

**CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
CAMPUS TIMÓTEO**

Rafael Lima da Silva

**UTILIZAÇÃO DO SCRATCH COMO FERRAMENTA DE ENSINO DE
CONCEITOS BÁSICOS DE PROGRAMAÇÃO PARA CRIANÇAS DE
12 A 14 ANOS: UMA PROPOSTA DE PROJETO DE EXTENSÃO
PARA O CEFET-MG**

Timóteo

2024

Rafael Lima da Silva

**UTILIZAÇÃO DO SCRATCH COMO FERRAMENTA DE ENSINO DE
CONCEITOS BÁSICOS DE PROGRAMAÇÃO PARA CRIANÇAS DE
12 A 14 ANOS: UMA PROPOSTA DE PROJETO DE EXTENSÃO
PARA O CEFET-MG**

Monografia apresentada à Coordenação de Engenharia de Computação do Campus Timóteo do Centro Federal de Educação Tecnológica de Minas Gerais para obtenção do grau de Bacharel em Engenharia de Computação.

Orientador: Viviane Cota Silva

Timóteo

2024

RAFAEL LIMA DA SILVA

**UTILIZAÇÃO DO SCRATCH COMO FERRAMENTA DE ENSINO DE
CONCEITOS BÁSICOS DE PROGRAMAÇÃO PARA CRIANÇAS DE
12 A 14 ANOS: UMA PROPOSTA DE PROJETO DE EXTENSÃO
PARA O CEFET-MG**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Computação do Centro Federal de Educação Tecnológica de Minas Gerais, campus Timóteo, como requisito parcial para obtenção do título de Engenheiro de Computação.

Trabalho aprovado. Timóteo, 14 de fevereiro de 2025:

Profa. Dra. Viviane Cota Silva
Orientadora

Prof. Dr. Lucas Pantuza Amorim
Professor Convidado

Prof. Me. Douglas Nunes de Oliveira
Professor Convidado

Timóteo, 2025



FOLHA DE APROVAÇÃO DE TCC N° 7/2025 - DECOMTM (11.63.11)

(N° do Protocolo: NÃO PROTOCOLADO)

(Assinado digitalmente em 21/02/2025 11:16)

DOUGLAS NUNES DE OLIVEIRA
PROFESSOR ENS BASICO TECN TECNOLOGICO
DECOMTM (11.63.11)
Matrícula: ###212#8

(Assinado digitalmente em 21/02/2025 11:45)

LUCAS PANTUZA AMORIM
PROFESSOR ENS BASICO TECN TECNOLOGICO
DECOMTM (11.63.11)
Matrícula: ###974#1

(Assinado digitalmente em 21/02/2025 11:16)

VIVIANE COTA SILVA
PROFESSOR ENS BASICO TECN TECNOLOGICO
DECOMTM (11.63.11)
Matrícula: ###884#9

Visualize o documento original em <https://sig.cefetmg.br/documentos/> informando seu número: 7, ano: 2025, tipo:
FOLHA DE APROVAÇÃO DE TCC, data de emissão: 21/02/2025 e o código de verificação: 36485c013a

.

.

.

Dedico a
algumas pessoas.

Agradecimentos

Em primeiro lugar, agradeço aos meus pais, Valéria e Adilson, que, mesmo à distância, me apoiaram ao longo de toda essa trajetória. Sou grato pelas horas de conversa, pelas distrações e pelos conselhos, que fazem cada vez mais sentido com o passar dos dias.

Agradeço também aos meus irmãos, Gabriel, Aloísio e Matheus, por serem companheiros incríveis e ombros nos quais pude me apoiar ao longo dos anos, tornando dias estressantes em momentos mais leves. Estendo minha gratidão a toda a minha família, pelo apoio incondicional, e aos meus queridos amigos, por sua presença e incentivo.

Agradeço, ainda, à minha companheira, Cíntia, pelo apoio e compreensão, sendo meu grande porto seguro.

Sou imensamente grato à minha professora orientadora, Viviane, cujo apoio e dedicação foram cruciais para minha trajetória acadêmica. Seu incentivo e orientação foram fundamentais para meu crescimento e aprendizado.

Por fim, agradeço a Deus.

“No mundo atual, muitos dos nossos problemas são causados pelo conhecimento e pela tecnologia.”.
Interstellar

Resumo

Este trabalho propõe um projeto de extensão no CEFET-MG voltado ao ensino de programação para alunos do 6º ao 9º ano do ensino fundamental, utilizando o Scratch como ferramenta didática. A pesquisa investiga as principais dificuldades enfrentadas por iniciantes no aprendizado de lógica e programação e analisa como a abordagem visual e interativa do Scratch pode facilitar esse processo.

O projeto resultará na criação de um caderno de atividades, abordando conceitos fundamentais como sequência, condicionais, laços de repetição e variáveis. A proposta busca democratizar o ensino da programação, tornando-o mais acessível e estimulante, ao mesmo tempo em que incentiva o desenvolvimento de habilidades de resolução de problemas desde a educação básica. Dessa forma, o projeto contribui para a inserção da programação no contexto escolar, preparando os alunos para desafios da era digital.

Palavras-chave: Scratch; Ensino de programação; Conceitos básicos de programação; Lógica de programação; Algoritmos; Estruturas de dados;

Abstract

This monograph proposes an extension project at CEFET-MG aimed at teaching programming to students from the 6th to the 9th grade of elementary school, using Scratch as a didactic tool. The research investigates the main difficulties faced by beginners in learning logic and programming and analyzes how Scratch's visual and interactive approach can facilitate this process.

The project will result in the creation of an activity book covering fundamental concepts such as sequencing, conditionals, loops, and variables. The proposal seeks to democratize programming education, making it more accessible and engaging while encouraging the development of problem-solving skills from an early age. In this way, the project contributes to integrating programming into the school context, preparing students for the challenges of the digital age.

Keywords: Scratch; Programming Education; Basic Programming Concepts; Programming Logic; Algorithms; Data Structures;

Lista de ilustrações

Figura 1 – Página Inicial do Scratch	17
Figura 2 – Comparação entre Scratch, C e Java	18
Figura 3 – Atividade 1	30
Figura 4 – Atividade 2	31
Figura 5 – Atividade 3	32
Figura 6 – Atividade 4	33
Figura 7 – Atividade 5	34
Figura 8 – Atividade 6	35
Figura 9 – Atividade 7	36
Figura 10 – Atividade 8	37
Figura 11 – Atividade 9	38
Figura 12 – Atividade 10	39
Figura 13 – Atividade 11	40
Figura 14 – Atividade 12	41
Figura 15 – Atividade 13	42
Figura 16 – Atividade 14	43
Figura 17 – Atividade 15	44
Figura 18 – Atividade 16	46
Figura 19 – Atividade 17	47
Figura 20 – Atividade 18	48

