre alternando entre lógica e prática, deixando as aulas mais interessantes. Nada.
Achei bem didático. Em geral foi muito boa, não tenho reclamações.
Incrível.
incrível.
Boas, o professor ensina muito bem
Achei bem boa
Muito boas. E sinceramente não melhoraria nada, achei o ritmo das aulas muito boas.
Eu gostei, somente a organização do tempo
Eu penso que uma melhor organização do tempo viria a calhar mas de resto consegui aprender muito bem tudo aquilo que me foi proposto.
Boa, foi explicado de forma com que fosse fácil de entender, às vezes não, mas foram aulas muito proveitosas.
Achei ótima e não mudaria nada
muito boa. Maior auxílio aos alunos individualmente
Legal e divertida. Trazer mais aulas assim
Ótima metodologia de ensino, não melhoraria nada
Muito boa e divertida.
A metodologia é ótima e muito boa. Funciona bastante. Não sei no que melhoraria, mas talvez colocasse mais de um exemplo, tanto cotidiano quanto da programação, com imagens em slide, para ajudar no entendimento. Exemplo: Variável -Cotidiano: Dar um nome a um cachorro e guardá-lo em uma coleira. -Programação: Usar a imagem de um código (de maneira simples) e mostrar uma imagem ao lado mostrando o que aquele código faz.
Achei muito incrível e dinâmico, houve muita interação com os alunos e pra mim está perfeito, só colocaria uma aula apenas de prática, para facilitar os alunos a se habituarem com o programa.
Ótima, principalmente os professores muito atenciosos.
Eu gostei, não acho que tenha algo a mudar
Achei excelente
Excelente
achei excelente, não tem o que melhorar
Muito boa
As aulas eram perfeitas
Muito bom
Formas de dar exemplo
Show de bola
Muito boa
Perfeito, nada
Acho ótima. pois propõem que os alunos observem a plataforma e criem suas soluções.
Melhor qualidade nos slides
Boa, mas acho que precisava de mais refinação nas explicações.
Foi muito boa. Uma boa mistura de prática e teoria o que tornou as aulas bem dinâmicas. Não sei ao certo o que poderia ser melhorado.

Você tem interesse em seguir nessa área de programação?	Entrada de Dados	Saída de Dados	Variáveis	Operações Aritméticas	Operações Lógicas	Estrutura de Decisão	Estrutura de Repetição	Funções
Sim, já pretendia seguir.	1,25	1,25	1,25	1,25	1,25	1,25	0	0
No momento não pretendo,mas pode ser um plano B caso o que eu quero de inicio não funcione	1,25	1,25	1,25	0	0	1,25	1,25	0
Sim, já pretendia seguir.	1,25	1,25	1,25	1,25	1,25	1,25	1,25	0
Sim, já pretendia seguir.	1,25	1,25	1,25	1,25	1,25	1,25	1,25	0
Não sei	1,25	1,25	1,25	1,25	0	0	0	0
Ainda não sei.	1,25	0	1,25	1,25	1,25	1,25	1,25	0
Sim, mas surgiu interesse durante o curso.	1,25	1,25	1,25	1,25	1,25	1,25	1,25	0
Sim, já pretendia seguir.	1,25	1,25	1,25	1,25	1,25	1,25	1,25	0
Ainda não sei.	1,25	1,25	1,25	0	1,25	1,25	1,25	0
Sim, já pretendia seguir.	1,25	1,25	1,25	1,25	1,25	1,25	0	0
Sim, já pretendia seguir.	1,25	1,25	1,25	1,25	1,25	1,25	0	0
Ainda não sei.	1,25	1,25	1,25	0	1,25	1,25	1,25	0
Ainda não tenho certeza	1,25	1,25	1,25	1,25	1,25	1,25	1,25	0
Ainda não sei.	1,25	1,25	1,25	1,25	1,25	1,25	1,25	0
Sim, mas surgiu interesse durante o curso.	1,25	1,25	1,25	1,25	1,25	1,25	1,25	0
Sim, já pretendia seguir.	1,25	1,25	1,25	1,25	0	0	0	0
Não.	0	0	0	0	0	0	0	0
Sim, já pretendia seguir.	1,25	1,25	1,25	1,25	1,25	1,25	1,25	0
Ainda não sei.	1,25	1,25	1,25	0	0	1,25	1,25	0
Ainda não sei.	1,25	1,25	1,25	1,25	1,25	1,25	1,25	0
Ainda não sei.	1,25	1,25	1,25	1,25	1,25	1,25	0	0
Sim, já pretendia seguir.	1,25	1,25	1,25	1,25	0	1,25	1,25	0
Sim, mas surgiu interesse durante o curso.	1,25	1,25	1,25	1,25	1,25	1,25	0	0
Sim, já pretendia seguir.	1,25	1,25	1,25	1,25	1,25	1,25	1,25	0
Sim, já pretendia seguir.	1,25	1,25	1,25	1,25	1,25	1,25	1,25	0
Ainda não sei.	1,25	1,25	1,25	1,25	1,25	1,25	1,25	0
Sim, mas surgiu interesse durante o curso.	1,25	1,25	1,25	0	1,25	1,25	1,25	0
Ainda não sei.	1,25	1,25	1,25	1,25	1,25	1,25	1,25	0
Sim, mas surgiu interesse durante o curso.	1,25	1,25	1,25	0	1,25	1,25	1,25	0
Sim, já pretendia seguir.	1,25	1,25	1,25	1,25	1,25	1,25	1,25	0
Sim, mas surgiu interesse durante o curso.	1,25	1,25	1,25	0	0	0	1,25	0
Sim, já pretendia seguir.	1,25	1,25	0	0	0	1,25	1,25	0
Sim, já pretendia seguir.	1,25	1,25	1,25	1,25	0	1,25	1,25	0
Sim, mas surgiu interesse durante o curso.	1,25	1,25	0	0	1,25	1,25	0	0
Sim, já pretendia seguir.	1,25	1,25	0	0	0	1,25	1,25	0
Sim, já pretendia seguir.	1,25	1,25	0	0	0	1,25	1,25	0
Sim, já pretendia seguir.	1,25	1,25	0	0	1,25	1,25	0	0
Sim, já pretendia seguir.	1,25	1,25	1,25	1,25	1,25	1,25	1,25	0
Sim, já pretendia seguir.	1,25	1,25	1,25	1,25	1,25	1,25	1,25	0
Sim, já pretendia seguir.	1,25	1,25	0	0	0	1,25	1,25	0
Sim, já pretendia seguir.	1,25	1,25	0	0	1,25	1,25	0	0
Sim, já pretendia seguir.	1,25	1,25	1,25	1,25	1,25	1,25	1,25	0
Sim, já pretendia seguir.	1,25	1,25	0	0	0	1,25	1,25	0
Sim, já pretendia seguir.	1,25	1,25	0	0	1,25	1,25	0	0
Sim, já pretendia seguir.	1,25	1,25	1,25	1,25	1,25	1,25	1,25	0
Sim, já pretendia seguir.	1,25	0	1,25	1,25	1,25	1,25	1,25	0
Sim, mas surgiu interesse durante o curso.	1,25	1,25	1,25	0	1,25	1,25	0	0
Sim, já pretendia seguir.	1,25	1,25	1,25	1,25	1,25	1,25	1,25	0
Sim, mas surgiu interesse durante o curso.	1,25	1,25	1,25	1,25	1,25	1,25	0	0
Sim, já pretendia seguir.	1,25	1,25	1,25	0	1,25	1,25	0	0

