

COLÉGIO PEDRO II

Pró-Reitoria de Pós-Graduação, Pesquisa, Extensão e Cultura
Especialização em Educação Matemática

Guilherme Polillo Filho

**ANÁLISE DE FLUXOGRAMAS E CRIAÇÃO DE
ALGORITMOS PARA ESTUDANTES DA EDUCAÇÃO BÁSICA NO
SÉCULO XXI:**

Atividades de contagem que utilizam o Scratch como artefato

Rio de Janeiro
2021



Guilherme Polillo Filho

**ANÁLISE DE FLUXOGRAMAS E CRIAÇÃO DE ALGORITMOS PARA
ESTUDANTES DA EDUCAÇÃO BÁSICA NO SÉCULO XXI:**

Atividades de contagem que utilizam o Scratch como artefato

Monografia de Especialização apresentada ao Programa de Especialização em Educação Matemática, vinculado à Pró-Reitoria de Pós-Graduação, Pesquisa, Extensão e Cultura do Colégio Pedro II, como requisito parcial para obtenção do título de Especialista em Educação Matemática.

Orientador: Prof. Dr. Daniel Felipe Neves Martins

Rio de Janeiro
2022

COLÉGIO PEDRO II

PRÓ-REITORIA DE PÓS-GRADUAÇÃO, PESQUISA, EXTENSÃO E CULTURA

BIBLIOTECA PROFESSORA SILVIA BECHER

CATALOGAÇÃO NA FONTE

P768 Polillo Filho, Guilherme

Análise de fluxogramas e criação de algoritmos para estudantes da educação básica no século XXI: atividades de contagem que utilizam o SCRATCH como artefato / Guilherme Polillo Filho. - Rio de Janeiro, 2022.

62 f.

Trabalho de Conclusão de Curso (Especialização em Educação Matemática) – Colégio Pedro II, Pró-Reitoria de Pós-Graduação, Pesquisa, Extensão e Cultura.

Orientador: Daniel Felipe Neves Martins.

1. Matemática – Estudo e ensino. 2. Teoria da aprendizagem significativa. 3. Letramento midiático. 4. Formação docente. I. Martins, Daniel Felipe Neves. II. Colégio Pedro II. III Título.

CDD 510

Ficha catalográfica elaborada pela Bibliotecária Simone Alves – CRB-7: 5692.

**ANÁLISE DE FLUXOGRAMAS E CRIAÇÃO DE ALGORITMOS NA
EDUCAÇÃO MATEMÁTICA NO SÉCULO XXI:**

Atividades de contagem que utilizam o Scratch como artefato

Monografia de Especialização apresentada ao Programa de Especialização em Educação Matemática, vinculado à Pró-Reitoria de Pós-Graduação, Pesquisa, Extensão e Cultura do Colégio Pedro II, como requisito parcial para obtenção do título de Especialista em Educação Matemática.

Aprovado em: 07/05/2022

Banca Examinadora:

Prof. Dr. Daniel Felipe Neves Martins (Orientador)
PPGEDMAT - CP II

Prof. M. Rony Henrique Barros (Membro interno)
PPGEDMAT - CP II

Prof. Dr. Haroldo Costa Silva Filho (Membro externo)
CEFET - RJ

Rio de Janeiro
2022

AGRADECIMENTOS

Ao meu pai, minha mãe e meu irmão, que sempre valorizaram a educação e me deram suporte para chegar aonde estou. Amo todos vocês!

Aos meus colegas da pós-graduação Raphael Martins, Ana Luiza Cardoso e Anderson.

Aos professores do Curso de Especialização em Educação Matemática do Colégio Pedro II de 2019, por trazer novos olhares sobre o processo de ensino e aprendizagem da Matemática. Em especial ao amigo e Professor Daniel Martins, cuja dedicação, competência e parceria tornaram nossos encontros presenciais e virtuais momentos ricos de troca e amizade.

Aos professores Dr. Daniel Felipe Neves Martins (Orientador), M. Rony Henrique Barros (CPII – Membro interno) e Dr. Haroldo Costa Silva Filho (CEFET-RJ – Membro interno) por aceitarem compor a banca examinadora e trazerem ricas contribuições através da leitura desta pesquisa.

Ao meu pai Guilherme Polillo e à minha mãe
Tânia Lúcia Carvalho Polillo.

A maior vantagem competitiva é a habilidade de aprender.

Seymour Papert

RESUMO

POLILLO FILHO, Guilherme. Análise de Fluxogramas e Criação de Algoritmos na Educação Matemática no Século XXI: Atividades de contagem que utilizam o Scratch como artefato. 2022. 59f. Monografia (Especialização) – Colégio Pedro II, Pró-Reitoria de Pós-Graduação, Pesquisa, Extensão e Cultura, Programa de Especialização em Educação Matemática, Rio de Janeiro, 2022.

A capacidade de sintetizar informações é uma habilidade que permite que as pessoas observem, analisem, classifiquem, discutam, concluam, testem e solucionem problemas. O formato metodológico no qual é possível enxergar essa síntese é o pensamento computacional, que ressalta o uso de tecnologias digitais, a fim de criar soluções eficazes para tipos específicos de problemas. Sob esse viés, a criação de algoritmos é uma aliada no estabelecimento de fluxogramas para a resolução de um ou de vários tipos de problema. A democratização dos computadores e da internet é o ponto de partida para este projeto. Na educação, o pensamento computacional tem contribuído para a adaptação das salas de aula às necessidades da sociedade contemporânea. A transformação digital trazida pelas novas tecnologias exige que as pessoas se tornem mais capazes de lidar com as ferramentas e a linguagem da cultura digital, incluindo professores e alunos. Países de primeiro mundo, como Estados Unidos, trabalham há algum tempo com o pensamento computacional ainda na Educação Básica. Podemos citar o Scratch como um atual exemplo, que traz a abordagem do pensamento criativo, sem falar nos aplicativos disponíveis para videoconferência, como Google Meets, Zoom, Skype, entre outros, que permitiram a realização de aulas a distância em tempos de covid-19. Embora seja um projeto que não tenha sido realizado com estudantes, acredita-se que sua parte prática é de caráter atemporal e necessária para os dias atuais. A intenção deste trabalho é fazer com que atividades de pensamento computacional, plugadas ou desplugadas, não obstante à Base Nacional Comum Curricular, sejam inseridas na Educação Básica.

Palavras-chave: Pensamento computacional. Pensamento criativo. Aprendizagem significativa. Formação de professores. Tecnologias digitais. Letramento digital.

ABSTRACT

POLILLO FILHO, GUILHERME. Flowchart Analysis and Algorithmic Creation in 21st Century Mathematics Education: Counting activities that use Scratch as an artifact. 2022. 63f. Monograph (Specialization) - Colégio Pedro II, Pró-Reitoria de Pós-Graduação, Pesquisa, Extensão e Cultura, Programa de Especialização em Educação Matemática, Rio de Janeiro, 2022.

The ability to synthesize information is a skill that allows people to observe, analyze, classify, discuss, conclude, test and solve problems. The methodological format in which it is possible to see this synthesis is computational thinking, which emphasizes the use of digital technologies in order to create effective solutions to specific types of problems. Under this bias, creating algorithms is an ally in establishing flowcharts for solving one or several types of problem. The democratization of computers and the internet is the starting point for this project. In education, computational thinking has contributed to the adaptation of classrooms to the needs of contemporary society. The digital transformation brought about by new technologies requires people to become more capable of dealing with the tools and language of digital culture, including teachers and students. First world countries, such as the United States, have been working for some time with computational thinking still in Basic Education. We can cite Scratch as a current example, which brings the approach of creative thinking, not to mention the applications available for video conferencing, such as Google Meets, Zoom, Skype, among others, which allowed the realization of lessons at distance in times of covid-19. Although it is a project that has not been carried out with students, it is believed that its practical part is timeless and necessary for the present day. The intention of this work is to make computational thinking activities, plugged or unplugged, notwithstanding the National Common Curriculum Base, be inserted in Basic Education.

Keywords: Computational thinking. Creative thinking. Meaningful learning. Formação de professores. Digital technologies. Digital literacy.

ABSTRACT EM ITALIANO

POLILLO FILHO, GUILHERME. Analisi dei diagrammi di flusso e creazione algoritmica nell'educazione matematica del XXI secolo: attività di conteggio che utilizzano Scratch come artefatto. 2022. 59f. Monografia (Specializzazione) - Colégio Pedro II, Pró-Reitoria de Pós-Graduação, Pesquisa, Extensão e Cultura, Programa de Especialização em Educação Matemática, Rio de Janeiro, 2022.

La capacità di sintetizzare informazioni è un'abilità che permette alle persone di osservare, analizzare, classificare, discutere, concludere, testare e risolvere i problemi. Il formato metodologico in cui è possibile vedere questa sintesi è il pensiero computazionale, che enfatizza l'uso delle tecnologie digitali per creare soluzioni efficaci a specifici tipi di problemi. Sotto questo bias, la creazione di algoritmi è un alleato nello stabilire diagrammi di flusso per risolvere uno o più tipi di problemi. La democratizzazione dei computer e di Internet è il punto di partenza di questo progetto. Nell'educazione, il pensiero computazionale ha contribuito all'adattamento delle aule alle esigenze della società contemporanea. La trasformazione digitale portata avanti dalle nuove tecnologie richiede che le persone diventino più capaci di affrontare gli strumenti e il linguaggio della cultura digitale, inclusi insegnanti e studenti. I paesi del primo mondo, come gli Stati Uniti, hanno lavorato per qualche tempo con il pensiero computazionale ancora nell'istruzione di base. Possiamo citare Scratch come un esempio attuale, che porta l'approccio del pensiero creativo, per non parlare delle applicazioni disponibili per le videoconferenze, come Google Meets, Zoom, Skype, tra gli altri, che hanno permesso la realizzazione di lezioni a distanza in tempi di covid-19. Anche se si tratta di un progetto che non è stato realizzato con gli studenti, si ritiene che la sua parte pratica è senza tempo e necessario per i giorni nostri. L'intenzione di questo lavoro è quello di rendere le attività di pensiero computazionale, collegato o scollegato, nonostante il National Common Curriculum Base, essere inserito nella formazione di base.

Parole chiave: Pensiero computazionale. Pensiero creativo. Apprendimento significativo. Formazione degli insegnanti. Tecnologie digitali. Alfabetizzazione digitale.

LISTA DE FIGURAS

Figura 1 – Mentalidade de crescimento.....	8
Figura 2 – <i>print</i> da tela de um novo projeto (Scratch).....	11
Figura 3 – Projeto Scratch que determina se um número natural é par ou ímpar.....	13
Figura 4 – Projeto Scratch que determina a quantidade de “caras” e “coroas” de n lançamentos de moedas, sendo n natural.....	14
Figura 5 – Projeto Scratch que calcula o fatorial de um número natural.....	15
Figura 6 – Algoritmo Scratch que lista os divisores de um número natural.....	18
Figura 7 – Fluxograma para listar os divisores de um número natural.....	20
Figura 8 – Espiral de Fibonacci.....	24
Figura 9 – Projeto Scratch que lista os primeiros n termos da sequência de Fibonacci, sendo n um número natural.....	25
Figura 10 – Fluxograma que lista os primeiros n termos da sequência de Fibonacci...	27
Figura 11 – Projeto Scratch que lista os primeiros n primos, sendo n um número natural.....	32
Figura 12 – Fluxograma que exhibe os primeiros n primos, sendo n um número natural.....	35
Figura 13 – Gráfico da complexidade de tempo da notação <i>Big-O</i>	37

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
BNCC	Base Nacional Comum Curricular
IBGE	Instituto Brasileiro de Geografia e Estatística
MIT	Massachusetts Institute of Technology
PNAD	Pesquisa Nacional por Amostra de Domicílios

SUMÁRIO

1.	INTRODUÇÃO	3
2.	REVISÃO DE LITERATURA.....	5
3.	SCRATCH	10
3.1.	FUNCIONALIDADES.....	11
3.2.	APLICAÇÕES DIRETAS	13
3.2.1.	PAR OU ÍMPAR?.....	13
3.2.2.	LANÇAMENTO DE MOEDA	14
3.2.3.	FATORIAL DE UM NÚMERO NATURAL	15
4.	ATIVIDADES TECNOLÓGICAS CRIADAS COM BASE NO SCRATCH.....	17
4.1.	ATIVIDADE DOS DIVISORES NATURAIS DE UM NÚMERO NATURAL	17
4.1.1.	CONSTRUÇÃO DA ATIVIDADE.....	18
4.1.2.	AOS PROFESSORES	20
4.1.3.	SUGESTÕES DE ABORDAGEM.....	22
4.1.4.	SUGESTÕES DE RESPOSTA.....	22
4.2.	ATIVIDADE DA SEQUÊNCIA DE FIBONACCI.....	24
4.2.1.	CONSTRUÇÃO DA ATIVIDADE.....	26
4.2.2.	AOS PROFESSORES	27
4.2.3.	SUGESTÕES DE ABORDAGEM.....	29
4.2.4.	SUGESTÕES DE RESPOSTA.....	29
4.3.	ATIVIDADE DOS NÚMEROS PRIMOS	32
4.3.1.	CONSTRUÇÃO DA ATIVIDADE.....	33
4.3.2.	AOS PROFESSORES	34
4.3.3.	SUGESTÕES DE ABORDAGEM.....	36
4.3.4.	SUGESTÕES DE RESPOSTA.....	37
5.	COMPLEXIDADE DE UM ALGORITMO	38
6.	RESULTADO DA PESQUISA.....	40
7.	CONSIDERAÇÕES FINAIS.....	48
8.	REFERÊNCIAS	50
	ANEXO A	51
	ANEXO B.....	52
	ANEXO C.....	53

1. INTRODUÇÃO

Com a confirmação de que as Tecnologias Digitais fazem parte da Base Nacional Comum Curricular, doravante BNCC, há a preocupação de verificar a democratização das ferramentas e dos conhecimentos necessários para sua aplicação na Educação no século XXI. Em 2019, a Pesquisa Nacional por Amostra de Domicílios (PNAD) registrou um aumento de 4,8% em relação a 2018 quanto ao uso da internet, totalizando 78,3% de brasileiros com 10 anos de idade ou mais. De 21,7% da população que não usa a internet, 75,4% alegaram não acessar por não saberem usá-la ou, simplesmente, por falta de interesse, e, dos que a utilizam, dentre os aparelhos utilizados – telefone móvel celular, microcomputador, televisão e *tablet* – 98,6% acessam à internet pelo celular, 46,2% pelo computador, 31,9% pela televisão e 10,9% pelo *tablet*. A pesquisa também registrou um aumento no uso da internet pelas pessoas com 60 anos ou mais. Em 2018, a porcentagem era de 38,7%; já em 2019, essa porcentagem pulou para 45,0%.

Na internet encontramos diversas ferramentas educacionais e muitas outras instituições que criam um ambiente voltado para a aprendizagem, como é o caso da Google que criou o Google Classroom, uma ferramenta *on-line* e gratuita que auxilia professores, estudantes e escolas com um espaço onde é possível realizar atividades, aulas virtuais, bem como trabalhar com ferramentas que podem vir a ser úteis no futuro, como formulários, planilhas, documentos, apresentações, entre outras. Em cenário à pandemia Covid-19 também podemos destacar o aumento do uso de aplicativos como Google Meets, Zoom, Moodle, Skype e outros que permitem videochamadas com múltiplas janelas em tempo real.

Muitas instituições de ensino já utilizaram essas ferramentas, permitindo, assim, que a educação se relacione com o mundo do trabalho, visto que esse uso durante o processo educativo torna o aprendizado significativo, já que todos os artefatos utilizados farão parte de um futuro que envolve o que foi denominado como Educação 5.0, e que traz desafios para os alunos que não possuem tecnologia disponível e para os professores que não tiveram uma formação didático-tecnológica.

Esse projeto foi pensado com base em atividades voltadas para o Ensino Fundamental que promovam a desenvoltura dos quatro pilares do pensamento computacional: decomposição, abstração, reconhecimento de padrões e criação de algoritmos, no Scratch – uma linguagem de programação criada por Mitchel Resnick e pelo grupo Lifelong Kindergarten do Media Lab na universidade americana MIT – que vem sofrendo atualizações constantes desde 2007,

promovendo, assim, um melhor funcionamento no momento de criar e compartilhar algoritmos interativos. Por meio da ferramenta é possível registrar um fluxograma que caracteriza o passo a passo do que ocorre em nossas mentes na hora de executar um processo repleto de desvios lógicos, isto é, através de uma entrada é possível obter, como saída, a solução desejada em uma velocidade computacional maior do que estamos acostumados a realizar no papel. O *site* também permite que os usuários conversem entre si, trocando informações e ideias acerca dos algoritmos expostos que envolvem desde a ordenação dos blocos de comando até a visualização da movimentação de elementos na tela de maneira inteiramente criativa.

Mas, como o Scratch pode auxiliar os professores e os alunos do Ensino Fundamental trazendo para a escola, não só o letramento digital, como também a autonomia, a criatividade, a colaboração, a comunicação e o raciocínio lógico dedutivo?

Neste projeto, por conta do baixo índice de formação didático-tecnológica, e pela necessidade da introdução da Tecnologia na Educação exposta na BNCC, busco difundir a ferramenta bem como algumas de suas potencialidades e alguns exemplos de aplicação dentro ou fora da sala de aula. O trabalho é voltado para uma ferramenta tecnopedagógica já presente em diversos países e em algumas escolas do Brasil – o Scratch – e de sua grandiosidade para com o desenvolvimento do pensamento computacional a partir da educação básica. O projeto tem como propósito o incentivo inicial para professores. Primeiramente, estes, precisam se apropriar da ferramenta, desenvolver atividades e compreenderem suas potencialidades para que depois possam pensar em avaliar seus estudantes.