Lista de tabelas

Tabela 1 – Classificação das Atividades, tempo e modalidade	28
---	----

Sumário

1	INTRODUÇÃO	13
1.1	Objetivo Geral	13
1.2	Objetivos Específicos	14
1.3	Justificativa	14
2	REVISÃO DE LITERATURA	15
2.1	Fundamentação Teórica	15
2.1.1	Ensino de Programação	15
2.1.2	Scratch	16
2.1.3	O uso de Scratch no Ensino	18
2.2	Trabalhos Correlatos	19
3	METODOLOGIA	21
3.1	Tipos de pesquisa	21
3.2	Seleção de Conteúdos de Programação	21
3.2.1	Sequência Lógica	22
3.2.2	Estruturas Condicionais	22
3.2.3	Laços de Repetição	22
3.2.4	Variáveis	23
3.3	Desenvolvimento das Atividades	23
3.3.1	Seleção e Adaptação das Atividades	23
3.3.2	Estrutura das Atividades	24
3.3.3	Categorias de Atividades	24
3.3.3.1	Atividades de Introdução ao Scratch	24
3.3.3.2	Atividades de Conceitos de Programação	24
3.3.3.3	Projetos Criativos	25
4	SCRATCH NA PRÁTICA DE ENSINO	26
4.1	Proposta Pedagógica	26
4.2	Elaboração do Caderno de Atividades	28
4.2.1	Organização Geral do Caderno	29
4.2.2	Design Visual e Layout	29
4.3	Atividades Práticas	29
4.3.1	Atividade 1: Movimentação Básica	29
4.3.2	Atividade 2: Animar seu Nome	30
4.3.3	Atividade 3: Desenhando um Círculo	31
4.3.4	Atividade 4: Veloz e Furioso	32
4.3.5	Atividade 5: Desenhando um Quadrado	33
4.3.6	Atividade 6: Astronauta Artista	34

4.3.7	Atividade 7: Jogo de Perguntas	35
4.3.8	Atividade 8: Desafio de Lógica	36
4.3.9	Atividade 9: Movimentos no Plano Cartesiano	37
4.3.10	Atividade 10: Calculando a Média	38
4.3.11	Atividade 11: Operações Matemáticas	39
4.3.12	Atividade 12: Par ou Ímpar	40
4.3.13	Atividade 13: Identificando o Maior Número	41
4.3.14	Atividade 14: Calculando a Média e Verificando Aprovação	42
4.3.15	Atividade 15: Calculando a Média e Verificando Aprovação com Recuperação	43
4.3.16	Atividade 16: Sorteio e Comparação de Números	44
4.3.17	Atividade 17: Identificação do Maior e Menor Número	46
4.3.18	Atividade 18: Cálculo da Média de 4 Notas e Verificação de Aprovação	48
5	CONSIDERAÇÕES FINAIS	49
	REFERÊNCIAS	50
	APÊNDICES	53
	APÊNDICE A – CADERNO DE ATIVIDADES	54

1 Introdução

*“A resposta certa, não importa nada:
o essencial é que as perguntas estejam certas”.*
Mário Quintana

Segundo Lima (2019), a presença constante das inovações tecnológicas na vida cotidiana é inegável, com atualizações e transformações contínuas. Observando o cenário escolar, percebe-se que ferramentas como jogos digitais, softwares educativos e plataformas de aprendizado virtual não apenas auxiliam, mas também transformam e enriquecem a experiência educacional, oferecendo novos caminhos para o ensino e a aprendizagem.

A presença das tecnologias digitais tem se tornado cada vez mais marcante em diversos aspectos da vida cotidiana, seja no âmbito do entretenimento, da educação ou do trabalho. O progresso tecnológico, ao longo do tempo, tem transformado a maneira como as pessoas interagem com o mundo ao seu redor, modificando comportamentos e redefinindo suas necessidades e expectativas. Em meio a esse cenário de mudanças contínuas, também se observa um crescente interesse acadêmico no campo da computação, refletido tanto no aumento de matrículas em cursos de graduação dessa área quanto na expansão de pesquisas que investigam a aplicação de conceitos computacionais em diversos campos do saber (INEP, 2022).

Sendo assim, o Scratch, uma linguagem de programação visual desenvolvida pelo MIT Media Lab, tem se destacado como uma das ferramentas mais eficazes para a introdução de conceitos de programação de forma acessível e lúdica (RESNICK et al., 2009). Através de uma interface baseada em blocos visuais, o Scratch permite que os alunos criem animações, jogos e histórias interativas, sem a necessidade de dominar sintaxes complexas de linguagens de programação tradicionais. (BRENNAN; RESNICK, 2012).

Com base nessas ideias e na proposta de que ferramentas como o Scratch podem ser determinantes na consolidação do aprendizado computacional de forma inclusiva e prática, este trabalho propõe a seguinte pergunta de pesquisa: Como a plataforma Scratch pode ser utilizada como uma ferramenta educacional eficaz para introduzir crianças do 6º ao 9º ano do ensino fundamental aos conceitos básicos de programação, de forma lúdica e acessível?

1.1 Objetivo Geral

Desenvolver um caderno de atividades utilizando a plataforma Scratch, com o intuito de ensinar conceitos básicos de programação para crianças do 6º ao 9º ano do ensino fundamental de forma lúdica e acessível, visando à aplicação do material como um futuro projeto de extensão pelo CEFET-MG.

1.2 Objetivos Específicos

Também, objetivam-se mais especificamente:

1. Investigar as principais dificuldades enfrentadas por crianças do 6º ao 9º ano do ensino fundamental no aprendizado inicial de programação e lógica.
2. Explorar a plataforma Scratch e selecionar as funcionalidades que melhor se adequem ao ensino de conceitos de programação básicos para o público do 6º ao 9º ano do ensino fundamental
3. Definir e organizar conceitos de programação fundamentais, como sequência, condicionais, laços de repetição e variáveis.
4. Estruturar o caderno de atividades de maneira didática, com seções introdutórias, práticas, permitindo um desenvolvimento no aprendizado.

1.3 Justificativa

O ensino de programação tem se tornado cada vez mais essencial em um mundo altamente digitalizado e dependente de tecnologia. No entanto, a complexidade inicial da programação pode ser um obstáculo significativo para iniciantes. O Scratch oferece uma abordagem amigável e interativa para introduzir conceitos básicos de programação. Através de blocos visuais e uma interface intuitiva, o Scratch permite que os alunos se concentrem na lógica e na resolução de problemas sem se preocuparem com a sintaxe complexa das linguagens de programação tradicionais (RESNICK et al., 2009).

A utilização do Scratch no ensino fundamental e médio pode democratizar o acesso ao aprendizado de programação, permitindo que mais estudantes desenvolvam habilidades críticas para o século XXI (BRENNAN; RESNICK, 2012).

Além disso, o Scratch tem se mostrado eficaz em contextos educacionais diversos, promovendo o desenvolvimento do pensamento computacional e habilidades como criatividade e colaboração. Para crianças e adolescentes, aprender a programar através de jogos e animações oferece uma experiência divertida e interativa, estimulando o interesse e o engajamento com o aprendizado de forma significativa (FARIAS; OLIVEIRA; SILVA, 2018a)

Neste contexto, o presente trabalho propõe a criação de um projeto de extensão para o CEFET-MG, voltado ao ensino de programação para crianças do 6º ao 9º ano do ensino fundamental por meio de um caderno de atividades em Scratch. Esse projeto visa democratizar o acesso ao aprendizado de programação, utilizando o Scratch para ensinar conceitos básicos de forma lúdica e acessível. O caderno de atividades, produto final do projeto, será estruturado com uma progressão de tarefas práticas e interativas que facilitarão o aprendizado de programação, contribuindo para o desenvolvimento de habilidades cognitivas e tecnológicas desde cedo.

2 Revisão de Literatura

A fundamentação teórica do presente trabalho se apoia em diversos conceitos e estudos relacionados ao uso de ferramentas visuais no ensino de programação, com ênfase no Scratch. A seguir, serão discutidos os principais elementos teóricos que sustentam a relevância e a eficácia do Scratch como ferramenta educacional.

2.1 Fundamentação Teórica

2.1.1 Ensino de Programação

O avanço das tecnologias digitais tem transformado diversas áreas da sociedade, e o campo educacional não é uma exceção. A programação, que antes era restrita a contextos técnicos e acadêmicos, vem ganhando destaque como uma habilidade essencial a ser desenvolvida desde cedo (ANJOS et al., 2024).