Nota de Conhecimento Técnico	Avaliador 1	Avaliador 2	Avaliador 3	Avaliador 4	Nota de Criatividade e Inovação	Nota Final
7,5	9	8,5	7	10	8,625	8,0625
6,25	8	7	8	10	8,25	7,25
8,75	8,5	8	9	10	8,875	8,8125
8,75	7,5	8	9	10	8,625	8,6875
5	6	7,5	8	10	7,875	6,4375
7,5	6	7	7	10	7,5	7,5
8,75	9	8,5	9	10	9,125	8,9375
8,75	10	8,5	8	10	9,125	8,9375
7,5	8	8	9	10	8,75	8,125
7,5	8	7	7	10	8	7,75
7,5	6	6	6	10	7	7,25
7,5	8,5	9	7,5	10	8,75	8,125
8,75	8	8	8	10	8,5	8,625
8,75	9,5	10	10	10	9,875	9,3125
8,75	7,5	8	9	10	8,625	8,6875
5	6	7,5	8	10	7,875	6,4375
0	0	0	0	0	0	0
8,75	10	8,5	8	10	9,125	8,9375
6,25	8	7	8	10	8,25	7,25
8,75	9,5	10	10	10	9,875	9,3125
7,5	6	6	6	10	7	7,25
7,5	7,7	7,5	7,5	10	8,175	7,8375
7,5	8	7	7	10	8	7,75
8,75	8	8	8	10	8,5	8,625
8,75	10	8,5	8	10	9,125	8,9375
8,75	7,5	8	9	10	8,625	8,6875
7,5	8,5	9	7,5	10	8,75	8,125
8,75	8,5	8	9	10	8,875	8,8125
7,5	8	8	9	10	8,75	8,125
8,75	8,5	8	9	10	8,875	8,8125
5	8	7,5	7	10	8,125	6,5625
5	8	8	7	10	8,25	6,625
7,5	7,7	7,5	7,5	10	8,175	7,8375
5	8	8	5	10	7,75	6,375
5	8	8	7	10	8,25	6,625
8,75	7	7	7,5	10	7,875	8,3125
5	8	8	7	10	8,25	6,625
5	8	8	5	10	7,75	6,375
8,75	9	8,5	9	10	9,125	8,9375
7,5	6	7	7	10	7,5	7,5
6,25	7,5	8,5	7	10	8,25	7,25
8,75	7	7	7,5	10	7,875	8,3125
7,5	9	8,5	7	10	8,625	8,0625
6,25	7,5	8,5	7	10	8,25	7,25

ANEXO A – TERMO DE CONSENTIMENTO LEARNINGLAB



TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

Prezado, Senhor(a),

Estamos solicitando a sua autorização para permitir que o menor pelo qual você é responsável participe da pesquisa proposta pelo projeto LearningLab. A participação é voluntária, e você tem o direito absoluto de recusar sua autorização, em qualquer fase da pesquisa.