O capítulo 2 é um condensado dos ideais e das ideias de algumas das autoridades no assunto e o resultado de uma pesquisa feita com os alunos do Curso de Especialização do Colégio Pedro II em 2021 que buscava entender o domínio/conhecimento de ferramentas digitais. O capítulo 3 busca fatores que podem motivar os professores a iniciar o uso das ferramentas disponíveis no Scratch. São explicitados, também, os comandos que foram utilizados para a criação das atividades deste projeto e alguns exemplos de aplicação direta. A partir do capítulo 4 são encontradas atividades um pouco mais avançadas, voltadas para a formação de professores que queiram se aprofundar e aplicar uma construção mais elaborada durante as aulas. No capítulo 5 encontram-se as considerações finais, enquanto as referências se encontram no capítulo 6, e os resultados da pesquisa realizada com os alunos da Pós-graduação em Educação Matemática do Colégio Pedro II, no capítulo 7.

2. REVISÃO DE LITERATURA

A educação tem apresentado uma constante transformação ocasionada pelo avanço das tecnologias e agravada pelo impacto da pandemia da Covid-19 que começou em meados de 2020. Muitas escolas e professores precisaram acompanhar essa evolução tecnológica e garantir um ensino-aprendizagem de qualidade no qual os estudantes tenham acesso a uma bagagem relevante para um futuro cada vez mais conectado com as novas tecnologias.

A Educação 5.0 surgiu a partir da evolução da Educação 4.0, – proposta que insere a tecnologia na educação – mas, agora, além de ser um método no qual a tecnologia está inserida no processo pedagógico das escolas, a Educação 5.0 visa um ensino mais humano a partir de competências socioemocionais como: colaboração, pensamento crítico, pensamento criativo, comunicação e empatia, que são destaques no processo de fomentar a geração de soluções que melhorem a interação entre pessoas, contribuindo, assim, na construção de um caráter cidadão. É importante mencionar que a Educação 5.0 não anula o significado da Educação 4.0, e, sim, complementa a ideia inicial buscando promover baixo impacto ambiental, o que favorece à saúde e a segurança.

A BNCC reforça o tema quando menciona a resolução de problemas através das tecnologias digitais e ao mencionar a importância do Pensamento Computacional, do Mundo Digital e da Cultura Digital. A primeira refere-se à capacidade de compreender, analisar, definir, modelar, resolver, comparar e automatizar problemas e suas soluções, de forma metódica e sistemática, por meio do desenvolvimento de algoritmos. A segunda envolve as aprendizagens relativas às formas de processar, transmitir e distribuir a informação de maneira segura e confiável em diferentes artefatos digitais – tanto físicos (computadores, celulares, *tablets* etc.) como virtuais (internet, redes sociais e nuvens de dados, entre outros) –, compreendendo a importância contemporânea de codificar, armazenar e proteger a informação. A última envolve aprendizagens voltadas para uma participação mais consciente e democrática por meio das tecnologias digitais, o que supõe a compreensão dos impactos da revolução digital e dos avanços do mundo digital na sociedade contemporânea, a construção de uma atitude crítica, ética e responsável em relação à multiplicidade de ofertas midiáticas e digitais, aos usos possíveis das diferentes tecnologias e aos conteúdos por elas veiculados e, também, à fluência no uso da tecnologia digital para expressão de soluções e manifestações culturais de forma contextualizada e crítica.

Na verdade, esse assunto não é novo. Ao final da década de 1970, Seymour Papert – responsável por ser um dos fundadores do laboratório de inteligência artificial MIT e pelo desenvolvimento da linguagem computacional Logo – escrevia o “*Mindstorms: children, computers, and powerful ideas*”, um livro que tratava da “filosofia Logo” – em que a criança programa o computador e não o contrário –, surgida graças ao grande avanço no movimento de inclusão da tecnologia na educação na qual a aprendizagem é baseada em projetos.

Em muitas escolas de hoje, a frase "ensino assistido por computador" significa fazer o computador ensinar a criança. Pode-se dizer que o computador está sendo usado para programar a criança. Na minha visão, a criança programa o computador e, ao fazê-lo, ambos adquirem um sentido de mestre sobre uma peça da mais moderna e poderosa tecnologia e estabelecem um contacto íntimo com algumas das mais profundas ideias da ciência, da matemática, e da arte de construir modelos intelectuais. (PAPERT, 1980)

No início dos anos oitenta, o livro tornou-se uma das obras mais conhecidas sobre o tema – o lançamento, no Brasil, só ocorreu em 1985 (Cysneiros, 1991). O sucesso foi tanto que Papert desenvolveu um brinquedo em parceria com a LEGO, introduzindo motores, engrenagens e sensores nos blocos tradicionais, possibilitando, assim, o controle dinâmico através de simples algoritmos de programação criados no Logo. A linguagem utilizada faz parecer que o livro não foi escrito apenas para a comunidade acadêmica, mas sim para provocar e incentivar a criatividade do leitor.

Em seu livro *Lifelong kindergarten*, Mitchel Resnick mostra não ter apenas defendido a ideia de Papert como também aprimorado. Ao criar os 4P’s da aprendizagem criativa, reforça a ideia de que todos conseguem desenvolver o Pensamento Criativo através de explorações criativas que sejam de seus próprios interesses (*passion*). As descrições dos objetivos a serem alcançados pelos 4P’s podem ser encontradas na página 21.

Em nosso programa de pós-graduação do MIT Media Lab, voltado ao uso criativo de novas tecnologias, adotamos uma abordagem inspirada no jardim de infância: os estudantes passam pouquíssimo tempo na sala de aula. Em vez disso, eles estão sempre trabalhando em projetos, orientados pela espiral da aprendizagem criativa. (RESNICK, 2017, p.13).

Além disso, segundo Resnick (2017), as afirmações “a criatividade está restrita à expressão artística”, “apenas uma pequena parte da população é criativa”, “a criatividade é uma ideia que surge como um raio e a criatividade é algo que não se pode ensinar” são falsas e a criatividade pode, sim, ser ensinada a alguém.

(...) podemos nutrir a criatividade. Todas as crianças nascem com a capacidade de serem criativas, mas essa criatividade não se desenvolverá necessariamente, sozinha. Ela precisa ser nutrida, incentivada, apoiada. O processo é semelhante ao trabalho de um jardineiro que cuida de suas plantas, criando um ambiente no qual elas possam florescer. Da mesma forma, podemos criar um ambiente de aprendizagem onde a criatividade floresça. (RESNICK, 2017, p. 20).

Diferentemente da abordagem behaviorista¹, quando trabalhamos com o que realmente gostamos, retemos um maior conhecimento para longo prazo. Segundo Daniel Pink em seu livro *Motivação 3.0 – Drive: A surpreendente verdade sobre o que realmente nos motiva*, as recompensas têm um efeito temporário. É como a cafeína que nos deixa elétricos por algumas horas, porém o efeito passa e, no pior dos casos, pode reduzir a motivação principal da pessoa de dar continuidade a um projeto que não era realmente do seu próprio interesse.

Um projeto nasce da necessidade de resolução de um problema e, desde o começo, é importante que cada membro do grupo busque sentido em suas soluções, fugindo de métodos convencionais e buscando investigar novas formas de resolução, que, segundo Jo Boaler, em seu livro *Mentalidades matemáticas*, é a transformação de um afazer matemático em uma tarefa investigativa. Com isso, ao propormos um problema a ser resolvido sem que os estudantes conheçam o “método adequado” de resolução, estamos trabalhando com a criatividade e intuição, em que eles irão utilizar seus conhecimentos prévios para desenvolver um método de resolução para o problema abordado. Segundo a autora, uma das ferramentas mais poderosas para a resolução de um problema é o componente visual desse problema. Esse componente auxilia o processo pedagógico, aumentando, assim, a compreensão de um problema pelos estudantes, podendo ser um diagrama ou um objeto físico. Portanto, desenhar durante o processo investigativo ajuda o estudante a conectar-se com o problema e também reforça que essas tarefas investigativas precisam ter “pisos baixos” e “tetos altos” – estes indicam as duas etapas do processo de resolução em que os estudantes se encontram. Na primeira etapa é necessário que o estudante diga como enxerga o problema, o que parece ser acessível e quais são os desafios a serem enfrentados. No segundo caso ele é convidado a fazer uma pergunta mais difícil, elevando, assim, o nível de dificuldade do problema para si. A última questão abordada pela autora inclui a necessidade de convencer um cético através de argumentos. Nessa atividade um dos estudantes precisa convencer, enquanto o outro é cético e, depois de um

¹ Conjunto de teorias psicológicas que postulam o comportamento como o mais adequado objeto de estudo da psicologia.

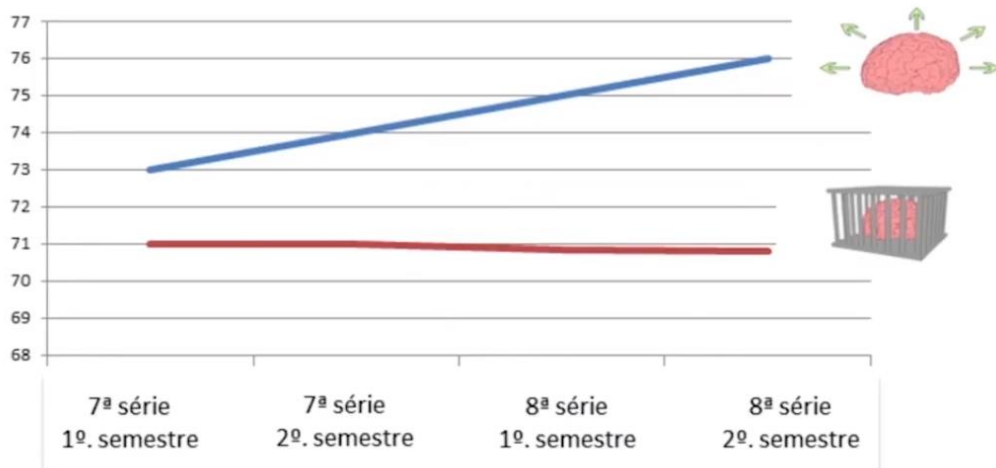
tempo, os papéis se invertem, trabalhando, assim, o senso crítico dos alunos além de instigá-los.

A ideia de “pisos baixos” e “tetos altos” é enfatizada por Papert e, segundo Resnick, seu grupo *Lifelong Kindergarten* criou uma nova dimensão: “paredes amplas”, que tem o intuito de desenvolver tecnologias que apoiem e proponham uma maior variedade de projetos.

Oferecer um único caminho do piso baixo ao teto alto não basta, é importante proporcionar vários caminhos. Por quê? Queremos que todas as crianças trabalhem em projetos baseados em suas próprias paixões e interesses pessoais, e como crianças diferentes têm paixões diferentes, precisamos de tecnologias compatíveis com vários tipos de projetos. Dessa forma, todas podem trabalhar em projetos que sejam pessoalmente relevantes para elas. (RESNICK, 2017, p. 60).

Boaler (2014) também reforça a ideia de que quando erramos durante o processo de aprendizagem, mais sinapses são disparadas em nossos cérebros e isso fortalece os caminhos neurais; enquanto ao acertarmos, há menos atividade cerebral. O *Estudo dos Erros* foi feito por Jason Moser e conta com a ressonância magnética registrada durante as provas. Nesse ponto, dois tipos de mentalidades podem ser destacadas: mentalidade de crescimento e mentalidade fixa. Uma pesquisa com estudantes da 7ª série foi feita do início do primeiro semestre até que chegassem ao final do segundo semestre da 8ª série. Os alunos que mais acreditavam em seu potencial (mentalidade de crescimento), apesar dos erros, foram os que tiveram um melhor resultado ao longo dos quatro semestres (a barra vertical indica as notas de Matemática que foram atribuídas aos estudantes).

Figura 01 – Mentalidade de crescimento



Fonte: Jo Boaler.

Com o intuito de pensar em maneiras de estimular a mentalidade de crescimento, a autora mostra como as dicotomias podem ser prejudiciais: Inteligentes ou não? Superdotados ou não?

Segundo Leslie (2015), os professores acreditam na ideia de “dom” e, de acordo com pesquisas, quanto mais o conceito de “dom” é valorizado em uma área do conhecimento, menos mulheres e estudantes afro-americanos estão nesse campo. Por isso, a formação de professores é um dos pontos-chave para a melhora do ensino-aprendizagem. A introdução desses conceitos, a formação digital e a pesquisa por novas metodologias de ensino precisam sempre perpetuar os métodos de avaliação da aprendizagem. Os professores precisam estar atualizados quanto às tecnologias para lidar com estudantes cada vez mais espertos frente ao mundo digital.

Uma pesquisa foi realizada através de um formulário Google na Pós-graduação em Educação Matemática do Colégio Pedro II e buscou mapear o nível de familiaridade (de 1 a 5, sendo 5 um alto domínio da ferramenta) dos professores e alunos da pós-graduação, totalizando 48 respostas para com as seguintes habilidades digitais: Saber administrar tarefas simples no computador (Salvar *links*, criar pastas, copiar e colar arquivos, etc.); Conseguir utilizar *sites*/aplicativos/jogos educativos (exemplos: Kahoot, Plickers, etc.); Moodle, Google Sala de aula, Google Drive, Google Apresentações, Google Planilhas, Google Documentos, Google Formulários, Google Meets, Zoom, Excel, Word, Power Point, Skype, GeoGebra e Scratch. A pesquisa encontra-se na página 49 e pode ser acessada pelo *link*: <<https://forms.gle/2NFwmkSZ9eDA9Xxc8>>. Disponível em: 19 mar. 2022.

A partir da pesquisa é possível concluir que o Pensamento computacional já é uma ideia conhecida por mais da metade dos professores da pós-graduação (60,4%). Do total, 89,6% dos professores têm mais de 5 anos de experiência na Educação Básica. Ainda do total, 72,9% alegam poder usar os computadores em sala de aula, 25% afirmam que, embora existam computadores, não podem utilizar com os estudantes e apenas um dos professores alegou não existir computadores no ambiente escolar. Apenas 25% dos professores trabalham apenas em escolas particulares, 60,4% trabalham em escolas públicas e 14,6% trabalham em ambas as redes. Das ferramentas analisadas, a única que apresentou os níveis decrescentes quanto ao domínio (1 a 5) foi o Scratch.

3. SCRATCH

O Scratch, como já mencionado anteriormente, é uma linguagem de programação criada pelo grupo Lifelong Kindergarten do MIT, na qual a codificação é feita através de blocos de comando. A ferramenta teve como inspiração as peças LEGO, cuja empresa reconheceu a linguagem e suas potencialidades ao ponto de criarem *kits* próprios de robótica que podem ser controlados por comandos criados no Scratch. Ao contrário do Kahoot, Resnick acredita que o Scratch é motivador por dar mais liberdade de criação para os estudantes num processo apelidado de Espiral da Aprendizagem Criativa², em que o protagonismo e a autonomia são aspectos fundamentais. Uma das importâncias do Scratch é criar o senso de autorregulação, isso é percebido quando professores, pais e alunos conseguem diferenciar quando um tempo gasto ao acessar uma tela foi útil ou não para o seu aprendizado. O tempo gasto em um jogo violento é diferente do tempo gasto enviando mensagens para os amigos que, por sua vez, é diferente do tempo gasto em uma pesquisa para um trabalho da escola, que é diferente do tempo gasto na criação de um projeto Scratch (RESNICK, 2017, p.23).

Outra ideia de Resnick remete à aprendizagem baseada no “aprender criando”, alegando que é preciso formar “pessoas que criam coisas” e, segundo ele, não é suficiente fazer algo, é necessário criar algo.

Inspirado nas ideias de Seymour, criou-se o conceito de aprendizagem criativa tomando por base a proposição de que a aprendizagem se torna mais eficiente quando envolve a experimentação, colaboração, criatividade e incentiva que os estudantes coloquem a mão na massa. Ou seja, em alguns casos a aprendizagem significativa está relacionada com a tecnologia digital, visto que o futuro depende dela.

A aprendizagem criativa é caracterizada por uma espiral que passa por quatro pilares estabelecidos por Resnick: *Project, passion, peers, play*, em inglês, cuja tradução utilizada no livro *Jardim de Infância para a Vida Toda* é dada como projeto, paixão, parcerias e o pensar brincando.

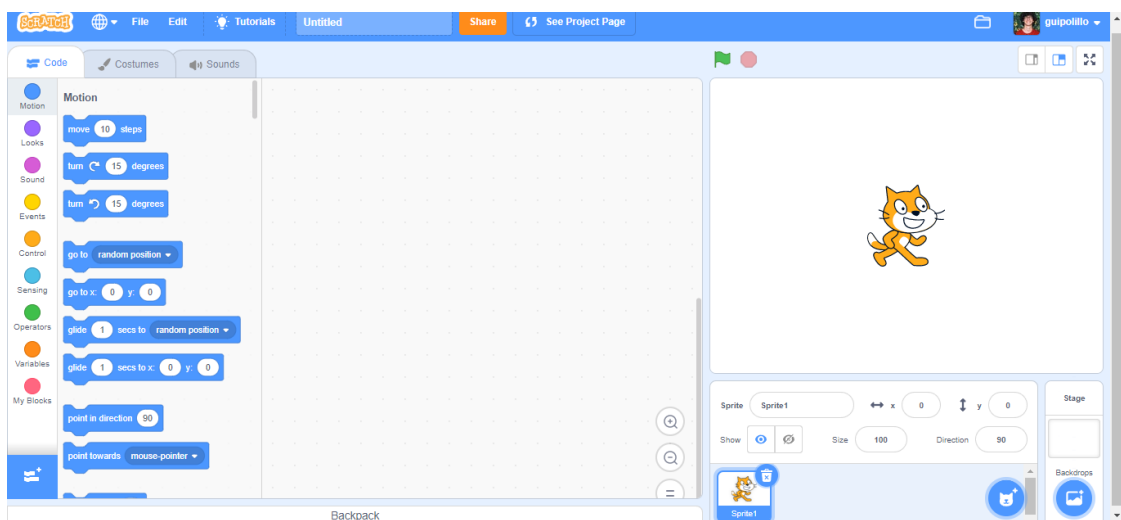
O primeiro P (projeto) engloba todo o planejamento e recursos necessários para a sua confecção. O projeto terá de ser feito com suporte à liberdade dos alunos que, através disso, podem trabalhar o seu protagonismo a ponto de trabalharem com o que desejarem e, nesse momento, surge o segundo P (paixão). A espiral também conta com uma etapa que estimula a

² Processo criativo sustentado pelos pilares: imaginar, criar, brincar, compartilhar e refletir. Propõe uma nova forma de impactar a educação na forma com que crianças, jovens e adultos se relacionam com o seu aprendizado.

cooperação e comunicação fazendo com que os estudantes se separem em grupos ou busquem o *feedback* de outros estudantes (parcerias) como forma de melhorar seus próprios projetos. No momento “mão na massa”, os estudantes devem se sentir livres para experimentarem de forma lúdica (pensar brincando) em como podem tornar o projeto palpável, o que vai de encontro com a premissa de Resnick (2020): “O pensamento criativo é resultado de explorações criativas”.