O primeiro contato dos alunos com uma linguagem de programação pode ser desafiador e exige um esforço conjunto entre professor e aluno, com uma prática constante para a compreensão do conteúdo. No Curso Técnico Integrado em Informática, os estudantes enfrentam dificuldades na disciplina de Lógica de Programação, que se acumulam ao longo do curso, dificultando a compreensão dos conteúdos subsequentes se os anteriores não forem bem entendidos. Essas dificuldades podem estar relacionadas à falta de conhecimento prévio no Ensino Fundamental, essencial para uma base sólida em programação. A Lógica de Programação está intimamente ligada à Matemática, e a falta de domínio nesta área pode prejudicar a aplicação prática dos conceitos de programação. Além disso, uma boa interpretação de texto e conhecimento básico de Inglês são importantes para a análise de enunciados e compreensão das funções das linguagens de programação, geralmente expressas em inglês (JUNIOR, 2016).

Aprender a programar é um processo complexo que exige do estudante um constante exercício de raciocínio lógico e capacidade de abstração para resolver problemas. Este aprendizado envolve interpretar, analisar requisitos e apresentar soluções algorítmicas, além de entender conceitos fundamentais como estruturas de controle, recursão e ponteiros. Além dessas dificuldades, o aprendiz também deve lidar com a sintaxe das linguagens de programação e com as ferramentas de edição, compilação/interpretação e depuração de código (KHOURI; SANTOS; BARBOSA, 2020).

No nível de graduação, uma das principais dificuldades que alunos provenientes do ensino médio enfrentam é a deficiência na base de conhecimentos fundamentais necessários para entender conceitos computacionais. Além disso, esses alunos frequentemente encontram dificuldades em desenvolver raciocínio lógico, especialmente quando estão acostumados a memorizar conteúdos. A dificuldade em compreender as abstrações necessárias para a programação também é um problema recorrente. Portanto, é essencial que os docentes adotem

um papel motivador, ajustando suas estratégias de ensino para tornar os conteúdos mais claros e compreensíveis. (HOED, 2016)

Já Farias, Oliveira e Silva (2018b) reiteram que as dificuldades enfrentadas por estudantes de cursos técnicos e superiores de informática no aprendizado de programação estão relacionadas a diversos fatores, incluindo aspectos dos alunos, dos professores e das metodologias de ensino. Entre os fatores destacados estão a base lógico-matemática deficiente dos alunos, a falta de dedicação aos estudos, as limitações dos professores e a ineficácia dos materiais didáticos de apoio.

Brackmann (2017) afirma que a programação está se tornando uma habilidade extremamente valiosa no mercado de trabalho atual. Profissionais que possuem conhecimento em programação podem desenvolver soluções para problemas complexos, contribuir em projetos tecnológicos e se adaptar rapidamente às mudanças do setor. Além disso, aprender a programar aprimora habilidades cruciais como raciocínio lógico, resolução de problemas e criatividade. Essas competências são úteis em diversas áreas e podem ajudar os profissionais de tecnologia a se destacarem em suas carreiras.

Diante dessas considerações, Gomes (2010) diz que é fundamental que as instituições de ensino ofereçam uma educação em computação que seja acessível e inclusiva, permitindo que todos os alunos, independentemente de seu conhecimento prévio ou habilidades técnicas, possam ter acesso a um aprendizado de qualidade. Garantir que os cursos técnicos e superiores na área da computação atendam a essas necessidades é uma tarefa essencial para promover um ambiente educacional eficaz e equitativo.

2.1.2 Scratch

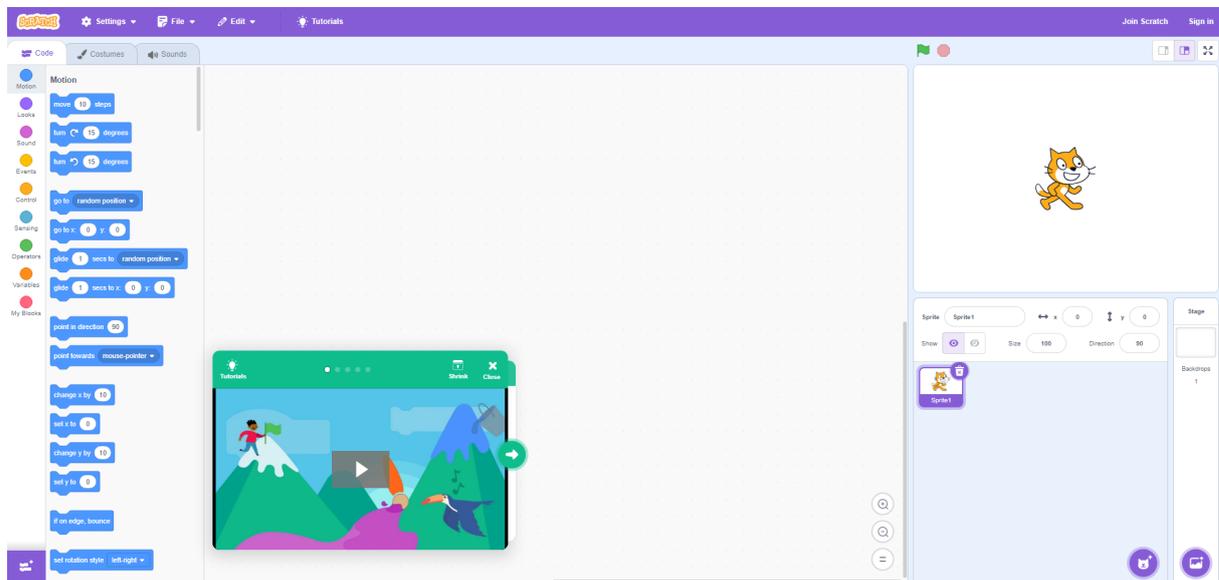
O Scratch é uma ferramenta educacional baseada em uma linguagem de programação visual, desenvolvida pelo grupo Lifelong Kindergarten do MIT Media Lab. Lançada em 2007 com o lema “imaginar, programar, compartilhar”, essa plataforma gratuita tem como objetivo incentivar pessoas de todas as idades a produzir conhecimento utilizando o computador (REZENDE; BISPO, 2018). O Scratch oferece uma ampla gama de possibilidades, como liberdade criativa, comunicação, e compartilhamento, além de facilitar a aprendizagem de conceitos escolares e a manipulação de mídias. Usuários podem compartilhar e adaptar projetos existentes, e integrar objetos físicos (SCRATCH, 2017).

Devido à sua abordagem lúdica, o Scratch é uma excelente ferramenta para o ensino de programação, particularmente para introduzir a lógica de programação e fomentar o pensamento computacional nos estágios iniciais da educação em computação. A linguagem do Scratch utiliza elementos visuais e multimídia, tornando o ensino e a aprendizagem de programação simples e eficazes por meio da programação de “blocos lógicos” que representam diferentes funcionalidades (REZENDE; BISPO, 2018).

A interface gráfica intuitiva do Scratch, como mostrado na figura 1, facilita o aprendizado de programação para iniciantes, especialmente crianças. Diferente de linguagens de programação tradicionais como Java e C, onde o código é escrito caractere por caractere, o

Scratch simplifica esse processo (AMBRÓSIO; COSTA, 2010).