1) Procedimento e Objetivo

Durante o processo, o participante será convidado(a) a responder a um formulário que abordará questões relacionadas às suas experiências em ações de extensão no ensino médio. Neste estudo, temos como objetivo avaliar como as ações de extensão no ensino médio impactam o desempenho acadêmico, o desenvolvimento de habilidades sociais e a consciência cívica dos estudantes.

2) Tratamento de possíveis riscos e desconfortos

Reconhecemos que esse processo pode despertar desconforto ao recordar experiências passadas e emoções desafiadoras. Para garantir a privacidade do participante, todas as respostas serão mantidas estritamente confidenciais.

3) Benefícios e Custos

Este estudo contribuirá com resultados importantes para a pesquisa de um modo geral nas áreas de Engenharia de Software e Ciência da Computação. Você não terá nenhum gasto ou ônus com a sua participação no estudo e também não receberá qualquer espécie de reembolso ou gratificação devido à participação na pesquisa.

4) Confidencialidade da Pesquisa

Os dados fornecidos pelo participante serão tratados com a máxima confidencialidade e não serão divulgados de forma que possam identificá-lo(a). As informações coletadas serão utilizadas apenas em congressos ou publicações científicas. Lembre-se de que você e/ou seu menor pelo qual é responsável poderão deixar de participar a qualquer momento, sem nenhum problema.

5) Organização da Pesquisa

Os pesquisadores responsáveis pelo estudo poderão fornecer qualquer esclarecimento sobre o mesmo, assim como tirar dúvidas, bastando entrar em contato pelos seguintes e-mails:

Pesquisador: Robério Girão Pinho - roberiogiraopinho@gmail.com / LearningLab UFC-RUSSAS

Orientadora/Coordenadora: Jacilane de Holanda Rabelo - jacilane.rabelo@ufc.br UFC-RUSSAS

6) Declaração de Consentimento

Eu responsável legal por _____, declaro ter sido informado(a) e concordo com a participação do(a) meu filho (a) como participante na pesquisa do projeto LearningLab, e que toda a linguagem técnica utilizada na descrição deste estudo de pesquisa foi explicada satisfatoriamente.

Local e Data:

Participante	Orientador
Nome: _____ Assinatura: _____	Nome: Jacilane de Holanda Rabelo Assinatura: _____

ANEXO B – TERMO DE ASSENTIMENTO LEARNINGLAB



TERMO DE ASSENTIMENTO

Prezado, Senhor(a),

Você está sendo convidado(a) a participar de uma pesquisa realizada pelo projeto LearningLab. Sua participação é completamente voluntária, e você não deve aceitar participar se não desejar. Por favor, leia atentamente as informações a seguir e sinta-se à vontade para fazer qualquer pergunta caso tenha dúvidas.

1) Procedimento e Objetivo

Durante o processo, o participante será convidado(a) a responder a um formulário que abordará questões relacionadas às suas experiências em ações de extensão no ensino médio. Neste estudo pretendemos avaliar como as ações de extensão no ensino médio impactam o desempenho acadêmico, o desenvolvimento de habilidades sociais e a consciência cívica dos estudantes.

2) Tratamento de possíveis riscos e desconfortos

Reconhecemos que esse processo pode despertar desconforto ao recordar experiências passadas e emoções desafiadoras. Para garantir sua privacidade, suas respostas serão mantidas estritamente confidenciais.

3) Benefícios e Custos

Este estudo contribuirá com resultados importantes para a pesquisa de um modo geral nas áreas de Engenharia de Software e Ciência da Computação. Você não terá nenhum gasto ou ônus com a sua participação no estudo e também não receberá qualquer espécie de reembolso ou gratificação devido à participação na pesquisa.

4) Confidencialidade da Pesquisa

Para participar deste estudo, você será informado sobre qualquer aspecto que desejar e o responsável por você deverá autorizar, assinando um termo. Mesmo que seu responsável autorize sua participação, você ainda tem o direito de recusar, estando livre para participar ou não. Lembre-se de que você e/ou seu responsável poderão deixar de participar a qualquer momento, sem nenhum problema.

5) Organização da Pesquisa

Os pesquisadores responsáveis pelo estudo poderão fornecer qualquer esclarecimento sobre o mesmo, assim como tirar dúvidas, bastando entrar em contato pelos seguintes e-mails:

Pesquisador: Robério Girão Pinho - roberiogiraopinho@gmail.com / LearningLab UFC-RUSSAS
Orientadora/Coordenadora: Jacilane de Holanda Rabelo - jacilane.rabelo@ufc.br UFC-RUSSAS

6) Declaração de Consentimento

Li ou alguém leu para mim as informações contidas neste documento antes de assinar este termo de consentimento. Declaro que concordo em participar desse estudo.

Local e Data:

Participante	Orientador
Nome: _____ Assinatura: _____	Nome: Jacilane de Holanda Rabelo Assinatura: _____