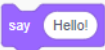
É possível acessar o Scratch e criar uma conta de maneira gratuita através do [link: <https://scratch.mit.edu/>](https://scratch.mit.edu/). Para criá-la, basta clicar na opção “Aderir ao Scratch”, ou similar, no canto superior direito da tela. Após ter criado a conta e iniciar um novo projeto, é possível ter acesso à *interface* de criação a seguir.


Figura 02 – *print* da tela de um novo projeto (Scratch).




Fonte: O autor.

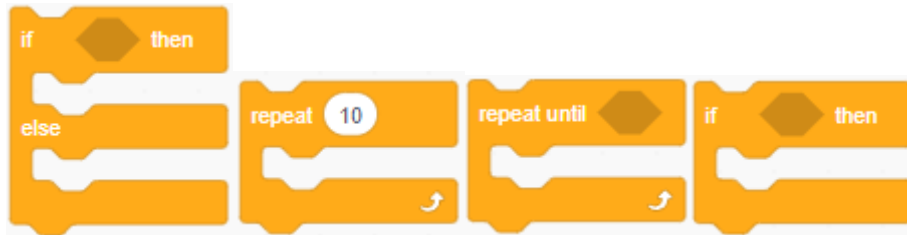
3.1. FUNCIONALIDADES


Pela característica dos conceitos utilizados nesse projeto, apenas as abas “Events”, “Control”, “Sensing”, “Operators” e “Variables” foram utilizadas. A seguir é possível consultar os blocos do Scratch que foram utilizados na elaboração das atividades deste projeto bem como suas funcionalidades. Na aba  é possível fazer com que o objeto interaja com o usuário por meio de uma frase utilizando o seguinte bloco. A frase pode ser programada para responder através de um longo percurso lógico de resolução de problemas dado um ou mais valores iniciais. 

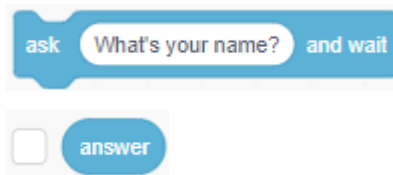
Na aba  utilizamos apenas um comando que depende do “click” humano para iniciar o projeto/ algoritmo:



Na aba  é possível visualizar blocos de comando que foram utilizados para controlar o fluxo do algoritmo através de desvios lógicos condicionais ou repetições baseadas em variáveis e/ou operações lógicas.




Na aba  foram utilizados blocos que perguntam (*ask*) determinada informação para o usuário e armazenam a resposta na variável pré-existente (*answer*).

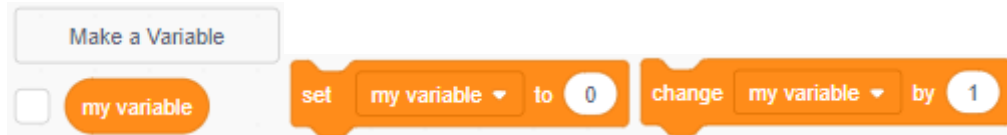


Na aba  encontram-se as operações matemáticas e junções utilizadas.

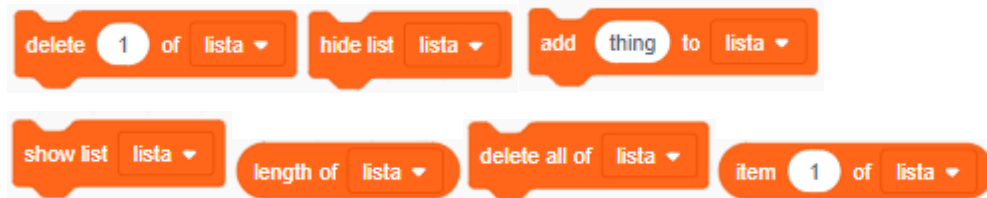


Na aba  é possível criar e controlar o valor de variáveis e quantidade de elementos de uma lista. Ao clicar no quadrado ao lado de “my variable”, a variável irá aparecer junto de seu valor absoluto na tela da direita.

Comandos para variáveis:



Comandos para listas:

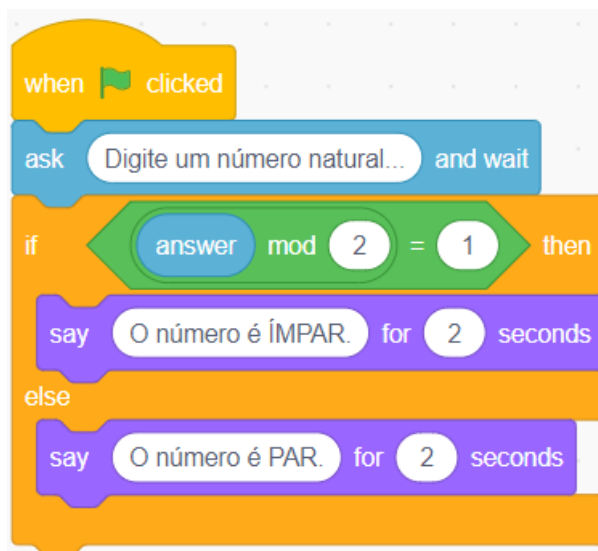


3.2. APLICAÇÕES DIRETAS

3.2.1. PAR OU ÍMPAR?

A aplicação a seguir depende da entrada de um número natural para decidir se ele é par ou ímpar.

Figura 03 – Projeto Scratch que determina se um número natural é par ou ímpar.



Fonte: O autor

A organização do fluxograma provém da linguagem corrente: Quando a bandeira verde for clicada, a aplicação perguntará para o usuário: “Digite um número natural...”. Após o usuário ter inserido um número no visor, e confirmado, o algoritmo irá realizar um teste em que

se o número digitado deixa resto 1 quando dividida por 2, então o personagem diz: “O número é ÍMPAR”, senão “O número é PAR”. Ambas as mensagens ficarão na tela por 2 segundos. Disponível em: <<https://scratch.mit.edu/projects/547179772/>>. Acesso em: 17 fev. 2022.

3.2.2. LANÇAMENTO DE MOEDA

O algoritmo a seguir simula n lançamentos de moeda em velocidade computacional.

Figura 04 – Projeto Scratch que determina a quantidade de “caras” e “coroas” de n lançamentos de moedas, sendo n natural.



Fonte: O autor.

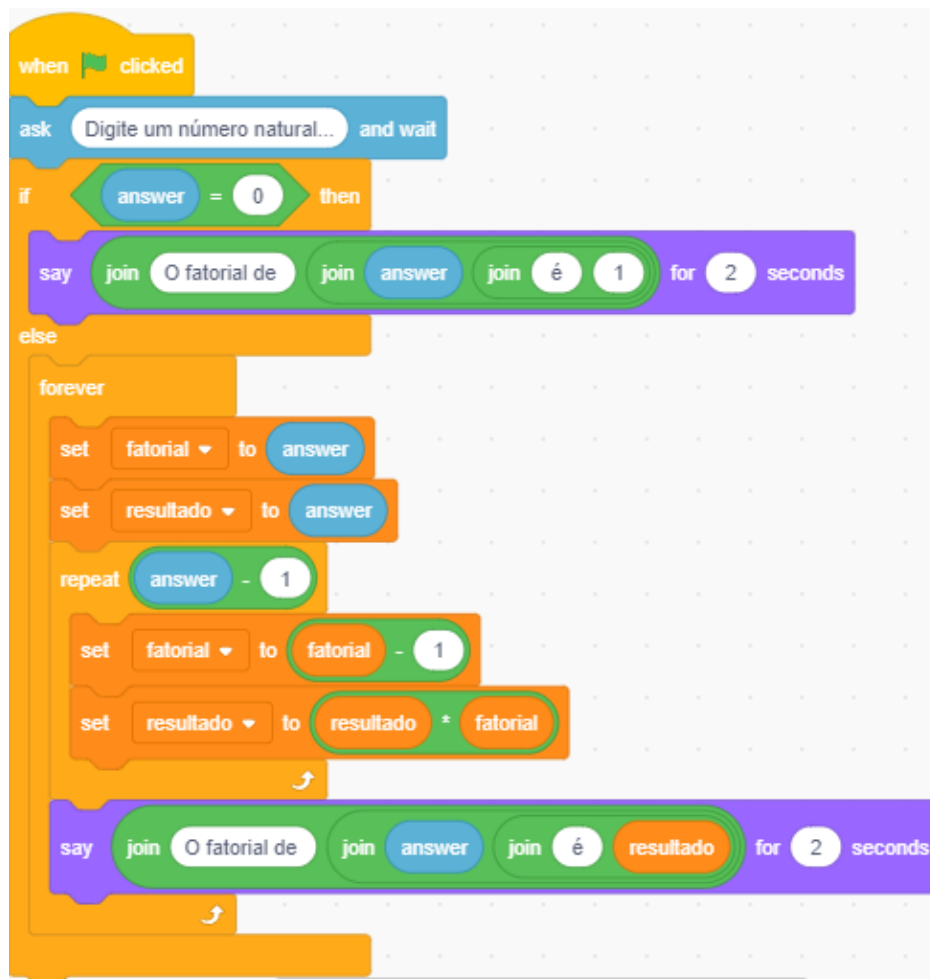
A linguagem corrente pode ser descrita como: quando a bandeira verde é clicada, a aplicação atribui o valor 0 para a variável “cara”, o valor 0 para a variável “coroa” e pergunta para o usuário: “Quantas vezes deseja jogar uma moeda?” Após o usuário ter digitado um valor

n , o processo a seguir se repetirá n vezes: atribuir um número natural entre 1 e 2 (1 e 2 inclusos) para a variável “jogada”. Se a variável jogada for igual a 1, então adicionamos uma unidade à variável “cara”, senão adicionamos uma unidade à variável “coroa”. No final dos lançamentos, é possível visualizar a quantidade de caras e coroas obtidas. Disponível em: <<https://scratch.mit.edu/projects/646400524/>>. Acesso em: 17 fev. 2022.

3.2.3. FATORIAL DE UM NÚMERO NATURAL

O fluxograma a seguir calcula o fatorial de um número natural.

Figura 05 – Projeto Scratch que calcula o fatorial de um número natural.



Fonte: O autor.

A linguagem corrente pode ser descrita como: Ao clicar na bandeira verde, irá ocorrer um *loop*³ a partir da digitação de um número natural e, a seguir, o valor digitado pelo usuário será atribuído à variável “fatorial” e à variável “resultado”, por conseguinte, o próximo passo a passo terá a quantidade de repetições igual ao número digitado pelo usuário menos 1: o novo valor da variável “fatorial” será “fatorial – 1”, o novo valor da variável “resultado” será “resultado * fatorial”. Ao final das repetições, o algoritmo terá feito todo o processo de multiplicação fatorial e imprimirá a resposta: “O fatorial de n é $n!$ ”. Note que, caso o número digitado for zero, o fatorial de zero é um. Disponível em: <<https://scratch.mit.edu/projects/647237558/>>. Acesso em: 19 fev. 2022.

³ Sequência que se repete dentro de um software até que uma condição pré-programada interrompa o laço de repetição ou execute outra função. Também é possível que um erro tenha ocorrido e gerado o *loop*, nesse caso, é necessário pausar o código e reprogramá-lo para evitar o *loop* indesejado.

4. ATIVIDADES TECNOLÓGICAS CRIADAS COM BASE NO SCRATCH

As atividades a seguir foram pensadas para o Ensino Fundamental, todas em torno da aritmética e da matemática discreta, elaboradas de acordo com as propostas da BNCC.

A cultura digital se faz presente no currículo trazendo ideias como a produção de conhecimento, resolução de problemas, protagonismo e autoconhecimento. A construção das competências 4 e 5 da BNCC elucida a necessidade de utilizar diferentes linguagens – verbal (oral ou visual-motora, como Libras, e escrita) corporal, visual, sonora e **digital** –, bem como conhecimentos das linguagens artística, matemática e científica, para se expressar e partilhar informações, experiências, ideias e sentimentos em diferentes contextos e produzir sentidos que levem ao entendimento mútuo; **compreender, utilizar e criar tecnologias digitais de informação e comunicação** de forma crítica, significativa, reflexiva e ética nas diversas práticas sociais (incluindo as escolares) para se comunicar, acessar e disseminar informações, produzir conhecimentos, resolver problemas e exercer protagonismo e autoria na vida pessoal e coletiva.

A primeira atividade envolve a criação de um algoritmo computacional, através do Scratch, que serve como aliado na obtenção dos divisores naturais de um número natural. A segunda atividade imprime a sequência de Fibonacci a partir dos dois primeiros termos. A terceira atividade tem como objetivo mostrar os n primeiros números primos positivos, sendo n um número natural dado pelo usuário.

4.1. ATIVIDADE DOS DIVISORES NATURAIS DE UM NÚMERO NATURAL

Ainda no sexto ano do Ensino Fundamental, a BNCC reforça a ideia de trabalhar, por meio de investigações, os diversos critérios de divisibilidade. Aprende-se que um número é divisível por 2 se for par; um número é divisível por 3 se a soma de seus algarismos for um múltiplo de 3; um número é divisível por 4 se os seus dois últimos algarismos formam um múltiplo de 4; um número é divisível por 5 se o seu último algarismo for 0 ou 5; um número é divisível por 6 se for por 2 e, ao mesmo tempo, por 3; um número é divisível por 7 se a soma do quádruplo da sua unidade com o restante do número resultar em um múltiplo de 7 – critério desenvolvido pelo nigeriano de 12 anos Chika Ofili; um número é divisível por 8 se os seus três últimos algarismos formam um múltiplo de 8; um número é divisível por 9 se a soma de seus algarismos for um múltiplo de 9; um número é divisível por 10 se terminar em 0. Alguns

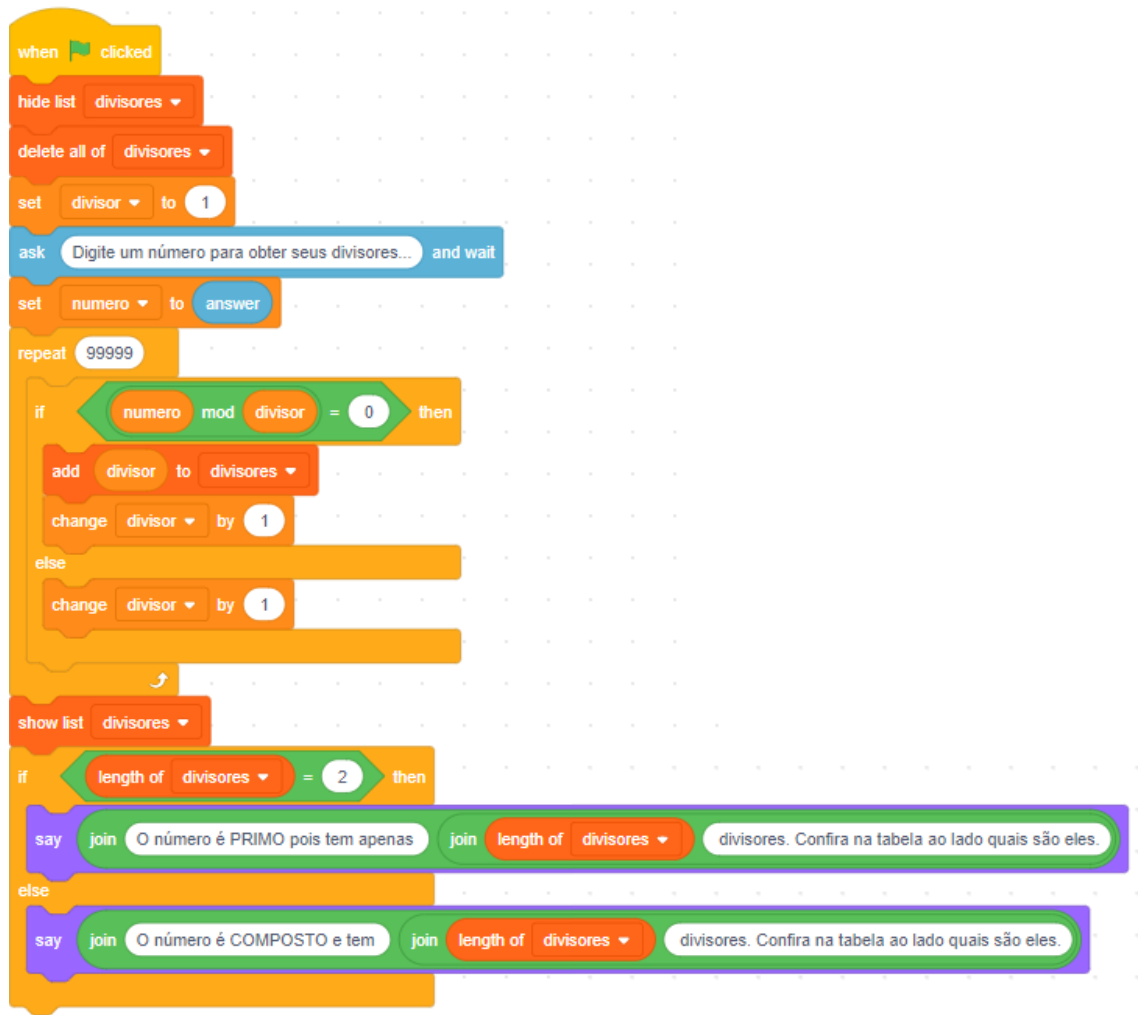
estudantes podem continuar com o raciocínio anterior e acabarem se perguntando sobre como seriam os critérios de divisibilidade por 11, 12, 13 etc.