Figura 1 – Página Inicial do Scratch



Fonte: Do próprio autor

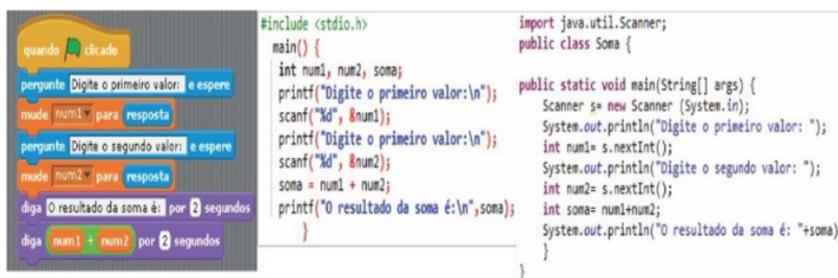
O sucesso do Scratch como ferramenta de ensino pode ser atribuído, em grande parte, ao seu enfoque no aprendizado através da construção e exploração. Papert, em sua teoria do construcionismo, já defendia que os alunos aprendem de forma mais efetiva quando estão ativamente envolvidos na criação de projetos significativos (PAPERT, 1980). O Scratch incorpora esse princípio, permitindo que os alunos desenvolvam projetos de interesse próprio, seja um jogo ou uma história interativa, promovendo um aprendizado significativo, no qual os alunos têm um papel ativo e criativo (BRENNAN; RESNICK, 2012). Em consonância com a visão de Papert, Salazar, Odakura e Barvinski (2015) destacam que o ambiente do Scratch incentiva os alunos a experimentarem livremente, construindo conhecimento a partir de suas próprias experiências e criando um senso de realização ao final de cada projeto.

A popularidade e o sucesso do Scratch têm incentivado inúmeras iniciativas educacionais pelo mundo, promovendo-o como uma ferramenta padrão para introdução à programação. A pesquisa de Souza (2019) mostra que o Scratch, em projetos escolares e extracurriculares, promove um ambiente de aprendizado que não só desenvolve habilidades técnicas, mas também habilidades socioemocionais, como trabalho em equipe e capacidade de adaptação. Por meio do Scratch, as instituições de ensino têm conseguido integrar a programação de forma inclusiva, permitindo que mais jovens tenham acesso a essa área e possam explorar seu potencial criativo e técnico.

O Scratch, apesar de ser uma linguagem de programação visual baseada em blocos, apresenta várias similaridades com linguagens tradicionais como C e Java, conforme ilustrado na figura 2. Assim como no Scratch, essas linguagens utilizam variáveis para armazenar valores, operações matemáticas para manipular dados e comandos de entrada e saída para interagir com o usuário. Por exemplo, a estrutura de entrada de dados em Scratch ("Pergunte

e espere") pode ser comparada ao uso de `scanf` em C e ao `Scanner` em Java, ambos usados para capturar valores fornecidos pelo usuário. Além disso, a lógica sequencial de execução é mantida, com instruções dispostas em ordem linear e clara. Essas semelhanças tornam o Scratch uma ferramenta introdutória eficiente, ajudando os iniciantes a compreenderem conceitos fundamentais que podem ser transferidos para linguagens de programação mais avançadas (ZANETTI et al., 2021).

Figura 2 – Comparação entre Scratch, C e Java



Fonte: (ZANETTI et al., 2021)

2.1.3 O uso de Scratch no Ensino

A utilização do Scratch no ensino de matemática proporciona uma maneira interativa de explorar conceitos matemáticos, como geometria e álgebra, por meio da programação em blocos. Essa abordagem facilita a resolução de problemas de maneira prática e lúdica, promovendo uma aprendizagem mais significativa. Segundo Corrêa (2021), a utilização de ferramentas como o Scratch na sala de aula permite que os alunos participem ativamente da construção de seu conhecimento matemático, favorecendo um ambiente de aprendizado mais dinâmico e criativo.

No ensino de Química, a aplicação de ferramentas como o Scratch pode ser uma estratégia eficaz para tornar o aprendizado mais atrativo e colaborativo. Conforme aponta Medeiros (2021), a utilização de recursos digitais, como a construção de jogos no Scratch, pode auxiliar na compreensão de conceitos complexos e abstratos, como os da Química, ao transformar o processo educativo em algo mais dinâmico e interativo. Essa abordagem proporciona aos alunos a oportunidade de aplicar o conteúdo de forma prática, promovendo uma aprendizagem significativa e conectada ao cotidiano dos estudantes.

No contexto educacional voltado ao atendimento de estudantes com Transtorno do Espectro Autista, o uso de ferramentas tecnológicas como o Scratch pode desempenhar um papel essencial na promoção de um ambiente de aprendizagem inclusivo e acessível. Silva (2022) destaca que, ao integrar o lúdico e as Tecnologias Digitais de Informação e Comunicação, os educadores podem favorecer o desenvolvimento de habilidades cognitivas e sociais. Essas ferramentas permitem criar experiências de aprendizagem adaptadas às necessidades individuais, facilitando o processo de ensino e promovendo a inclusão.

Além das suas funcionalidades visuais intuitivas, o Scratch tem sido amplamente utilizado para ensinar os primeiros conceitos de programação. Ele permite que alunos desenvol-

vam habilidades de pensamento lógico, resolução de problemas e criatividade sem a complexidade de sintaxes textuais rígidas. Isso é especialmente importante no início da jornada de aprendizado, onde muitos estudantes se sentem desencorajados pelas dificuldades encontradas em linguagens tradicionais. Segundo Salazar, Odakura e Barvinski (2015), a abordagem visual do Scratch possibilita que os alunos experimentem, cometam erros e refinem suas soluções de maneira mais fluida, o que promove um ambiente de aprendizagem mais motivador e dinâmico.

Quando aplicado ao ensino superior, o Scratch apresenta vantagens ao abordar as dificuldades comuns enfrentadas por estudantes em disciplinas introdutórias de programação, como Algoritmos e Lógica de Programação. Embora tradicionalmente associado ao ensino básico, o Scratch tem demonstrado ser uma ferramenta eficaz para facilitar a compreensão de conceitos complexos em cursos de graduação. Salazar, Odakura e Barvinski (2015) observam que, mesmo que o uso da ferramenta não resulte sempre em melhorias diretas nas notas, ela desempenha um papel fundamental no aumento do engajamento e da motivação dos alunos, fatores críticos para o sucesso acadêmico a longo prazo.

Embora o Scratch seja amplamente aceito como uma ferramenta eficaz para o ensino de programação, ele também apresenta alguns desafios e limitações. Rezende e Bispo (2018) apontam que, embora o Scratch seja útil para introduzir lógica e conceitos básicos de programação, ele pode não ser suficientemente robusto para ensinar certos tópicos avançados, como estruturas de dados complexas ou algoritmos. Além disso, para alunos que já possuem um conhecimento prévio em programação, o Scratch pode ser considerado uma ferramenta básica demais, limitando a exploração de habilidades mais avançadas. Assim, é importante que educadores adaptem o uso do Scratch de acordo com o nível de conhecimento dos alunos, garantindo que a ferramenta complemente, em vez de substituir, o aprendizado de linguagens de programação mais complexas.

Em resumo, o Scratch tem se mostrado uma ferramenta de grande valor no ensino de programação e em projetos interdisciplinares, com ampla aplicabilidade e benefícios pedagógicos. Ao incentivar o aprendizado prático e visual, ele facilita a compreensão de conceitos abstratos e promove o engajamento e a colaboração entre os alunos. No entanto, educadores devem considerar as limitações da ferramenta e adaptar as atividades para assegurar que todos os alunos, independentemente do nível de habilidade, possam tirar o máximo proveito do aprendizado oferecido pelo Scratch (AMBRÓSIO; COSTA, 2010; MEDEIROS, 2021)

2.2 Trabalhos Correlatos

1. A dissertação de mestrado “O uso do Scratch no ensino de programação” de Anastacio (2020) explora o impacto do Scratch no ensino de programação. A pesquisa destaca que os alunos enfrentam dificuldades significativas em disciplinas de programação, levando a altos índices de reprovação e evasão. Anastacio aponta que o Scratch pode motivar o aprendizado de programação de forma mais agradável, promovendo o Pensamento Computacional através de etapas como Decomposição, Reconhecimento de

Padrões, Abstração e Algoritmo. Além de beneficiar o desempenho acadêmico, o Pensamento Computacional ajuda na tomada de decisões e resolução de problemas cotidianos. A pesquisa incluiu um caderno de atividades para o Scratch e um curso introdutório para alunos de Ciência da Computação, cujos resultados qualitativos indicaram a eficácia do caderno em desenvolver o Pensamento Computacional e confirmar a adequação do Scratch no ensino de programação.