A atividade nasceu diante dos questionamentos: “Precisamos memorizar todos os critérios de divisibilidade exibidos anteriormente?”, “Com o uso de Tecnologias Digitais, seria possível descobrir os divisores naturais de um número?”. O pensamento computacional surge da necessidade de transpor um pensamento lógico, não só para a máquina, mas, também, ao descrevê-lo em um processo algorítmico, o que vai de encontro com a premissa de WING (2008, p. 3): “Um algoritmo é a abstração de uma sequência de passos utilizados para processar entradas e produzir saídas desejadas”.

4.1.1. CONSTRUÇÃO DA ATIVIDADE

Com a intenção de criar um fluxograma computacional que listasse os divisores de um número, o algoritmo a seguir foi construído utilizando os blocos de comando: lista, repetição, desvio condicional e a função mod.

Figura 06 – Algoritmo Scratch que lista os divisores de um número natural.



Fonte: O autor.

O algoritmo se inicia quando a bandeira verde é clicada. O bloco “hide list” esconde a lista, enquanto o “delete all of divisores” limpa a lista de divisores. O bloco “set divisor to 1” faz com que o teste se inicie pelo número 1. O comando “ask Digite um número para obter seus divisores...” armazena uma informação no bloco “answer”; a função “set numero to answer” faz com que a variável “numero” seja definida para “answer”, isto é, o número escolhido pelo utilizador é armazenado. A função “repeat 99999” faz com que o desvio que vem a seguir se repita 99999 vezes.

O desvio “if numero mod divisor = 0 then add divisor to divisores change divisor by 1 else change divisor by 1” pode ser lido como: se o resto da divisão do “numero” pelo “divisor” for 0, então a variável “divisor” será adicionada na lista “divisores” e, após isso, será acrescida em 1 unidade. Caso contrário, a variável “divisor” apenas será acrescida em 1 unidade.

Independentemente de ter sido adicionado na lista ou não, o teste irá se repetir pelas próximas 99998 vezes. Nota-se que, a cada tentativa, o divisor será acrescido em 1 unidade, ou seja, serão feitos os mesmos testes com os números 2, 3, 4, 5, ..., 99998 e 99999.

Após todos os testes, o comando “show list divisores” faz com que a lista seja mostrada apenas com os divisores que apresentaram resto 0 durante o processo algorítmico. Ao mostrar a lista com os divisores, um desvio condicional é acionado com o intuito de mostrar quando um número é primo, ou não, mostrando, também, a quantidade de divisores do número natural escolhido. A atividade encontra-se *on-line* para eventuais consultas. Disponível em: <<https://scratch.mit.edu/projects/546948678/>>. Acesso em: 20 ago. 2021.

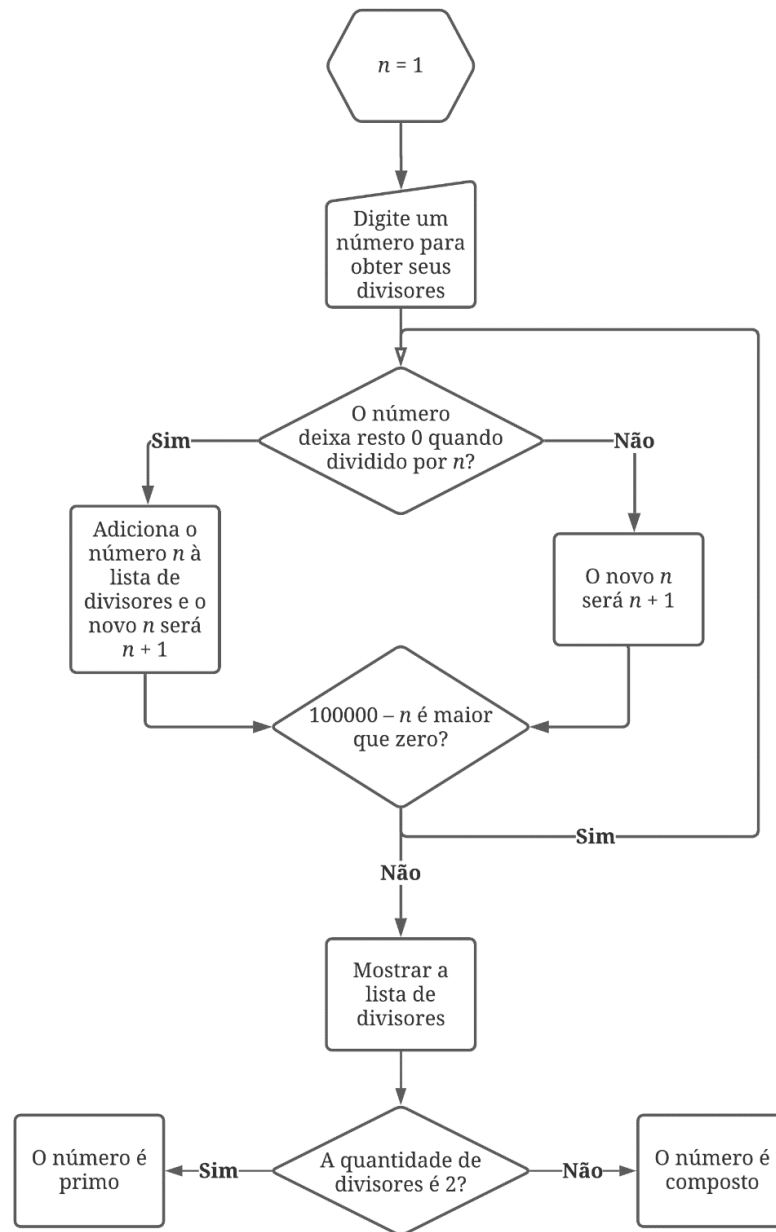
4.1.2. AOS PROFESSORES

Mostrar-se-á necessário fazer com que os estudantes, num primeiro contato, adquiram fluência com os artefatos que os auxiliarão na construção do conhecimento. A capacidade de construção de um algoritmo é elástica, isto é, precisa haver um equilíbrio entre os desafios e as habilidades aprendidas para que os estudantes continuem interessados em expandir suas capacidades o que, segundo Mihaly (1990), é denominado “estado de *flow*”: “Este estado prazeroso pode ser controlado, e não apenas deixado ao acaso, ao nos colocar desafios, ou seja, tarefas que não sejam difíceis nem tão simples para nossas habilidades”.

Como ponto de partida, é crucial iniciar a construção do algoritmo através de um problema simples, como: determinar se um número natural qualquer é par; classificar números naturais em primos e compostos; entender a função mod; estabelecer investigações por meio dos critérios de divisibilidade; resolver e elaborar problemas que envolvam as ideias de múltiplo e divisor; por fim, criar um algoritmo que faça tudo isso, concomitantemente, em apenas um milésimo de segundo.

A utilização de um fluxograma pode ajudar no entendimento do processo, visto que é uma forma visual do que acontece durante a compilação. O esquema a seguir mostra, de forma resumida, os caminhos a serem percorridos a partir da inserção do número natural na primeira atividade (esquema criado no: <www.lucidchart.com>.).

Figura 07 – Fluxograma para listar os divisores de um número natural.



Fonte: O autor.

Mostra-se importante saber ressaltar as inúmeras possibilidades de caminhos a serem percorrido. A figura anterior nos indica apenas uma forma de visualizar o algoritmo da primeira atividade – é o mesmo algoritmo criado no Scratch –, mas desplugado das Tecnologias digitais. Embora ele tenha sido feito no Lucidchart, também seria possível reproduzir esse fluxograma utilizando lápis, papel e borracha.

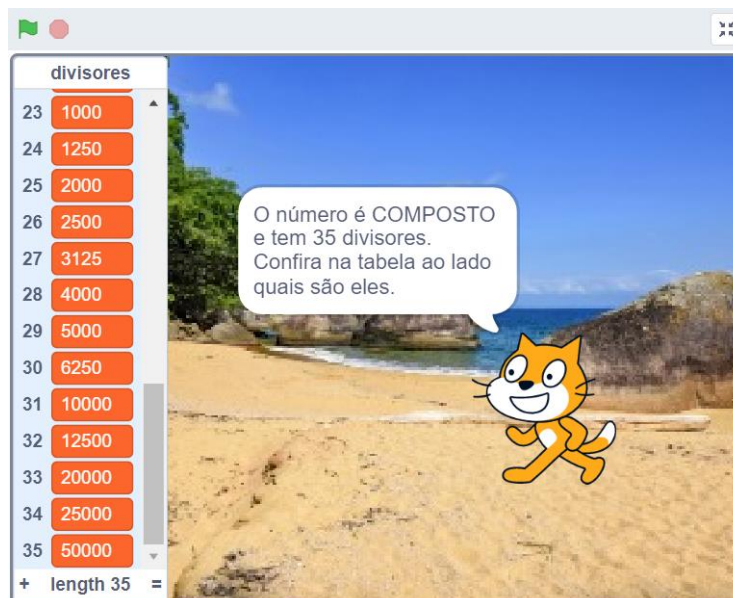
Os fluxogramas permitem que o processo seja melhorado através de *feedbacks*, fazendo com que o arquiteto revise-o buscando falhas ou oportunidades de melhoria, a fim de atualizá-lo para um melhor funcionamento.

4.1.3. SUGESTÕES DE ABORDAGEM

Nesse momento é interessante buscar as falhas do algoritmo apresentado, fazendo perguntas aos estudantes, com o intuito de obter possíveis propostas de melhorias: (1) Inicie a primeira atividade e digite o número 100.000. O que aconteceu? (2) O algoritmo mostrou todos os seus divisores? (3) O que aconteceria se utilizássemos números maiores ou iguais a 100.000? Porque isso acontece? (4) Como resolver esse problema? (5) Utilizando o mesmo esqueleto de atividade, seria possível criar um algoritmo que listasse os números pares compreendidos entre 2 e n , sendo n um número natural escolhido pelo usuário?

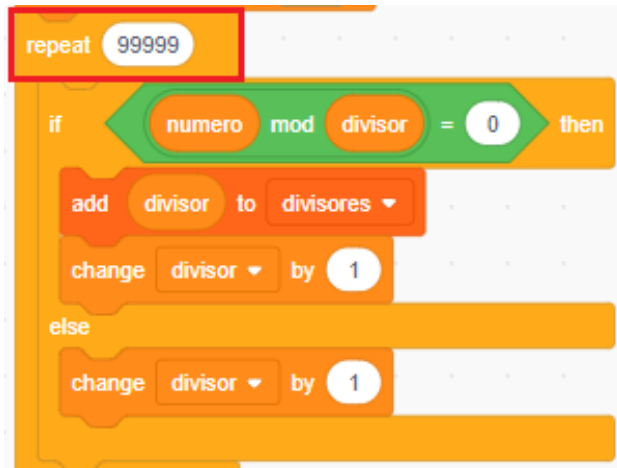
4.1.4. SUGESTÕES DE RESPOSTA

(1) Ao digitar o número 100.000 no visor, obtemos 35 divisores.

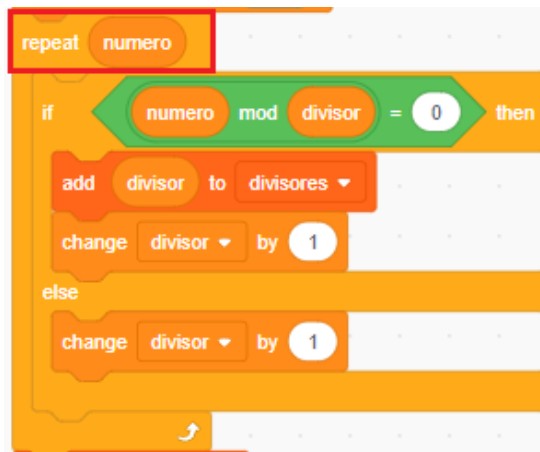


(2) O algoritmo não mostrou todos os divisores, pois o número 100.000 tem 36 divisores.

- (3) Caso fossem utilizados números maiores ou iguais a 100.000, também obteríamos um resultado não verdadeiro. Isso ocorre porque limitamos a testagem do algoritmo apenas para os primeiros 99.999 naturais. Logo, para números maiores que 99.999, não teremos o resultado correto.

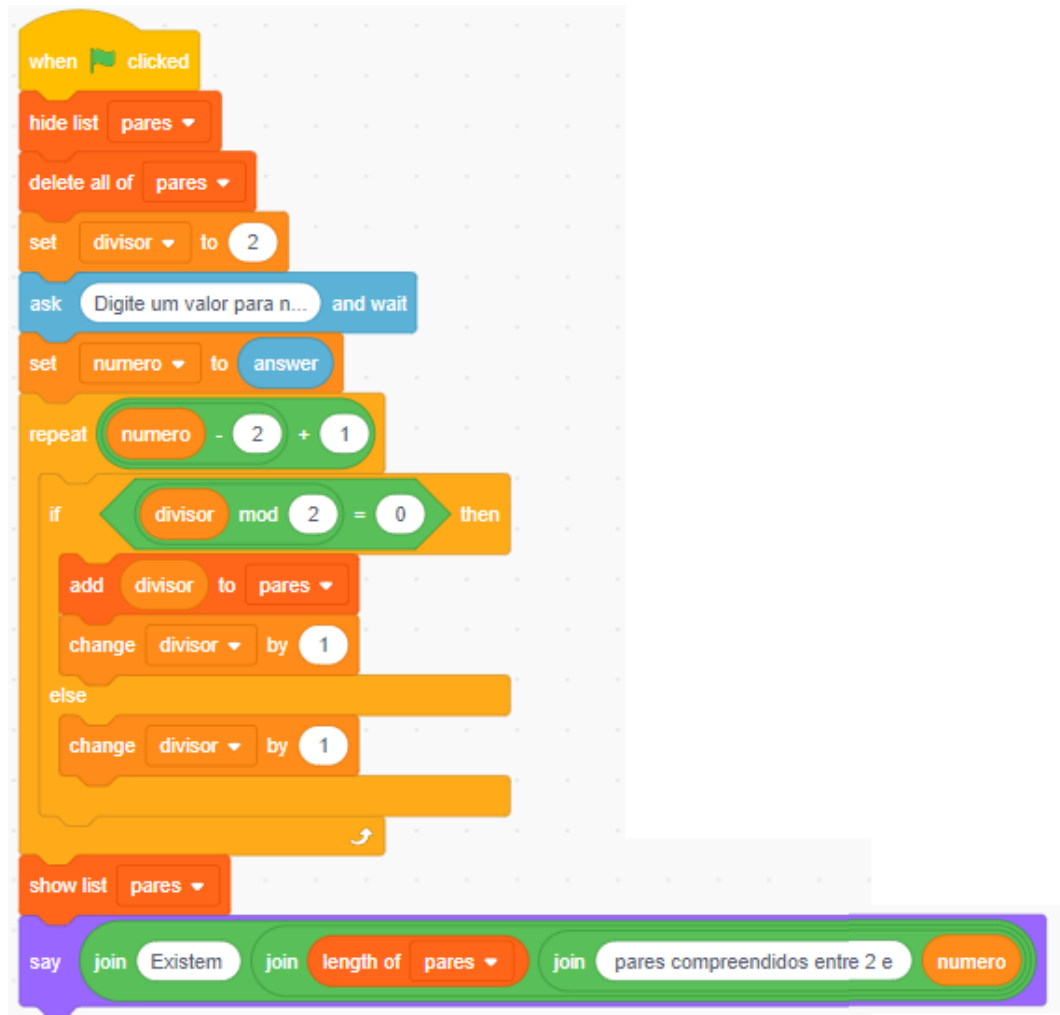


- (4) Para resolver esse problema, basta substituir o número “99.999” pela variável “numero”, assim, a testagem será feita até o número digitado pelo usuário. Dessa forma é possível corrigir o erro de compilação⁴.



- (5) Sim. Observe o algoritmo a seguir e veja as semelhanças com o anterior.

⁴ Ação de conversão de uma linguagem lógica para um código que possa ser lido por um computador.



Esse algoritmo está disponível em: <<https://scratch.mit.edu/projects/645704773/>>.

Acesso em: 16 fev. 2022.

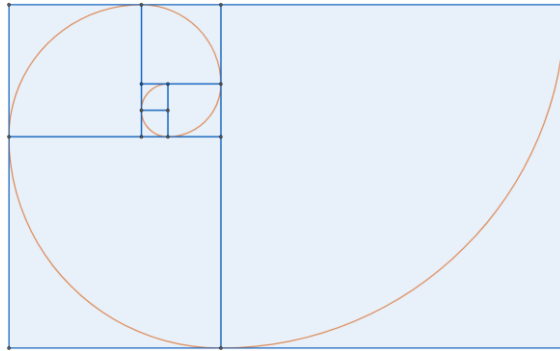
4.2. ATIVIDADE DA SEQUÊNCIA DE FIBONACCI

A habilidade de reconhecer padrões e classificar sequências em recursivas⁵ e não recursivas se faz presente na BNCC, que aparece não apenas na Matemática, mas também nas Artes e na Literatura.

Nesses campos, a sequência de Fibonacci se faz presente na forma de uma espiral que pode ser representada da seguinte forma (imagem criada no GeoGebra):

⁵ Propriedade sintática pela qual um elemento sofre uma mesma ação até que deixe de atender a um critério específico.

Figura 08 – Espiral de Fibonacci.



Fonte: O autor.

O menor dos quadrados tem lado 1; o segundo também tem lado 1; o terceiro tem lado 2; o quarto tem lado 3; o quinto tem lado 5; o sexto tem lado 8; o sétimo tem lado 13 e assim por diante. Calculando a razão entre o maior e o menor lado, em sequência, teremos:

- $\frac{1}{1} = 1$
- $\frac{2}{1} = 2$
- $\frac{3}{2} = 1,5$
- $\frac{5}{3} = 1,666\dots$
- $\frac{8}{5} = 1,6$
- $\frac{13}{8} = 1,625$

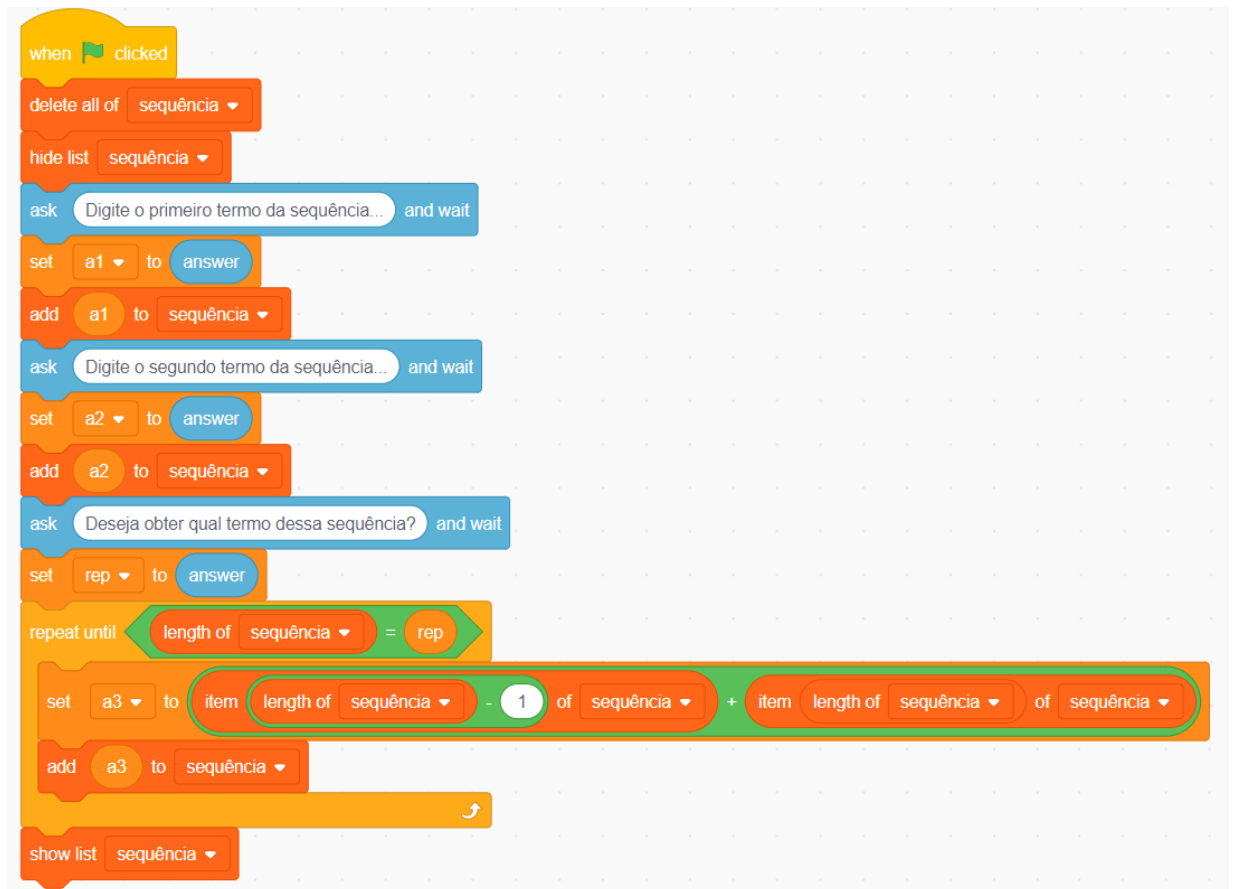
Ao continuar os cálculos, é perceptível que sempre atingiremos um número aproximado de 1,618, também conhecido como proporção ou razão áurea.

Com isso, é possível dizer que o 87º termo dessa sequência é 679.891.637.638.612.200 (seiscentos e setenta e nove quatrilhões, oitocentos e noventa e um trilhões, seiscentos e trinta e sete bilhões, seiscentos e trinta e oito milhões, seiscentos e doze mil e duzentos). Sim! É isso mesmo que você está pensando. Este resultado não foi calculado mentalmente, mas ele foi obtido através da recursividade e da lógica matemática que, alinhada com a computação, permitiu a realização dos cálculos em menos de um segundo para poder imprimir esta resposta.

4.2.1. CONSTRUÇÃO DA ATIVIDADE

O algoritmo computacional foi construído utilizando os blocos de comando relacionados a perguntas, variáveis, recursividade, repetição condicional e lista.

Figura 09 – Projeto Scratch que lista os primeiros n termos da sequência de Fibonacci, sendo n um número natural.



Fonte: O autor.

O algoritmo é iniciado quando a bandeira verde é clicada. O bloco “delete all of sequencia” deleta todos os elementos da sequencia, caso existam. O bloco “hide list sequencia” esconde a lista que apresenta a sequencia, visto que não é importante mostrá-la no início. O comando “ask Digite o primeiro termo da sequencia...” pede uma entrada do utilizador, que é armazenada na variável “a1” através do comando “set a1 to answer” e o bloco “add a1 to sequencia” irá adicionar a entrada armazenada na variável a1 na lista ordenada a qual chamamos de sequencia. O comando “ask Digite o segundo termo da sequencia...” pede uma segunda entrada do utilizador, que é armazenada na variável “a2” através do comando “set a2 to answer”

e o bloco “add a2 to sequencia” irá adicionar a entrada armazenada na variável a2 na lista. O bloco “ask Deseja obter qual termo dessa sequência?” pede uma entrada do utilizador, que é armazenada na variável “rep” através do comando “set rep to answer”. O comando “repeat until length of sequencia = rep” garante que a programação que virá a seguir irá se repetir até que o tamanho da lista “sequencia” seja igual à posição do termo que queremos. A junção de blocos “set a3 to item (length of sequencia) – 1 of sequencia + item length of sequencia of sequencia” faz com que o novo termo (a3) seja sempre a soma do penúltimo termo da lista com o último termo da lista “sequencia”, daí a recursividade quando seguido do comando “add a3 to sequencia”. O último comando “show list sequencia” fará a lista ordenada aparecer com o resultado da sequência a ser obtida. A atividade encontra-se *on-line* para eventuais consultas. Disponível em: <<https://scratch.mit.edu/projects/633306167/>>. Acesso em: 24 jan. 2022.

4.2.2. AOS PROFESSORES

Para um bom proveito desta atividade, é importante explorar ao máximo o conceito lógico de repetição condicional. Essa não é a única maneira de construir o fluxo de programação, então também é interessante investigar outras formas de representação do pensamento através da tentativa e erro. Não obstante, é também oportuno aproveitar o momento para mostrar as inúmeras aplicações da sequência de Fibonacci e suas peculiaridades.

A neurociência trouxe à luz a importância dos erros em meio ao aprendizado. Pesquisas realizadas na Universidade de Michigan mostram que, ao errar, o cérebro dispara sinapses e reage mais intensamente do que ao acertar um cálculo. Segundo Moser, diante do erro, o cérebro pode responder de duas formas: a primeira foi apelidada de Negatividade Relativa ao Erro (ERN – *Error-Related Negativity*), a qual denomina o aumento da atividade elétrica que se imagina ocorrer quando o cérebro está em conflito entre a resposta certa e a equivocada. Também é observado que ela ocorre independentemente de a pessoa saber se errou ou não. A segunda forma é chamada de Positividade de Erro (PE – *error positivity*), ou seja, ocorre quando há consciência de que um erro foi cometido e se presta atenção nele.

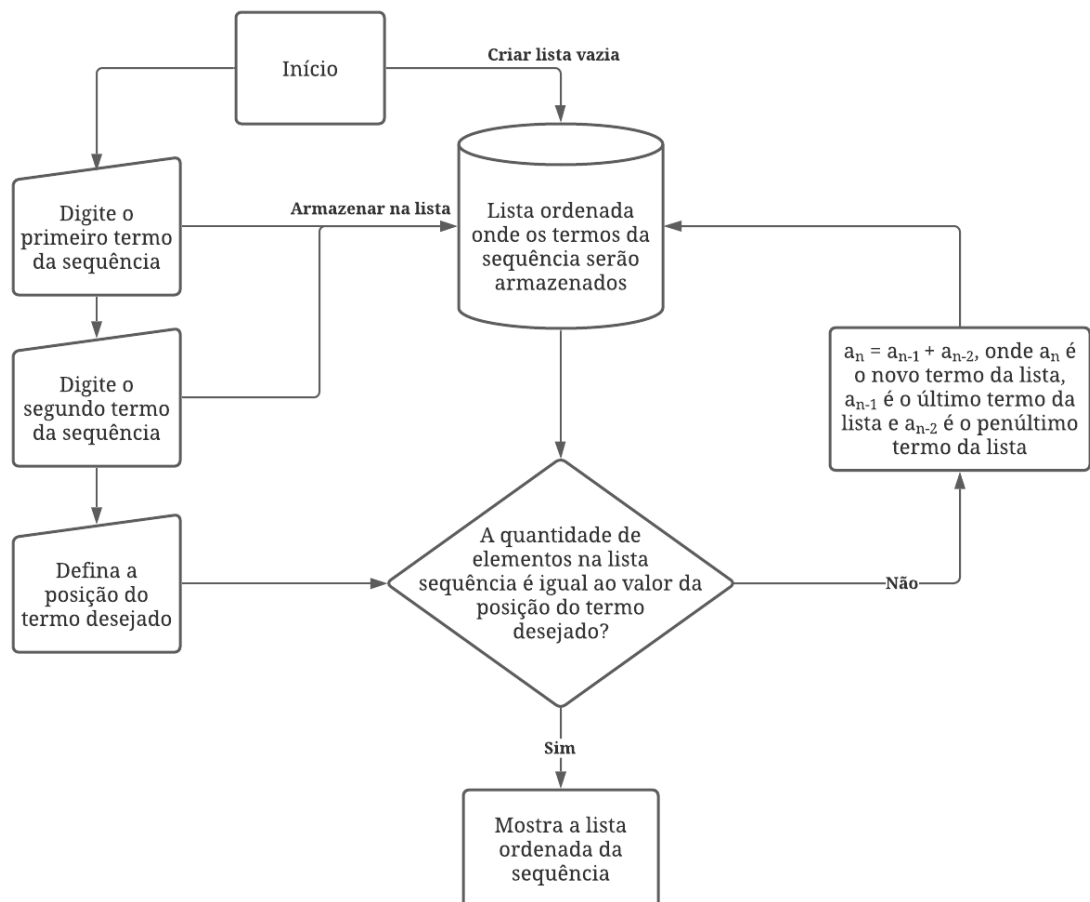
Quando os professores me perguntam como isso pode ser possível, respondo que o melhor entendimento que podemos ter sobre o assunto é que o cérebro cresce diante do erro, mesmo quando não estamos cientes dele, pois esse é um momento de conflito. E os melhores momentos para o cérebro crescer são aqueles em que somos desafiados. (MOSER et al., 2011)

Os disparos das sinapses são ainda mais intensos em pessoas que possuem ou adquirem a mentalidade de crescimento, isto é, pessoas que se deparam com os erros e continuam tentando até que um problema, uma vez complexo, se torne tangível de solução. Segundo Boaler, é preciso conscientizar os estudantes quanto à importância dos erros para o aumento da capacidade cerebral e, ao saberem disso, tornam-se mais dispostos e persistentes.

É de suma importância entender o poder do erro, pois crianças e adultos em qualquer lugar do mundo se sentem péssimos quando não acertam a resposta em desafios matemáticos. Aham que isso significa que elas não têm jeito para a disciplina, pois foram criadas em uma cultura de desempenho, na qual errar é desvalorizado e acarreta punições. (BOALER, 2014)

O fluxograma a seguir foi criado no Lucidchart para ilustrar uma outra forma de visualizar o pensamento utilizado para a confecção de um algoritmo funcional.

Figura 10 – Fluxograma que lista os primeiros n termos da sequência de Fibonacci.



Fonte: O autor.

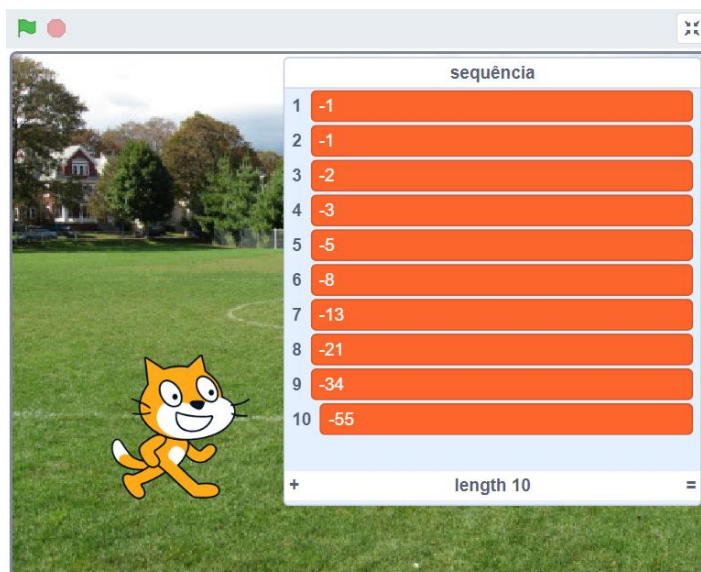
É possível interpretá-lo como mais uma forma de registrar o pensamento lógico para formar a sequência finita de elementos, em que o próximo elemento sempre será a soma dos dois anteriores. Torna-se significativo esquadrihar outras formas de representação do pensamento, com ou sem o uso das Tecnologias Digitais.

4.2.3. SUGESTÕES DE ABORDAGEM

Durante a atividade anterior, é possível levantar algumas questões reflexivas como: (1) O que acontece com a sequência se os números digitados forem negativos? (2) E se um for positivo e o outro negativo? E vice-versa. (3) O que aconteceria se, no momento da decisão, invertêssemos o “Sim” com o “Não”? (4) Por que é necessário pedir e armazenar os dois primeiros termos da lista? (5) Por qual motivo o critério de decisão utilizado para mostrar o resultado final da lista é a igualdade entre a quantidade de elementos da lista e o valor da posição do termo desejado? (6) Supondo que algoritmo esteja calculando o 23º termo, qual será o valor de a_n dados 2 e 7 como os primeiros termos dessa sequência? (7) Qual é a importância desse processo frente aos desafios tecnológicos atualmente?

4.2.4. SUGESTÕES DE RESPOSTA

(1) Obteremos uma sequência de números negativos, na qual o último termo será o menor de todos.



(2) Caso o segundo termo seja o oposto/simétrico do primeiro, sendo o primeiro negativo, teremos uma sequência crescente. Caso o segundo termo seja o oposto/simétrico do primeiro, sendo o primeiro positivo, teremos uma sequência decrescente. Se o primeiro termo for o oposto/simétrico do segundo termo menos um, sendo o segundo termo positivo, teremos sempre uma sequência crescente. Se o primeiro termo for o oposto/simétrico do segundo termo mais um, sendo o segundo termo negativo, teremos sempre uma sequência decrescente.

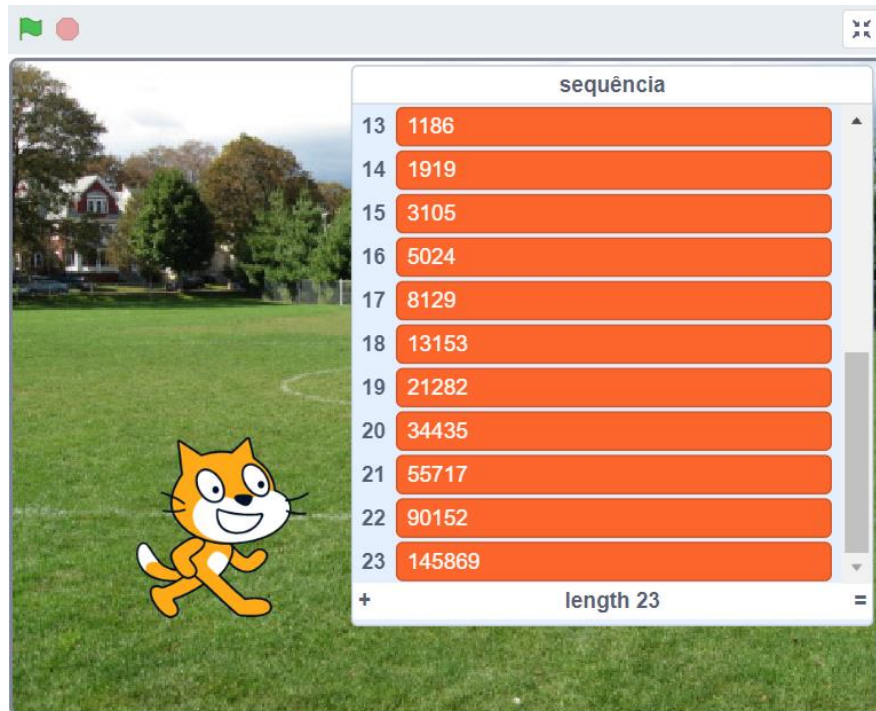
É possível pensar em outras possibilidades para o item (2) com o intuito de estimular o raciocínio lógico quanto aos processos algébricos envolvidos.