2. A dissertação “Ensino e Tecnologias: aprendizagem significativa por meio da plataforma Scratch” de Souza (2019) investiga a possibilidade de promover a autonomia e a aprendizagem significativa através do uso da tecnologia na educação. O estudo explora como a plataforma Scratch pode contribuir para as práticas pedagógicas, envolvendo atividades com alunos de pós-graduação e do Ensino Fundamental. Os objetivos incluem analisar as percepções de professores e alunos sobre o Scratch, desenvolver oficinas para capacitar o uso da plataforma, criar um “Estúdio” para interação entre participantes, e elaborar um guia para orientadores. A pesquisa, com abordagem descritiva e qualitativa, utiliza questionários e observações para coletar dados. Os resultados mostram que a integração do Scratch na educação pode fomentar a autonomia, criatividade e colaboração, impactando positivamente a qualidade do ensino.
3. A dissertação “O uso de programação com o Scratch na sala de aula de Matemática no Ensino Básico: Um estudo de atividades com possibilidades de utilização no Ensino Fundamental II” de Costa (2022) explora a aplicação do software Scratch no ensino de matemática para o Ensino Fundamental II. A pesquisa foca na seleção, análise e discussão de dissertações que tratam da utilização do Scratch em atividades matemáticas. O objetivo principal foi desenvolver um produto educacional na forma de um site, com o intuito de auxiliar professores a integrar o Scratch em suas práticas pedagógicas. O site oferece atividades de diversos autores que abordam competências da Base Nacional Comum Curricular de maneira simplificada e objetiva. A pesquisa utilizou a Revisão Sistemática de Literatura e analisou 14 dissertações para criar o produto educacional. Baseando-se na metodologia de ensino de Papert, que enfatiza o Construcionismo e o uso de computadores na educação, a pesquisa busca facilitar o trabalho docente e incentivar o uso do Scratch como ferramenta educacional.

3 Metodologia

3.1 Tipos de pesquisa

Neste trabalho, foram adotados diferentes tipos de pesquisa para garantir uma investigação abrangente e coerente sobre o uso do Scratch no ensino de conceitos básicos de programação para estudantes do ensino técnico e superior.

Conforme Gil (2008), a pesquisa exploratória tem como objetivo proporcionar uma melhor compreensão do problema, sendo utilizada para a formulação de hipóteses e direcionamentos futuros. Sua característica principal é a flexibilidade, permitindo ao pesquisador familiarizar-se com o fenômeno em questão e explorar áreas que ainda são pouco investigadas, funcionando como uma primeira etapa investigativa.

A pesquisa bibliográfica, segundo Lakatos e Marconi (2003), consiste no levantamento de materiais já publicados, como livros, artigos científicos e outros documentos que tratem do tema em estudo. Esse tipo de pesquisa é fundamental para construir o embasamento teórico de um trabalho acadêmico, fornecendo uma visão abrangente sobre o estado atual do conhecimento e identificando lacunas ou divergências na literatura existente.

Essa combinação de métodos de pesquisa fornece uma base robusta para o desenvolvimento e avaliação das atividades baseadas no Scratch, assegurando que o estudo seja conduzido com rigor científico e relevância prática.

3.2 Seleção de Conteúdos de Programação

A seleção dos conteúdos de programação para o caderno de atividades destinado a crianças do 6º ao 9º ano do ensino fundamental foi fundamentada em referências que destacam a importância de introduzir conceitos básicos de programação de forma lúdica e acessível. Estudos indicam que o uso de ferramentas como o Scratch facilita o aprendizado, pois permite que os alunos criem histórias, jogos e animações interativas, promovendo o desenvolvimento do raciocínio lógico e da criatividade.(SILVA, 2024)

A seleção foi orientada por dois critérios principais:

- **Adequação ao Nível de Compreensão do Público Alvo:** Optou-se por conceitos de programação que são acessíveis a iniciantes, permitindo que os alunos avancem sem necessidade de conhecimento prévio em programação.
- **Funcionalidades do Scratch:** A escolha dos tópicos foi orientada pelas capacidades da plataforma Scratch, que oferece blocos visuais para representar conceitos básicos de programação.

A escolha dos conteúdos foi orientada pela necessidade de oferecer uma introdução acessível à lógica computacional, garantindo que os alunos possam compreender e aplicar conceitos fundamentais sem dificuldades. Além disso, o desenvolvimento cognitivo na faixa etária do 6º ao 9º ano do ensino fundamental são apontados com a importância de metodologias ativas e aprendizagem baseada em projetos para o engajamento dos estudantes. O Scratch, por sua abordagem intuitiva e visual, permite que os alunos desenvolvam habilidades essenciais de programação de maneira lúdica e interativa, o que contribui para a construção de um conhecimento sólido e aplicável em diferentes contextos educacionais (ZANETTI et al., 2021). Com base nesses critérios e na fundamentação teórica, foram selecionados os seguintes conceitos de programação:

3.2.1 Sequência Lógica

A sequência lógica é o conceito fundamental de que as ações em um programa ocorrem em uma ordem específica, sendo executadas uma após a outra. No contexto do Scratch, a sequência é representada por blocos que são organizados verticalmente e executados em ordem, o que facilita a visualização do fluxo do programa (RESNICK et al., 2009).

A introdução à sequência lógica será abordada nas atividades iniciais do caderno, com projetos simples que ensinam aos alunos que a ordem dos blocos de comandos influencia diretamente o comportamento do projeto. Estudos mostram que a compreensão de sequência é um dos primeiros passos para desenvolver o raciocínio lógico em programação (BRENNAN; RESNICK, 2012). Exemplos de atividades incluem criar uma animação em que um personagem se move de um ponto a outro, executando ações em uma ordem definida.

3.2.2 Estruturas Condicionais

As estruturas condicionais permitem que o programa tome decisões com base em condições específicas, como “se... então” (REZENDE; BISPO, 2018). No Scratch, esses blocos condicionais são visualizados de forma acessível e permitem que o aluno crie programas que respondem a eventos, como pressionar uma tecla ou clicar em um objeto.

Esse conceito será trabalhado em atividades intermediárias, que apresentam aos alunos a ideia de “se acontecer X, então faça Y”. Por exemplo, uma atividade típica pode envolver a criação de um jogo em que o personagem muda de direção ou executa uma ação quando encontra um obstáculo. Essa experiência prática com condicionais ajuda os alunos a entender a lógica de tomada de decisões, um conceito fundamental em programação (ANASTACIO, 2020).

3.2.3 Laços de Repetição

Os laços de repetição são estruturas que permitem a repetição de uma série de comandos até que uma condição específica seja atendida ou indefinidamente. No Scratch, esses laços são representados por blocos como “repita até” e “para sempre”, que visualmente delimitam o início e o fim da repetição de comandos (AMBRÓSIO; COSTA, 2010).

A introdução aos laços será feita por meio de atividades que envolvem movimentos repetitivos ou sequências de animações. Por exemplo, os alunos podem criar uma animação onde um personagem caminha em um padrão circular ou salta indefinidamente em resposta a uma tecla. Ao entender os laços de repetição, os alunos aprendem a otimizar o código, substituindo múltiplos comandos por um bloco único, o que é um conceito importante para a programação eficiente (CORRÊA, 2021).