(3) Inicialmente, os dois primeiros termos digitados pelo usuário são enviados para dentro da lista, logo, se no momento da decisão invertêssemos o “Sim” com o “Não”, caso o n escolhido fosse maior que 2, o fluxograma traria apenas os dois termos digitados pelo usuário como resposta. No caso $n = 2$, um terceiro termo (a soma dos dois primeiros termos digitados pelo usuário) seria adicionado na lista e, logo após, apareceria uma lista com apenas 3 termos.

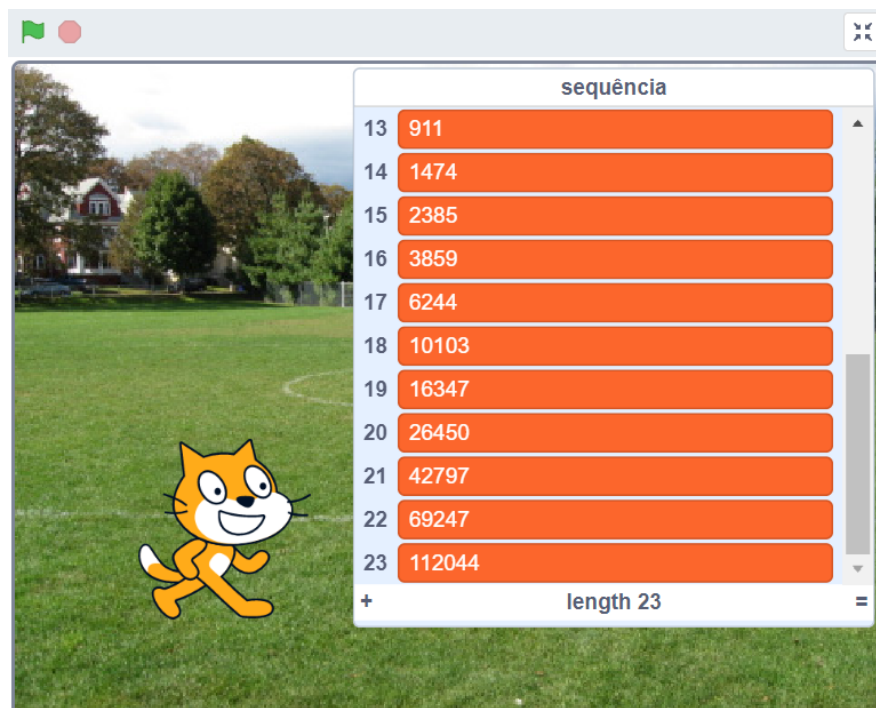
(4) Mostra-se necessário adicionar os 2 primeiros termos na lista para que a ordenação seja criada a partir deles. Além disso, a recursividade presente no algoritmo do Scratch ocorre até que a quantidade de termos da lista seja igual à posição do termo que desejamos. Dessa maneira, por meio da condição supracitada, a recursividade cessa e aciona o próximo bloco de comando, este exibe a lista dos termos até a posição pedida.

(5) Voltando ao item anterior (4) percebe-se que a recursividade foi criada a partir dessa necessidade e, por isso, revela uma condição que foi pré-estabelecida pelo criador do fluxo.

(6) Caso o número 2 seja o primeiro termo da sequência, 7 o segundo, e o termo pedido fosse o 23º, $a_{23} = 145.869$. Veja na próxima página.



Caso o número 7 seja o primeiro termo da sequência, 2 o segundo, e o termo pedido fosse o 23º, $a_{23} = 112.044$.



- (7) Por trás desse processo nos deparamos com os quatro pilares do Pensamento Computacional: Decomposição, Abstração, Reconhecimento de padrões e Criação/Validação de algoritmos. A partir de uma necessidade é possível decompor um problema complexo em partes menores, classificar os dados relevantes para resolver cada parte do problema por meio da abstração, reconhecer os padrões

envolvidos no processo de resolução de problemas que tenham características em comum e, por fim, transformar o pensamento em uma linguagem corrente, sistematizá-la após ter analisado um padrão intrínseco e registrá-la através de um algoritmo – no caso, a linguagem de programação Scratch. A importância desse feito se dá quando registramos um raciocínio lógico em linguagem corrente no computador através de uma linguagem computacional. Os computadores possuem um rápido processamento em ascensão e, combinando-os com a nossa capacidade de raciocínio, conseguem, a passos largos, nos mostrar um pouco mais do que para nós é considerado infinito.

4.3. ATIVIDADE DOS NÚMEROS PRIMOS

Muitos problemas matemáticos ainda continuam sem uma solução. Dentre eles, podemos citar a ausência de regularidade na sequência dos números primos. Euclides demonstrou por volta de 300 a.C. que essa sequência é infinita. Ainda assim, é possível determinar se um número inteiro é primo através da divisão por tentativas. Ou seja, caso seja preciso descobrir se um número é primo, basta verificar se ele é divisível por todos os primos inferiores a ele.

No que preconiza a BNCC a respeito do reconhecimento de padrões, a atividade nasceu dos seguintes questionamentos: “Como descobrir se um número é primo?”, “Em uma sequência de primos, com $a_1 = 2$, qual seria o milésimo termo?”, “É possível criar um algoritmo que indique a sequência dos primos?”. A resposta está no Scratch ou em qualquer outra linguagem computacional que seja capaz de criar um algoritmo tão poderoso a ponto de calcular os n primeiros números primos positivos em uma velocidade computacional, sendo n um número natural digitado pelo usuário. Portanto, ainda que não haja uma regularidade, é possível listar os termos dessa sequência utilizando *softwares*. Antes de partir para a linguagem de programação, é interessante relembrar a demonstração a seguir:

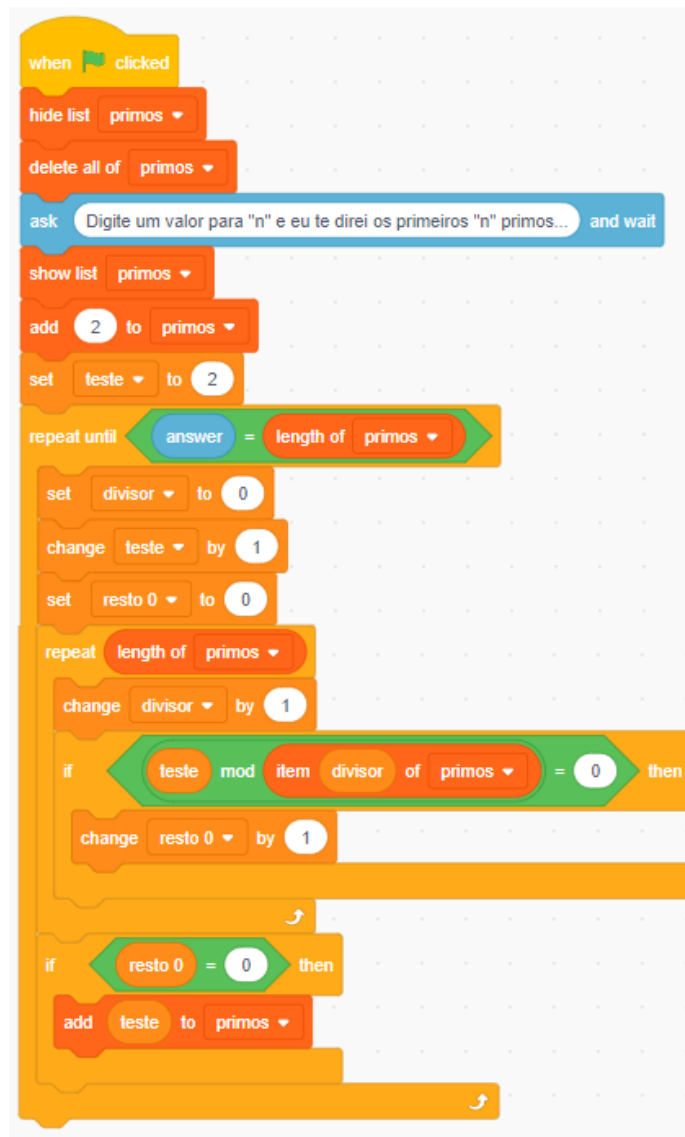
Se p um número primo, \sqrt{p} é irracional. Por absurdo, vamos supor que \sqrt{p} seja racional e, com isso, $\sqrt{p} = \frac{a}{b}$ e m.d.c. $(a, b) = 1$. Daí, $p = \frac{a^2}{b^2} \Rightarrow b^2 p = a^2$. Assim, a^2 seria um múltiplo de p , e a também seria um múltiplo de p , então $a = kp$ e

$b^2 p = (kp)^2 \Rightarrow b^2 p = k^2 p^2 \Rightarrow b^2 = k^2 p$. Logo, b seria múltiplo de p , o que seria uma contradição.

4.3.1. CONSTRUÇÃO DA ATIVIDADE

O algoritmo a seguir foi criado utilizando blocos de comando do tipo: variáveis, listas, função mod, desvio condicional, repetições e recursividade.

Figura 11 – Projeto Scratch que lista os primeiros n primos, sendo n um número natural.



Fonte: O autor.

O algoritmo se inicia quando a bandeira verde é clicada, a seguir o bloco “hide list primos” esconde a lista chamada “primos” e o bloco “delete all of primos” deleta qualquer conteúdo da lista “primos”. A lista dos primos servirá como um banco de dados para o armazenamento sequencial dos números primos encontrados ao longo do processo. O comando “ask Digite um valor para n e eu te direi os primeiros n primos...” irá armazenar o número natural digitado pelo usuário no bloco “answer”. Dado o número n , o bloco “show list primos” irá mostrar a lista dos “primos” e, em seguida, o comando “add 2 to primos” irá adicionar o número 2 a ela, este ocupará a primeira posição da lista, ou seja, o primeiro termo. A variável “teste” refere-se ao candidato à lista “primos”. O bloco de comando “set teste to 2” faz com que o algoritmo comece o teste de divisibilidade com o número 2, visto que ele já foi adicionado à lista “primos”. O comando “repeat until answer = length of primos” faz com que o conteúdo a seguir se repita até que a lista dos primos contenha exatamente a quantidade n digitada pelo usuário. O comando “set divisor to 0” faz com que a variável “divisor” assuma o valor zero. O comando “change teste by 1” faz com que o próximo número seja testado, no caso o número 3 será o primeiro a ser testado, visto que a variável “teste” foi definida para 2 anteriormente. O comando “set resto 0 to 0” define o valor da variável “resto 0” para 0, indicando que, inicialmente, não existem quaisquer divisões cujo resto foi igual a zero. O bloco de comando “repeat length of primos” repete o que vem a seguir de forma que a quantidade de vezes dessa repetição está atrelada à quantidade de primos na lista “primos”. O comando “change divisor by 1” usa o primeiro termo da lista “primos”, assim, os blocos “if teste mod item divisor of primos = 0 then change resto 0 by 1” testam a divisão do número testado pelo primeiro primo da lista, no caso o número 2. O bloco de comando “if resto 0 = 0 then add teste to primos” adiciona o número testado na lista “primos” caso os restos das divisões por todos os primos anteriores a ele não sejam iguais a zero. A atividade encontra-se *on-line* para eventuais consultas. Disponível em: <<https://scratch.mit.edu/projects/602406892/>>. Acesso em: 17 nov. 2021.

4.3.2. AOS PROFESSORES

Acredita-se que a atividade traz certa complexidade quanto aos desvios lógicos condicionais e ao apresentar uma repetição dentro de outra, bem como o processo de listar os primos e é exatamente por conta desse desafio que o processo se torna interessante. Nesse momento, os estudantes se deparam com a necessidade de resolver o problema proposto. Ao

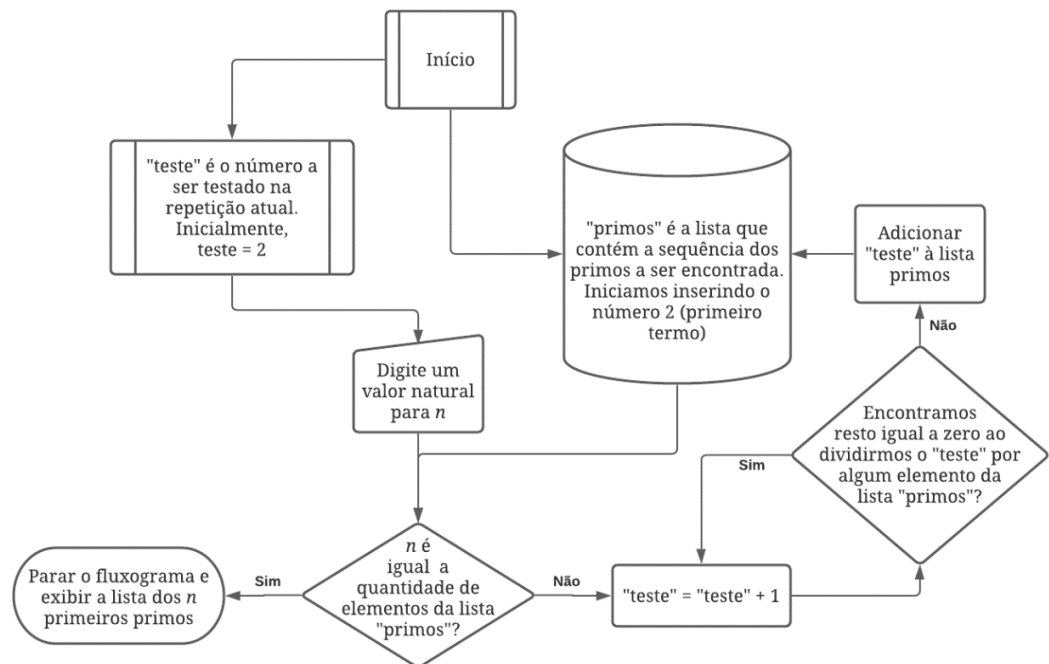
tentarem resolver, irão errar, tentar novamente, errar em um platô mais alto e tentar novamente, até que, finalmente, consigam resolver os questionamentos propostos através do algoritmo computacional. Segundo Blackwell (2017), ao estimular o desenvolvimento da mentalidade de crescimento, provemos de uma melhor intencionalidade pedagógica, visto que os estudantes procuram se arriscar mais e experimentar novos caminhos de solução ao invés de apenas aceitarem o nível de aprendizado adquirido enxergando o cérebro como um músculo em constante desenvolvimento.

Estudantes com uma mentalidade de crescimento persistem mais em problemas, desafio prazer e aprender com os erros, enquanto aqueles com uma mentalidade fixa desistir facilmente, evitar problemas desafiadores e ódio para cometer erros. Quando os alunos têm uma mentalidade de crescimento, eles atingem níveis mais elevados e quando os alunos recebem uma intervenção mental, as suas trajetórias de aprendizagem aceleram imediatamente em direção a realizações cada vez maiores. (BLACKWELL et al., 2007)

Complementa-se que o algoritmo existente nesse projeto não é único, isto é, não é interessante, de maneira alguma, fazer com que os estudantes repitam o código anterior de forma mecânica, mas sim que criem os seus próprios algoritmos, ou pseudocódigos, com o uso harmônico dos pensamentos algébrico e computacional. O único motivo, que pode ser significativo, é mostrar, previamente, o algoritmo anterior com o intuito de fazer com que os estudantes vejam a sequência de comandos. Desse modo poderão entender a funcionalidade de cada um dos blocos de comando e utilizarão os conceitos absorvidos para criarem novas sequências finitas de passos que permitem a solução de um problema que seja impraticável em um ponto de vista para o qual os humanos não tivessem o auxílio das tecnologias digitais.

Utilizando o fluxograma a seguir é possível ilustrar o que ocorre ao iniciarmos o algoritmo:

Figura 12 – Fluxograma que exibe os primeiros n primos, sendo n um número natural.



Fonte: O autor.

O fluxograma anterior ilustra o percurso lógico do algoritmo criado no Scratch de maneira desplugada⁶, isto é, sem o uso das tecnologias digitais. Vale ressaltar que esse modelo não é único e que é possível representar esse processo utilizando diferentes pensamentos lógicos. Estimular a confecção de fluxogramas que possam representar o mesmo processo pode ser um grande aliado na busca pelo pensamento computacional.

4.3.3. SUGESTÕES DE ABORDAGEM

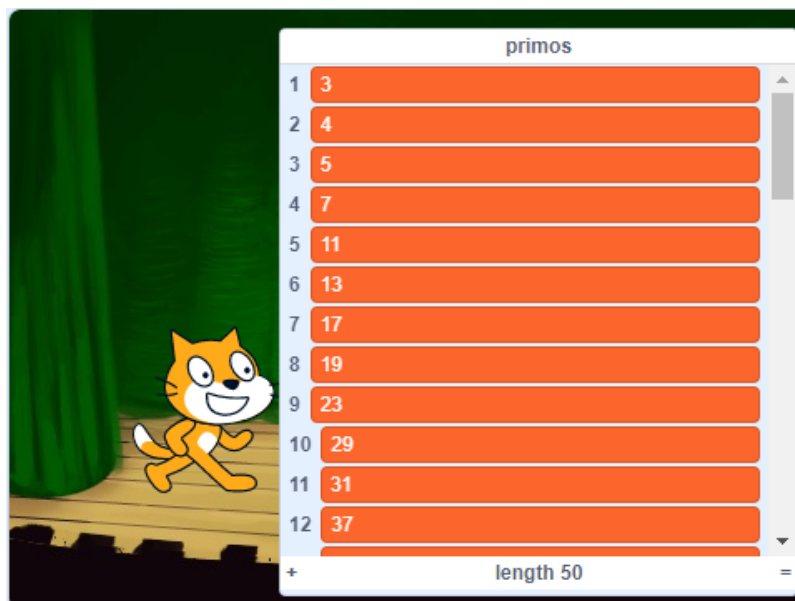
Através da atividade anterior torna-se possível colocar em pauta algumas questões do tipo: (1) Dado o início do processo, quando o algoritmo “percebe” que precisa parar o processo? (2) Por que existe um momento no fluxo em que o algoritmo começa a testar se o “teste” deixa resto zero quando dividido pelos primos da lista “primos”? (3) O que aconteceria se tivéssemos iniciado o processo sem deixar claro que o primeiro termo da lista “primos” é o número 2? (4) O que garante a correta ordenação da sequência de primos

⁶ Representação de um algoritmo sem o uso de dispositivos eletrônicos. É possível utilizar fluxogramas, pseudocódigos, ou, qualquer outro modo de ilustrar uma sequência de passos.

apresentada no final do processo? (5) Qual parte do fluxograma anterior poderia ser escrita de uma melhor forma?

4.3.4. SUGESTÕES DE RESPOSTA

- (1) O algoritmo percebe que precisa parar o processo quando a quantidade de termos dentro da lista “primos” for igual ao valor de n digitado inicialmente pelo usuário.
- (2) Esse momento de testagem é necessário porque, caso o “teste” deixe resto zero quando dividido por algum primo da lista “primos”, significa que é um múltiplo, ou seja, o “teste” não será primo e não entrará na lista “primos”.
- (3) O algoritmo foi construído a partir da inserção do número primo 2; logo, remover o bloco de ação “add 2 to primos” faz com que o fluxo não funcione da maneira correta e, por esse motivo, o número 2 não será o primeiro termo na sequência de primos e o número 4 irá aparecer na lista dos primos, o que é uma contradição.



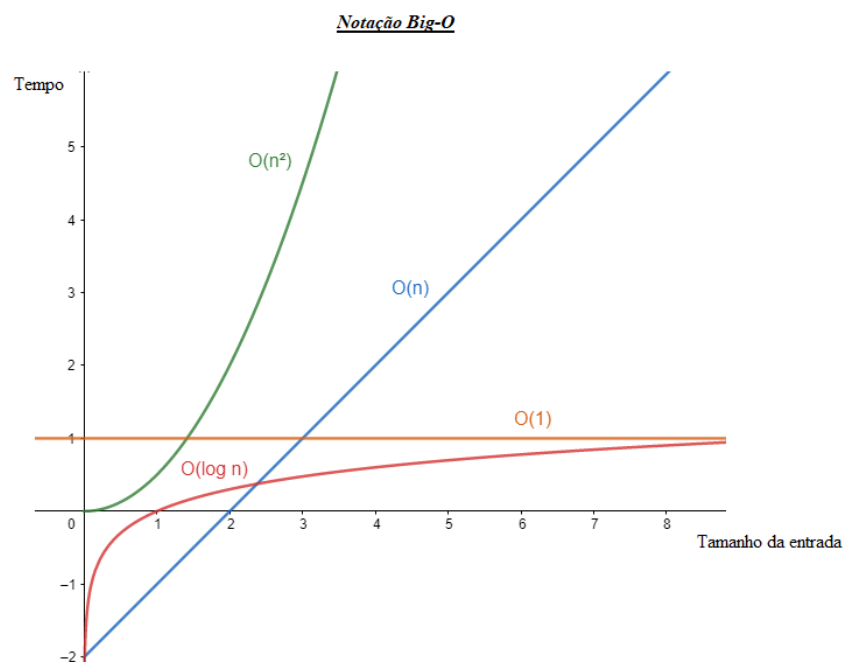
- (4) A correta ordenação é dada pela forma como o fluxograma foi construído. Após a testagem, caso o número seja primo, ele será adicionado à lista “primos”, e o fato da testagem ser sequencial garante a correta ordenação dos termos.
- (5) Seria possível escrever de uma melhor forma: Adicionar “teste” à lista primos depois do número primo de maior valor absoluto nela contido.

5. COMPLEXIDADE DE UM ALGORITMO

Um algoritmo somente passará a ser chamado de “programa” quando for convertido de uma linguagem corrente para uma linguagem aceita computacionalmente. Os programas de um computador exigem dois tipos de recurso: processamento (capacidade de realizar cálculos) e armazenamento, isto é, a memória (capacidade de recuperar informações disponíveis). Nesse contexto, é possível utilizar o termo “complexidade” para referir-se à essa quantidade de recursos que é demandado pelo programa. O pensamento intuitivo inicial seria calcular o tempo que determinado programa demoraria, do começo da execução até o final dela, onde ele imprime o resultado final. Esse pensamento não é levado em consideração pois dependeria da velocidade de processamento e armazenamento de cada máquina. Para avaliar corretamente a complexidade de um algoritmo é preciso levar em consideração a quantidade de passos ou interações que o código leva para executar do início ao fim.

A notação mais utilizada para conseguir descrever a complexidade de tempo de determinado código é chamada de *Big-O*. Essa notação mostra o comportamento do algoritmo em termos de crescimento de tempo de execução baseado em seus valores de entrada.

Figura 13 – Gráfico da complexidade de tempo da notação *Big-O*.



Fonte: O autor.

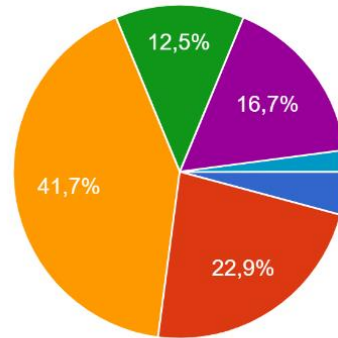
Por exemplo, queremos executar um algoritmo em menos de um segundo e sabemos que em um segundo são executados 10^6 passos e temos que a entrada feita pelo usuário é 10^5 . Caso a complexidade desse algoritmo seja $O(n^2)$ teremos $(10^5)^2 = 10^{10}$, ou seja, não seria possível rodar o programa em menos de um segundo pois $10^{10} > 10^6$. Se a complexidade, nesse caso, fosse $O(n)$ seria possível rodar o programa em menos de um segundo pois $10^5 < 10^6$.

No exemplo 3.2.1 (PAR OU ÍMPAR?), a complexidade de cada passo é $O(1)$, visto que a variável de entrada faz com que cada passo sejam executados uma única vez. Nesse caso, a resposta é dada de maneira instantânea, em um milésimo de segundo. Já no exemplo 3.2.2. (LANÇAMENTO DE MOEDA), a complexidade de tempo será $O(n)$ no passo em que há o comando de repetição (repeat answer), ou seja, o algoritmo só irá parar quando o comando for acionado n vezes.

6. RESULTADO DA PESQUISA

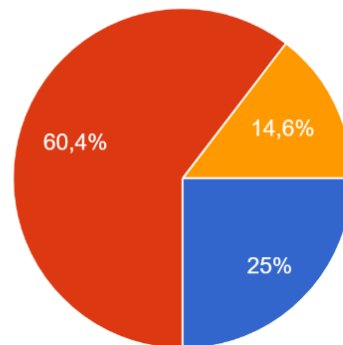
Você se encontra em qual faixa etária?

48 respostas



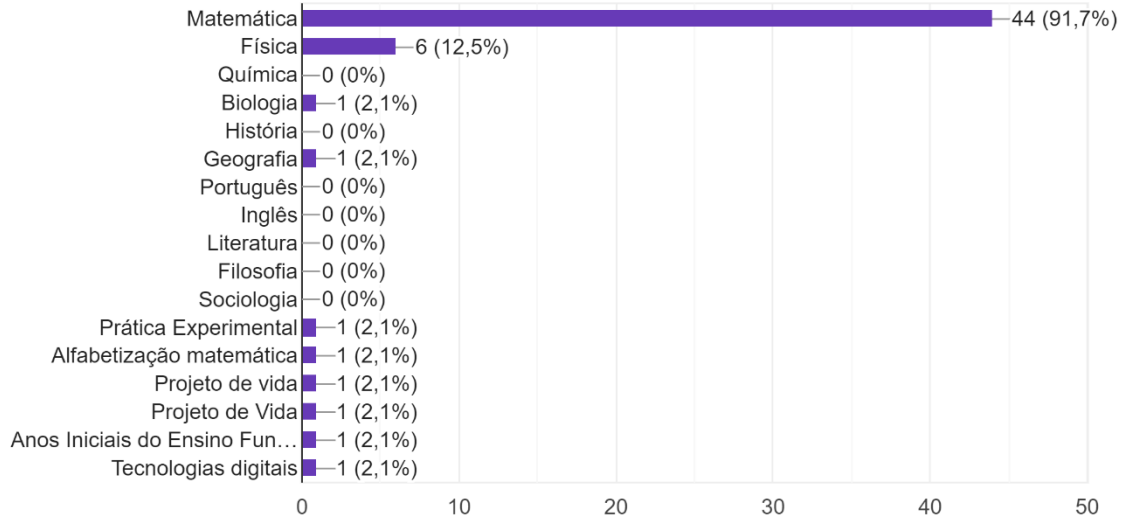
Você trabalha em uma rede/escola particular ou pública?

48 respostas



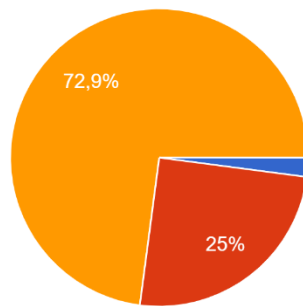
Qual(is) disciplina(s) você leciona?

48 respostas



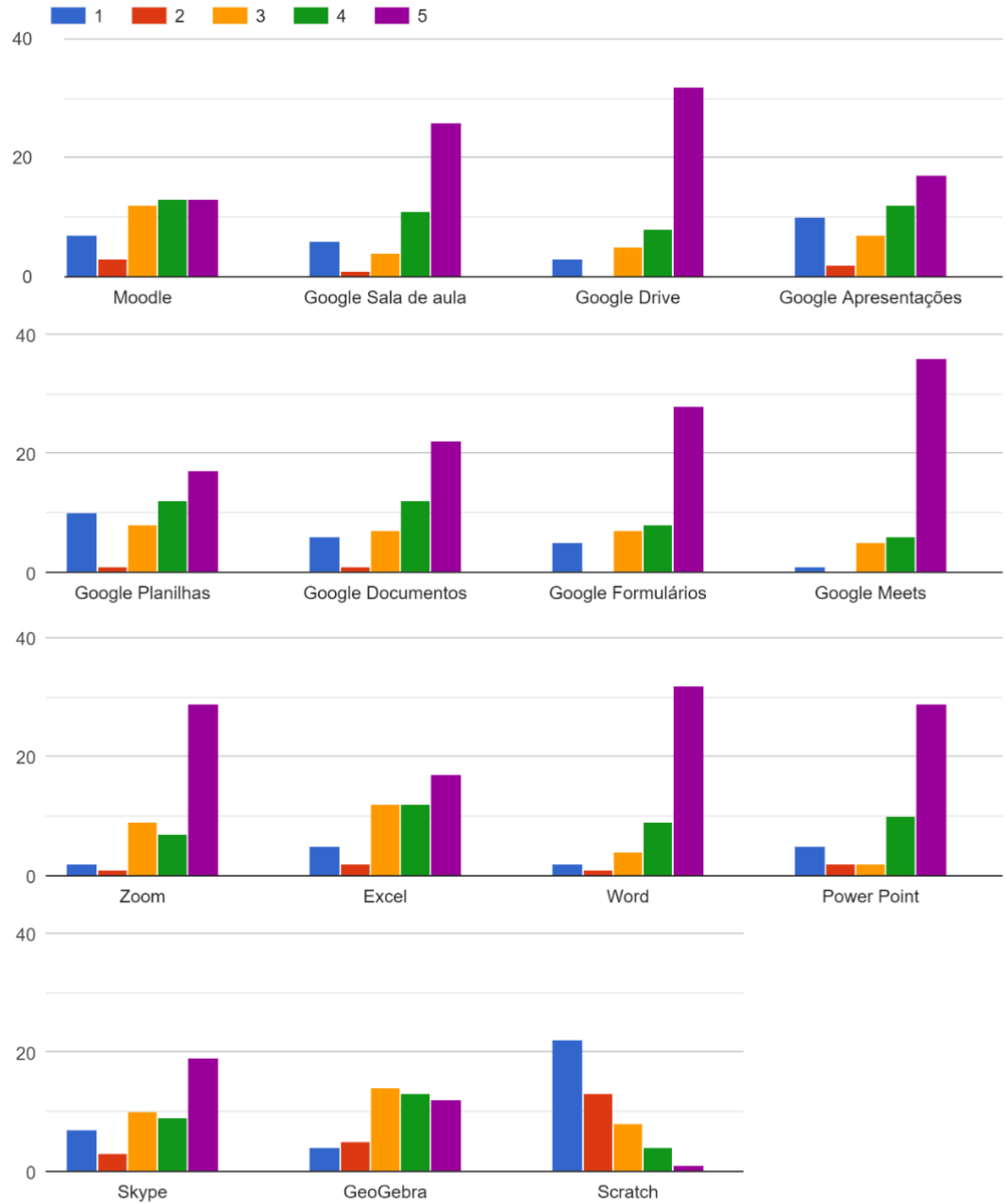
Caso existam computadores na sua escola, você consegue utilizá-los juntamente aos estudantes? (Exemplo: uma aula dentro de uma sala de informática).

48 respostas



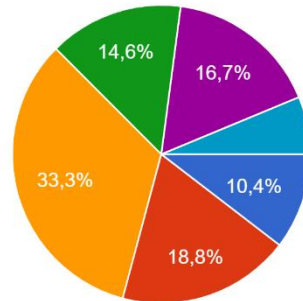
- Não existem computadores.
- Existem computadores mas não posso utilizá-los com os estudantes.
- Sim. Consigo utilizar os computadores com os alunos.

Em uma escala de 1 a 5, qual é a sua experiência em cada uma das ferramentas a seguir? (Sabendo que a nota 5 demonstra o domínio total de uma ferramenta).



Há quantos anos você trabalha na Educação Básica?

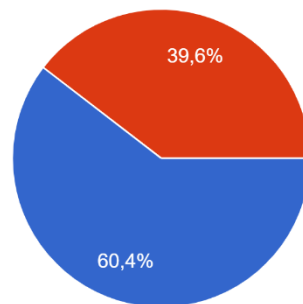
48 respostas



- Até 5 anos.
- Entre 6 e 10 anos
- Entre 11 e 15 anos
- Entre 16 e 20 anos.
- Entre 21 e 25 anos.
- Mais de 30 anos.

Você já ouviu falar no Pensamento Computacional na Educação?

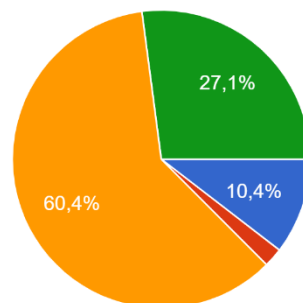
48 respostas



- Sim.
- Não.

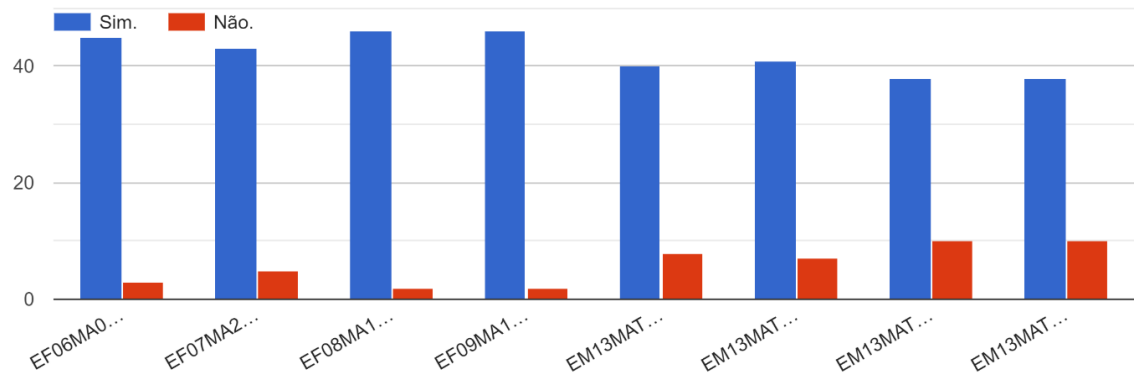
Você gosta da ideia da inclusão da Tecnologia na Educação?

48 respostas



- Nunca ouvi falar.
- Ouvi falar mas não gosto da ideia.
- Conheço e gosto da ideia.
- Conheço e já aplico essa ideia nas minhas aulas.

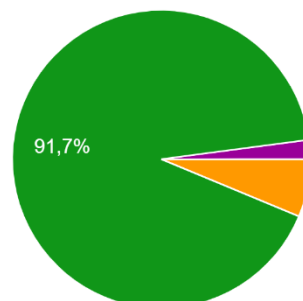
Na sua opinião, as habilidades a seguir da Base Nacional Comum Curricular estão, de alguma forma, referendando o Pensamento Computacional?



EM06MA04, EF07MA28, EF08MA11, EF09MA15, EM13MAT101, EM13MAT315, EM13MAT403 e EM13MAT510, respectivamente.

Marque a opção que mais se encaixa com o significado de fluxograma. (ou escreva com as suas palavras).

48 respostas



- Não conheço o significado de fluxograma.
- É um complicador. Tudo fica mais difícil de entender quando falamos sobre flu...
- É uma representação esquemática de como podemos dar aulas para as cria...
- É um tipo de diagrama que pode ser entendido como uma representação e...
- É uma representação gráfica de um processo que pode ou não ser associ...