3.2.4 Variáveis

As variáveis são um conceito básico de armazenamento de informações que pode ser manipulado e exibido conforme o programa é executado. No Scratch, as variáveis são visualmente representadas e permitem que os alunos compreendam de forma prática como valores podem ser alterados e utilizados ao longo do programa (BRENNAN; RESNICK, 2012).

Esse conceito será explorado em atividades mais avançadas, onde os alunos aprenderão a criar e utilizar variáveis para contar pontos em um jogo ou armazenar a posição de um personagem. Estudos sugerem que introduzir variáveis de forma visual e prática facilita o entendimento dos alunos sobre como dados são utilizados e manipulados em um programa (MEDEIROS, 2021).

3.3 Desenvolvimento das Atividades

O desenvolvimento das atividades neste caderno será baseado na **compilação e adaptação de atividades existentes** sobre o ensino de programação com Scratch, produzidas por diversos autores e fontes, como de Anastacio (2020) e Souza (2019). A seleção das atividades será guiada por critérios pedagógicos que assegurem que o conteúdo seja adequado ao público infantojuvenil do 6º ao 9º ano do ensino fundamental, promovendo um aprendizado acessível. A seguir, são descritas as etapas e critérios para a seleção, organização e adaptação das atividades que comporão o caderno.

3.3.1 Seleção e Adaptação das Atividades

O processo de compilação das atividades envolverá a análise e escolha de materiais previamente desenvolvidos por pesquisadores, educadores e outras fontes confiáveis, que tenham como foco a introdução de conceitos básicos de programação no Scratch para crianças do 6º ao 9º ano do ensino fundamental (COSTA, 2022). A seleção das atividades será orientada pelos seguintes critérios:

- **Adequação ao Público-Alvo:** As atividades selecionadas devem ser apropriadas para crianças do 6º ao 9º ano do ensino fundamental, utilizando uma linguagem clara e visual, e alinhando-se às capacidades de compreensão deste público.
- **Relevância Pedagógica:** As atividades devem abordar conceitos fundamentais de programação, como sequência lógica, condicionais, laços de repetição e variáveis, garantindo que cada atividade contribua para o aprendizado progressivo dos alunos.

- **Metodologia Lúdica e Acessível:** As atividades precisam utilizar abordagens interativas e lúdicas, conforme proposto pela plataforma Scratch, facilitando o engajamento dos alunos e a assimilação dos conceitos.

3.3.2 Estrutura das Atividades

Para garantir que o caderno tenha uma estrutura didática e uniforme, foi proposto um formato como o do autor Anastacio (2020), cada atividade será adaptada e organizada em um formato padrão, que incluirá:

- **Introdução ao Conceito:** Uma breve contextualização do conceito de programação abordado, de modo que os alunos compreendam seu propósito antes de iniciarem a atividade.
- **Objetivo da Atividade:** Definição clara do objetivo de aprendizado da atividade, para orientar o aluno sobre o que será alcançado.
- **Instruções Passo a Passo:** As instruções originais das atividades serão adaptadas para manter a linguagem uniforme do caderno.

3.3.3 Categorias de Atividades

As atividades compiladas serão organizadas em três categorias principais, garantindo uma progressão de complexidade e facilitando o aprendizado gradual dos alunos:

3.3.3.1 Atividades de Introdução ao Scratch

Essa categoria reúne atividades que introduzem os alunos à interface do Scratch e aos blocos de comandos básicos. O objetivo é familiarizar os alunos com o ambiente da plataforma antes de abordarem conceitos mais complexos. Exemplos de atividades incluem:

- Navegar pela interface do Scratch e explorar os principais blocos;
- Realizar ações básicas de movimento e aparência com os personagens;
- Experimentar comandos simples de controle e som.

3.3.3.2 Atividades de Conceitos de Programação

A segunda categoria inclui atividades que abordam os conceitos fundamentais de programação. Cada atividade dessa seção ensinará um conceito específico, como:

- **Sequência Lógica:** Atividades que demonstram como organizar blocos em uma ordem lógica para realizar ações sequenciais.
- **Estruturas Condicionais:** Atividades que apresentam decisões condicionais, permitindo que o aluno compreenda “se... então” de forma prática.

- **Laços de Repetição:** Atividades que utilizam comandos de repetição, como “repita” e “para sempre”, mostrando aos alunos como simplificar comandos repetitivos.
- **Variáveis:** Atividades que introduzem a criação e manipulação de variáveis para armazenar e modificar dados durante o programa.

Cada atividade será escolhida com o intuito de ensinar e reforçar o aprendizado de um único conceito, assegurando uma abordagem focada e gradual.

3.3.3.3 Projetos Criativos

A última categoria é composta por atividades de projetos criativos, onde os alunos poderão aplicar de forma integrada os conceitos aprendidos nas atividades anteriores. Exemplos de projetos incluem:

- **Criação de um Jogo Simples:** Um jogo onde o aluno usará variáveis e condicionais para acumular pontos e interagir com o cenário.
- **Animações Interativas:** Atividades onde os alunos criam animações que respondem a comandos, permitindo a personalização e criação livre.
- **Histórias Interativas:** Atividades para criar histórias onde as ações dos personagens são guiadas por escolhas condicionais feitas pelo usuário.

Esses projetos são destinados a estimular a criatividade dos alunos e oferecer uma experiência prática, onde eles podem aplicar o que aprenderam em um contexto livre e divertido.

4 Scratch na prática de ensino

Nesta seção, apresentamos um conjunto de atividades práticas, que integram o caderno de atividades desenvolvido como parte deste trabalho. Estas atividades têm como objetivo introduzir conceitos fundamentais de programação, como sequência lógica, condicionais, laços de repetição e variáveis, utilizando o Scratch como plataforma educativa.

Cada atividade foi organizada para atender diferentes níveis de aprendizado, alternando entre atividades guiadas, realizadas com o suporte de um professor ou facilitador, e atividades autônomas, nas quais os alunos podem aplicar os conceitos aprendidos de forma independente.

Para ensinar com o Scratch, é essencial que o professor conheça a plataforma e seus recursos básicos. As aulas devem ser realizadas em um ambiente com computadores equipados com teclado, mouse e acesso à internet, sendo ideal um dispositivo por aluno ou para cada dois alunos. Para cada integrante ou dupla deve ser criado uma nova conta de acesso a plataforma. As contas podem ser criadas pelos próprios alunos ou pelo professor no site oficial da plataforma: <http://scratch.mit.edu/>. É indicado, também, que o professor ou facilitador veja o tutorial disponibilizado pela plataforma no link scratch.mit.edu/story para que experimente os primeiros passos (MIT Media Lab, 2023).

4.1 Proposta Pedagógica

Este caderno de atividades é um compilado de atividades pedagógicas baseadas em contribuições de diversos autores, como Anastacio (2020) e Souza (2019), com o objetivo de promover o ensino de conceitos básicos de programação de forma lúdica e acessível, utilizando a plataforma Scratch como ferramenta principal. A seleção e organização das atividades foram realizadas com base em materiais já existentes para atender ao público infantojuvenil do 6º ao 9º ano do ensino fundamental. Além de seu caráter didático, este material foi concebido como uma proposta para um projeto de extensão e indicado, também, como disciplina de ensino básico, possibilitando sua aplicação em escolas, centros comunitários e demais espaços educativos.

Os objetivos pedagógicos deste compilado são:

1. Introduzir os alunos à lógica de programação de forma progressiva, partindo de conceitos simples como sequência e repetição até tópicos mais avançados, como variáveis e condicionais;
2. Desenvolver habilidades cognitivas como resolução de problemas, criatividade e pensamento lógico por meio de atividades práticas e interativas;
3. Estimular a autonomia dos alunos ao alternar atividades guiadas, realizadas com o apoio

de um professor, e atividades autônomas, promovendo a aplicação prática dos conceitos aprendidos;

4. Fornecer aos professores e facilitadores um material estruturado e adaptável para uso em projetos educacionais, permitindo que as atividades sejam personalizadas para diferentes contextos e faixas etárias;
5. Incentivar a exploração criativa dos alunos, oferecendo oportunidades para que desenvolvam projetos próprios, como jogos, histórias interativas e animações.

Por meio dessa abordagem, busca-se criar um ambiente de aprendizado que não apenas introduza conceitos técnicos de programação, mas também motive os alunos a explorar suas ideias, resolver problemas de maneira criativa e desenvolver habilidades que serão úteis em diversas áreas da vida acadêmica e profissional. O caderno está alinhado com os princípios do ensino construcionista, que valorizam a aprendizagem por meio da criação de projetos significativos (ANASTACIO, 2020).

A implementação deste trabalho como um projeto de extensão tem o potencial de ampliar o acesso ao ensino de programação para crianças do 6º ao 9º ano do ensino fundamental, utilizando o Scratch como ferramenta didática. O caderno de atividades proposto pode ser aplicado em escolas, espaços comunitários e programas educacionais, proporcionando um ambiente de aprendizagem interativo e acessível. Conforme apresentado na Tabela 1, o caderno permite que os participantes avancem nos conceitos de programação, desde a introdução ao Scratch até a criação de projetos próprios.

Tabela 1 – Classificação das Atividades, tempo e modalidade

Aula e Atividade	Conceitos Teóricos	Tempo em Minutos	Modalidade
1. Movimentação Básica	Sequência Lógica, Laços	30-40	Com Professor
2. Animar seu Nome	Sequência Lógica, Laços	40-50	Com professor
3. Desenhando Círculo	Laços , Sequência Lógica	40-50	Aluno Sozinho
4. Veloz e Furioso	Condicionais, Variáveis	50-60	Com Professor
5. Desenhando Quadrado	Laços, Sequência Lógica	40-50	Aluno Sozinho
6. Astronauta Artista	Laços, Condicionais	50-60	Com Professor
7. Jogo de Perguntas	Condicionais, Validação	30-40	Aluno Sozinho
8. Desafio de Lógica	Variáveis, Condicionais	50-60	Com Professor
9. Plano Cartesiano	Coordenadas, Laços	40-50	Aluno Sozinho
10. Calculando Média	Variáveis, Operações	30-40	Com Professor
11. Operações Matemáticas	Condicionais, Operadores	40-50	Aluno Sozinho
12. Par ou Ímpar	Condicionais, Operadores	30-40	Com Professor
13. Identificando o Maior Número	Condicionais	50-60	Aluno Sozinho
14. : Calculando a Média	Variáveis, Condicionais	60-70	Com Professor
15.: Calculando a Média Avançado	Variáveis, Condicionais	40-50	Aluno Sozinho
16. : Sorteio e Comparação de Números	Variáveis, Condicionais	50-60	Com Professor
17. Identificação do Maior e Menor Número	Condicionais, Laços	50-60	Aluno Sozinho
18. Cálculo da Média de 4 Notas	Variáveis, Laços	60-70	Com Professor

Fonte: Do próprio autor

4.2 Elaboração do Caderno de Atividades

A elaboração do caderno de atividades segue uma organização didática e visual acessível, com o objetivo de facilitar o aprendizado de programação para crianças do 6º ao 9º ano do ensino fundamental, além de orientar professores e facilitadores.

4.2.1 Organização Geral do Caderno

A presente organização do caderno foi baseada na proposta de Souza (2019) e dividido nas seguintes partes principais:

- **Introdução ao Scratch e ao Caderno:** Apresenta a plataforma Scratch e o objetivo do caderno, incentivando os alunos a explorar as atividades de forma lúdica.
- **Instruções para Facilitadores:** Fornece orientações para professores sobre como usar o caderno, sugestões para adaptar as atividades a diferentes idades e estratégias para engajamento dos alunos.
- **Atividades por Categorias:** Organiza as atividades em três seções (Introdução ao Scratch, Conceitos de Programação e Projetos Criativos), com uma progressão de complexidade.

4.2.2 Design Visual e Layout

O caderno terá um design visual padronizado para tornar o material acessível e atraente:

- **Elementos Visuais:** Cada atividade incluirá capturas de tela e ícones dos blocos do Scratch para facilitar a compreensão.
- **Instruções Passo a Passo:** Orientações detalhadas e numeradas, com destaques visuais para os comandos principais.
- **Formato Consistente:** Todas as atividades seguirão o mesmo formato, com introdução, objetivo, instruções, desafios e resumo, promovendo uma navegação fácil e organizada.

Essa estrutura garante que o caderno seja prático, visualmente acessível e adequado para o aprendizado de programação de forma independente ou com o apoio de facilitadores.

4.3 Atividades Práticas

4.3.1 Atividade 1: Movimentação Básica

Objetivo: Ensinar aos alunos os conceitos básicos de programação no Scratch, introduzindo-os à interface da plataforma e ao uso de blocos de movimentação, laços de repetição e interação com o palco. O foco está em desenvolver um entendimento inicial sobre como criar movimento contínuo e como utilizar bordas como elementos interativos.

Descrição: Desenvolva um programa que faça o personagem mover 10 passos, esperar um segundo e trocar de traje. Caso toque na borda, ele deve virar e continuar andando.

Como fazer no Scratch:

1. Arraste o bloco “mova 10 passos” para o palco de programação.

2. Adicione o bloco “espere 1 segundo”.
3. Use o bloco “se tocar na borda, vire” para corrigir a direção do personagem.
4. Encaixe os blocos dentro de um laço “para sempre”.

Resultado:

Figura 3 – Atividade 1



Fonte: (ANASTACIO, 2020)

4.3.2 Atividade 2: Animar seu Nome

Objetivo: Introduzir a interface do Scratch, familiarizar o aluno com a área de programação e ensinar o uso básico de blocos de movimento, aparência e controle para criar uma animação personalizada.

Descrição: Nesta atividade, os alunos aprenderão a criar uma animação utilizando as letras do seu próprio nome como sprites. Cada letra terá um efeito visual diferente, permitindo que os alunos explorem recursos como mudança de cor, rotação e movimentação.

Como fazer no Scratch:

1. Insira as letras do seu nome como sprites no palco.
2. Para cada letra, programe blocos para movê-la, alterar sua cor e tamanho em intervalos.
3. Adicione um bloco de repetição para que as ações sejam contínuas.
4. Execute o programa clicando na bandeira verde e observe a animação do seu nome.

Resultado:

Figura 4 – Atividade 2



Fonte: (SOUZA, 2019)

4.3.3 Atividade 3: Desenhando um Círculo

Objetivo: Explorar conceitos matemáticos básicos, como ângulos e repetições, enquanto os alunos aprendem a utilizar laços e ferramentas de desenho no Scratch para criar formas geométricas simples, promovendo o raciocínio lógico e a relação entre código e geometria.

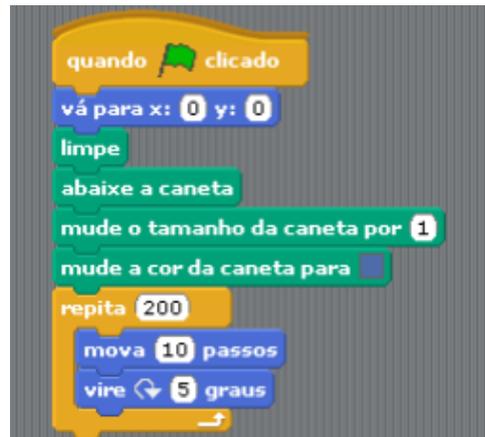
Descrição: Desenvolva um programa que mova o personagem e desenhe um círculo.