Na sua opinião: Como o pensamento computacional pode ser trabalhado com os estudantes utilizando as tecnologias disponíveis? (digitais ou não).

Resposta 1: “O pensamento pode ser alocado em diversas situações do dia a dia para tomadas de decisão, por exemplo, para que os alunos entendam e associem de forma mais natural.”

Resposta 2: “Importante acompanhar esse desenvolvimento e possibilidade, a pandemia demonstrou isso”

Resposta 3: “No caso da Matemática, todo processo lógico pode ser pensado de forma computacional. Para colocar em prática é preciso trabalhar com a tradução da

linguagem, e o fluxograma é uma boa ferramenta. Tecnologias digitais podem acelerar/facilitar este processo, tanto para a construção da linguagem quanto para a sua verificação/validação.”

Resposta 4: “Pode ser trabalhado utilizando a lógica matemática, condicionais, esquemas e as ideias de gamificação de conteúdos.”

Resposta 5: “Facilitar o trabalho do professor.”

Resposta 6: “Não sei.”

Resposta 7: “Acredito que o pensamento computacional pode ser trabalho com lógica e algoritmos aplicáveis em conteúdos matemáticos, auxiliando os estudantes a desenvolverem e estruturarem o raciocínio matemático.”

Resposta 8: “diversas maneiras”

Resposta 9: “De inúmeras maneiras, durante todo o ensino básico, à BNCC dá pistas, mas não creio que o foco seja em fluxograma, e não acho que uma linguagem de programação deva ser utilizada. Acredito em atividades cada vez mais diversificadas como o programa de pensamento computacional desenvolvido pela UNICAMP que não usa necessariamente processos digitais.”

Resposta 10: “Não sei”

Resposta 11: “Trabalhando com programas que estimulem a construção de algoritmos para produzir algo, como por exemplo o Scratch, para simular movimentos, histórias. Ou algo de programação básica mesmo. Mas acredito que é necessário capacitar os professores de Matemática para tal. E fica a questão se isso seria mesmo incumbência dos professores de matemática.”

Resposta 12: “Para pensar a lógica de construção, em etapas, por exemplo.”

Resposta 13: “Utilizando estratégias que levam ao raciocínio investigativo, aplicando pensamento lógico matemático na resolução de problemas, incentivando a estruturação do raciocínio em forma de algoritmo de resolução”

Resposta 14: “Usando softwares que ajudem na visualização de conceitos difíceis e possibilitem expandir a mente”

Resposta 15: “Primeiramente oferecendo formação ao professor e depois, disponibilizando o suporte/recursos à escola.”

Resposta 16: “Quando a instituição dispõe de instrumentos tecnológicos, podemos utilizar as tics, como por exemplo, Phet, Scratch, GeoGebra etc. Mas caso não dispõe, pode-

se trabalhar a partir de esquemas na lousa ou por meio de material concreto confeccionamos pelo prof.”

Resposta 17: “Para construir conhecimentos”

Resposta 18: “Não sei responder”

Resposta 19: “No ensino de matemática isso é possível diversas vezes, sejam nas disciplinas que envolvam conceitos geométricos, sejam nas disciplinas que possui uso de algoritmos. O uso de aplicativos como GeoGebra, outros que iniciam a programação etc.”

Resposta 20: “Esquematização e realização de problemas cotidianos com sequência de dados”

Resposta 21: “Através de programas como o GeoGebra.”

Resposta 22: “Não trabalho o pensamento computacional.”

Resposta 23: “Na prática! Com a utilização de plataformas, programação exercitando o protagonismo.”

Resposta 24: “Pode contribuir para dentro de uma proposta de ensino investigativo da Matemática, em ambientes de aprendizagens que permitam ao aluno ou aluna a serem ativos nesse processo, com ou sem a utilização de recursos tecnológicos digitais. Sendo assim, acredito que a forma como é entendido o ensino de Matemática deva ser encarado de outra forma, considerando aspectos individuais que interferem no desenvolvimento de conceitos.”

Resposta 25: “Trabalhando com softwares que pode reproduzir leis de formação. Tanto geometricamente (construções geométricas) quanto em forma de gráficos (funções e leis de formação de equações).”

Resposta 26: “Uma forma é criar jogos com os alunos. Outra é resolver atividades utilizando programas como o GeoGebra, por exemplo.”

Resposta 27: “.”

Resposta 28: “Provocar o pensamento de como e quando usar, se julga necessário ou não, visando uma nova visualização que possa ser mais natural para os nativos digitais do que trabalhar apenas no papel.”

Resposta 29: “Na criação de algoritmos, processos com passos bem definidos, que resolvam problemas. Essa criação se dá em qualquer disciplina para qualquer assunto.”

Resposta 30: “Gosto de utilizar atividades introdutórias das ideias como as da computação desplugada ou atividades como as disponíveis no code.org... mas o acesso restrito a computadores dificultam a evolução”

Resposta 31: “Um exemplo: elaborar um passo a passo de dobradura de papel ou origami para que os colegas consigam um bom resultado”

Resposta 32: “Estimulando o raciocínio lógico dos estudantes. Oportunizando situações em q as tecnologias digitais sirvam para resolver situações relativas ao cotidiano escolar”

Resposta 33: “Preciso me apropriar com conceito antes de opinar.”

Resposta 34: “viabilizando a consolidação de objetos antes estáticos através de uma proposta dinâmica, iterativa e participativa do conhecimento em questão. O ensino, por ex., de funções contínuas pode ser amplamente abordado a partir de tecnologias digitais acessíveis, comparando com funções em domínios discretos.”

Resposta 35: “Pode ser desenvolvimento através de construções de algoritmos, que ainda não sei fazer, mas pretendo aprender.”

Resposta 36: “A partir da familiarização dos alunos (e antes disso, dos docentes) com ferramentas digitais básicas e acessíveis e de novas metodologias de ensino-aprendizagem. Entendo ser importante também apresentar ao aluno a razão pela qual o pensamento computador é importante e pode agregar em seu futuro, apresentando aplicações práticas desse conceito.”

Resposta 37: “em jogos matemáticos em sites especializados ou apps de celular”

Resposta 38: “Mostrando os processos algorítmicos. Quase tudo na Matemática é um algoritmo.”

Resposta 39: “Xxx”

Resposta 40: “De forma gradual e interdisciplinar”

Resposta 41: “Acredito que o pensamento computacional pode ser trabalhado até mesmo sem computadores. A construção do pensamento lógico independe de tecnologia, mas a conexão do pensamento computacional com o cotidiano do aluno pode mostrá-lo a importância da matemática de uma forma bem atual.”

Resposta 42: “Acredito que o pensamento computacional pode contribuir bastante na organização e desenvolvimento de estratégias para a resolução de problemas.”

Resposta 43: “O educando aprende fazendo, quanto mais oportunidades lhe forem dadas, maiores são as chances do desenvolvimento pensamento computacional.”

Resposta 44: “Sendo adaptado a realidade de cada turma.”

Resposta 45: “GeoGebra, Matlab, Excel, aplicativos e *sites* matemáticos com jogos para sala de aula”.

7. CONSIDERAÇÕES FINAIS

A tecnologia faz parte da sociedade e, inevitavelmente, está em constante evolução sua constante evolução é inevitável. A principal motivação para a realização desse projeto é a de salientar a importância da tecnologia na educação, principalmente quando mencionamos a aprendizagem significativa. A intencionalidade pedagógica no ato de ensinar tecnologia torna-se evidente pelo fato dela se fazer presente no cotidiano e prometer inovações a passos largos, principalmente no mundo do trabalho, no qual empresas e profissões tecnológicas têm se destacado no período pandêmico. Não obstante, as capacidades adquiridas no processo fornecem estímulos ao raciocínio lógico e à resolução de problemas de maneira estratégica, cooperativa e crítica, que fazem com que os estudantes estejam preparados para quaisquer desafios do futuro, inclusive para profissões que sequer existem.

O desenvolvimento desse projeto perpassa pela BNCC que, assim como as autoridades citadas durante o projeto, demonstra o quanto as competências e habilidades atuais estão ligadas à tecnologia e de como podem ser úteis na resolução de problemas por meio da decomposição, do reconhecimento de padrões, da abstração e da criação de algoritmos/fluxogramas.

O intuito das atividades e da pesquisa presentes neste trabalho é de ampliar os conhecimentos tecnológicos para professores de Matemática do Ensino Básico e esclarecer as principais habilidades e competências que dizem respeito ao currículo comum nacional, vigente desde 2018. Apresenta-se o artefato Scratch e promove-se a explanação de atividades iniciais para o entendimento do comportamento algorítmico seguido de atividades mais elaboradas, e de cunho formativo, compostas por três momentos cada: construção da atividade, aos professores e sugestões de resposta.

Para os estudantes, mostra-se importante ressaltar a importância de motivá-los a investigar, a partir de seus próprios interesses, tarefas matemáticas que possam ser “algoritmizadas” por meio de tentativa e erro, trabalhando, assim, os quatro pilares do pensamento computacional, transcodificando a linguagem corrente em linguagem computacional Scratch. As crianças e adolescentes que estão nascendo em um mundo digital estão acostumadas com diferentes dispositivos e formas de interação; sendo assim, a relatividade desse fato vai de encontro à necessidade da adaptação das escolas e das metodologias de ensino, quanto às tecnologias disponíveis atualmente e que farão parte do futuro.

Por fim, sugere-se a democratização dos computadores para as escolas, pautada em um ideal equalitário, visando ao acesso à educação de qualidade para todos, à formação de professores que possam integrar o computador em suas metodologias de ensino e à reestabelecer as formas de aprendizagem e avaliação, enfatizando o protagonismo dos estudantes para que eles exerçam um papel consciente na autoavaliação.

O tema em si não se esgota nessas linhas e aponta diversos estudos futuros que partem dos estudos da computação clássica ou da informática educativa, perpassando pela psicologia cognitiva, pela lógica, pelos temas que envolvem computação e sociedade, os fatos e artefatos como construções sociotécnicas e por todos os ramos do pensamento humano que utilizam o pensamento de máquinas como aqueles que podem contribuir significativamente para a aprendizagem humana, seja esta escolar ou do dia a dia.

8. REFERÊNCIAS

BLACKWELL, L.; TRZESNIEWSKI, K., & DWECK, C. (2007). **Implicit Theories of Intelligence Predict Achievement across an Adolescent Transition: A Longitudinal Study and an Intervention.** Child Development.

BOALER, J. **The Mathematics of Hope.** 2014.

BOALER, J. **Mathematical Mindsets: Unleashing Students' Potential Through Creative Math, Inspiring Messages and Innovative Teaching.** San Francisco, CA: Jossey-Bass, 2015.

BRASIL. Ministério da Educação. **Base Nacional Comum Curricular.** Brasília, 2018. Disponível em: <<http://basenacionalcomum.mec.gov.br/abase/>>. Acesso em: 23 maio 2022.

CSIKSZENTMIHALYI, M. **Flow: The Psychology of Optimal Experience.** 1990.

CYSNEIROS, P.G. “Resenha Crítica. Logo: Computadores e Educação”. **Revista Brasileira de Estudos Pedagógicos.** Brasília, (MEC/INEP), vol.72, n.170, p.106-109, jan./abr. 1991.

IBGE. **Pesquisa Nacional por Amostra de Domicílios Contínua.** Acesso à internet e à televisão e posse de telefone móvel celular para uso pessoal 2019. Rio de Janeiro, 2021. Disponível em: <https://biblioteca.ibge.gov.br/visualizacao/livros/liv101794_informativo.pdf>. Acesso em: 26 jan. 2022.

LESLIE, S-J.; CIMPIAN, A.; MEYER, M., & FREELAND, E. “Expectations of brilliance underlie gender distributions across academic disciplines”. **Science.** Washington, DC, 347, p. 262-265, 2015.

MORAES MELLO, C. **Educação 5.0: Educação Para o Futuro.** 2020.

MOSER, J. S.; SCHRODER, H. S.; HEETER, C.; MORAN, T. P., & LEE, Y. H. “Mind Your Errors Evidence for a Neural Mechanism Linking Growth Mind-Set to Adaptive Posterror Adjustments”. **Psychological Science,** 10.1177/0956797611419520. 2011.

PINK, D. **Motivação 3.0 – Drive: A surpreendente verdade sobre o que realmente nos motiva.**

RESNICK, M. **Jardim de infância para a vida toda.** 2017.

WING, J. “Computational thinking and thinking about computing”. **Computer Science Department.** Pittsburgh, 2008.

ANEXO A

Atividade	Par ou ímpar?
Objetivo	Criar um fluxograma/ algoritmo que, a partir da entrada de um número natural, verifica se o mesmo é par ou ímpar, com ou sem o uso de tecnologias digitais.
Unidade temática	Números
Competências da BNCC	<p>2 – Desenvolver o raciocínio lógico, o espírito de investigação e a capacidade de produzir argumentos convincentes, recorrendo aos conhecimentos matemáticos para compreender e atuar no mundo.</p> <p>6 – Enfrentar situações-problema em múltiplos contextos, incluindo-se situações imaginadas, não diretamente relacionadas com o aspecto prático-utilitário, expressar suas respostas e sintetizar conclusões, utilizando diferentes registros e linguagens (gráficos, tabelas, esquemas, além de texto escrito na língua materna e outras linguagens para descrever algoritmos, como fluxogramas, e dados).</p>
Habilidade da BNCC	(EF06MA04) Construir algoritmo em linguagem natural e representá-lo por fluxograma que indique a resolução de um problema simples (por exemplo, se um número natural qualquer é par).
Objetos do conhecimento	Conceito de variável, resto da divisão de um número natural por outro e desvio condicional (lógica if-then-else).
Sugestões de abordagem	Professor, nessa atividade o estudante precisará criar um fluxograma a partir de um papel, lápis e borracha ou tecnologia digital como, por exemplo, o Scratch, que represente um algoritmo capaz de verificar se um número natural é par ou ímpar. É importante citar que conhecer os aspectos cognitivos da sua classe é muito importante para não dar de menos, subestimando a capacidade das crianças e nem errando a mão sufocando-as com conteúdos ou, até mesmo, esperando mais delas do que elas podem dar, no momento.

ANEXO B

Atividade	Lançamento de moeda
Objetivo	Criar um fluxograma/ algoritmo que, a partir da entrada de um número natural n , realizará n lançamentos e, por fim, exibirá a quantidade total de lançamentos que resultaram em “cara” e em “coroa”.
Unidade temática	Probabilidade e estatística
Competências da BNCC	<p>2 – Desenvolver o raciocínio lógico, o espírito de investigação e a capacidade de produzir argumentos convincentes, recorrendo aos conhecimentos matemáticos para compreender e atuar no mundo.</p> <p>6 – Enfrentar situações-problema em múltiplos contextos, incluindo-se situações imaginadas, não diretamente relacionadas com o aspecto prático-utilitário, expressar suas respostas e sintetizar conclusões, utilizando diferentes registros e linguagens (gráficos, tabelas, esquemas, além de texto escrito na língua materna e outras linguagens para descrever algoritmos, como fluxogramas, e dados).</p>
Habilidade da BNCC	<p>(EF07MA34) Planejar e realizar experimentos aleatórios ou simulações que envolvam cálculo de probabilidades ou estimativas por meio de frequência de ocorrências.</p> <p>(EF07MA05) Resolver um mesmo problema utilizando diferentes algoritmos.</p>
Objetos do conhecimento	Conceito de variável, números aleatórios, probabilidade, estatística e desvios condicionais (lógica if-then-else).
Sugestões de abordagem	Nessa atividade, a partir da linguagem de programação Scratch, os estudantes deverão construir um algoritmo capaz de realizar lançamentos sucessivos que registra o resultado de cada lançamento, julgando se o mesmo resultou em “cara” ou “coroa”. Professor, é importante mencionar a velocidade computacional para tal experimento, visto que é possível realizar mais de 10.000 lançamentos em menos de 1 segundo. Ao final, é possível pedir para que os estudantes correlacionem os números obtidos em “cara” e “coroa” e verifiquem a probabilidade a partir da frequência de cada ocorrência.

ANEXO C

Atividade	Fatorial de um número natural
Objetivo	Criar um fluxograma/ algoritmo que seja capaz de calcular o fatorial de um número natural.
Unidade temática	Operações básicas (multiplicação)
Competências da BNCC	<p>2 – Desenvolver o raciocínio lógico, o espírito de investigação e a capacidade de produzir argumentos convincentes, recorrendo aos conhecimentos matemáticos para compreender e atuar no mundo.</p> <p>6 – Enfrentar situações-problema em múltiplos contextos, incluindo-se situações imaginadas, não diretamente relacionadas com o aspecto prático-utilitário, expressar suas respostas e sintetizar conclusões, utilizando diferentes registros e linguagens (gráficos, tabelas, esquemas, além de texto escrito na língua materna e outras linguagens para descrever algoritmos, como fluxogramas, e dados).</p>
Habilidade da BNCC	(EF06MA03) Resolver e elaborar problemas que envolvam cálculos (mentais ou escritos, exatos ou aproximados) com números naturais, por meio de estratégias variadas, com compreensão dos processos neles envolvidos com e sem uso de calculadora.
Objetos do conhecimento	Conceito de variável, multiplicação e desvios condicionais (lógica if-then-else).
Sugestões de abordagem	Professor, nessa atividade os estudantes precisam pensar em um fluxo algorítmico que seja capaz de calcular o fatorial de qualquer número natural. O método utilizado consiste na multiplicação do número n , dado pelo usuário, por $n - 1$, $n - 2$, $n - 3$, ... e 1, sucessivamente. É importante estar atento no desvio condicional responsável pela resposta individual para quando $n = 0$. Portanto, quando n for igual a zero, é preciso fazer com que o algoritmo imprima o número “1”.