Como fazer no Scratch:

1. Use o bloco de movimento para avançar 10 passos e girar 5 graus.
2. Ative a ferramenta “caneta” para que o personagem desenhe no palco.
3. Repita o processo até completar um círculo.

Resultado:

Figura 5 – Atividade 3



Fonte: (ANASTACIO, 2020)

4.3.4 Atividade 4: Veloz e Furioso

Objetivo: Introduzir a programação de jogos simples no Scratch, com foco no uso de eventos e controle de movimentação.

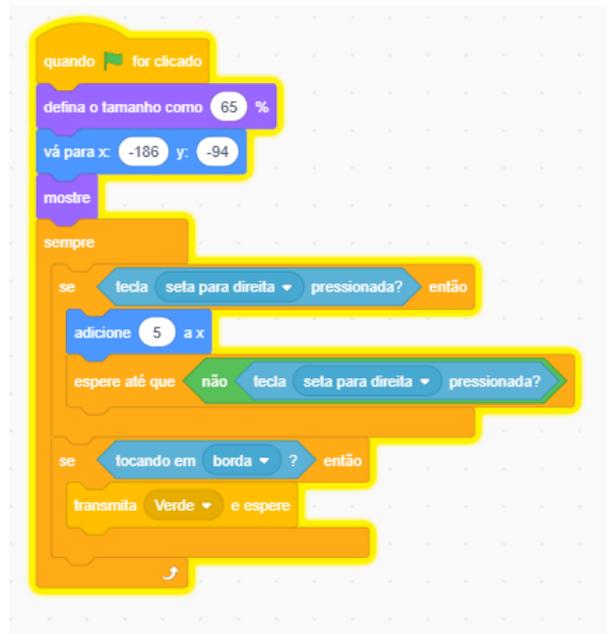
Descrição: Nesta atividade, os alunos criarão um jogo de corrida onde dois carros competem para chegar ao final da pista. Eles aprenderão a programar o movimento dos carros com o teclado e a definir condições para determinar o vencedor.

Como fazer no Scratch:

1. Adicione dois carros como sprites e configure um cenário de corrida.
2. Programe cada carro para se mover utilizando teclas específicas (ex.: seta para direita e barra de espaço).
3. Utilize blocos condicionais para determinar o vencedor ao alcançar o final da pista.
4. Exiba uma mensagem de vitória para o carro que chegar primeiro.

Resultado:

Figura 6 – Atividade 4



Fonte: (SOUZA, 2019)

4.3.5 Atividade 5: Desenhando um Quadrado

Objetivo: Aproximar os alunos da lógica por trás da programação estruturada, ensinando a criar formas geométricas usando ângulos e laços de repetição. Os alunos aprendem a planejar sequências de passos e aplicar noções de geometria, como ângulos retos.

Descrição: Desenvolva um programa que desenhe um quadrado ao ser executado.

Como fazer no Scratch:

1. Adicione um laço de repetição “repita 4 vezes”.
2. Programe o personagem para se mover 100 passos e girar 90 graus após cada movimento.
3. Utilize a ferramenta “caneta” para desenhar as linhas.

Resultado:

Figura 7 – Atividade 5



Fonte: (ANASTACIO, 2020)

4.3.6 Atividade 6: Astronauta Artista

Objetivo: Ensinar o uso de extensões no Scratch (como a ferramenta “Caneta”) para criar desenhos geométricos, estimulando a lógica matemática e a criatividade.

Descrição: Nesta atividade, os alunos programarão um astronauta para desenhar formas geométricas no espaço, utilizando a extensão “Caneta” do Scratch. Eles aprenderão a criar figuras como círculos, triângulos e quadrados ao pressionar diferentes teclas, explorando conceitos de ângulos, repetições e coordenação de movimentos.

Como fazer no Scratch:

1. Adicione o sprite de um astronauta e configure um cenário espacial.
2. Habilite a extensão “Caneta” no Scratch.
3. Programe o astronauta para desenhar formas geométricas básicas (círculos, triângulos, quadrados) ao pressionar diferentes teclas.
4. Adicione efeitos visuais e de som para enriquecer o projeto.

Resultado:

Figura 8 – Atividade 6



Fonte: (SOUZA, 2019)

4.3.7 Atividade 7: Jogo de Perguntas

Objetivo: Introduzir o conceito de entrada e validação de dados pelo usuário, utilizando blocos de "perguntar" e "se... então". Os alunos desenvolvem habilidades para criar interatividade em projetos e entender lógica condicional básica.

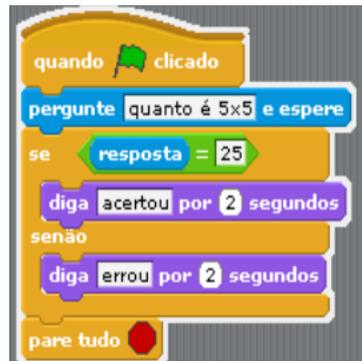
Descrição: Desenvolva um programa que pergunte “Quanto é 5 x 5?” ao usuário. Analise a resposta e informe se está correta.

Como fazer no Scratch:

1. Use o bloco “perguntar” para coletar a resposta.
2. Adicione um bloco condicional “se... então” para comparar a resposta com o valor esperado.
3. Exiba uma mensagem indicando se a resposta está certa ou errada.

Resultado:

Figura 9 – Atividade 7



Fonte: (ANASTACIO, 2020)

4.3.8 Atividade 8: Desafio de Lógica

Objetivo: Trabalhar o raciocínio lógico e ensinar o uso de variáveis no Scratch para criar um quiz interativo com feedback automático.

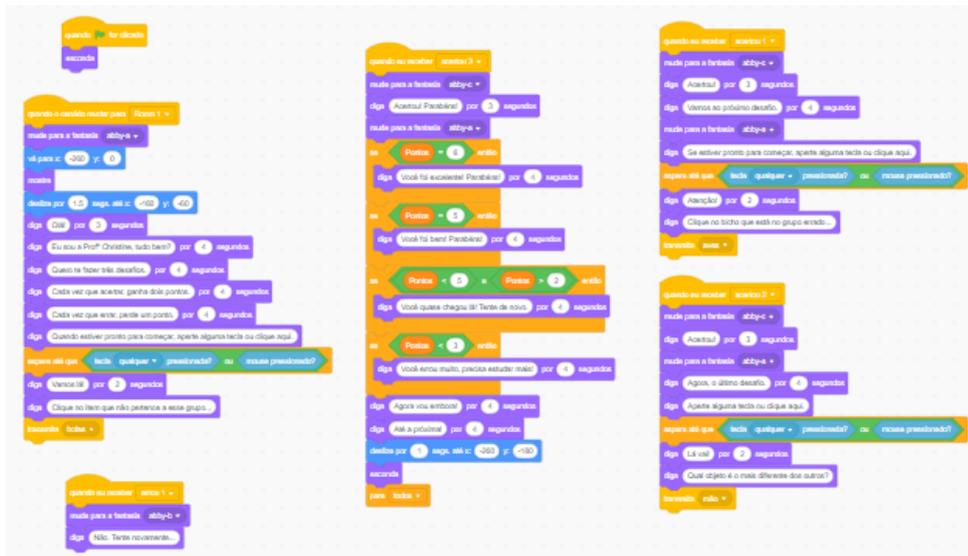
Descrição: Desenvolva um programa que funcione como um quiz interativo, onde o jogador responde a perguntas e recebe feedback automático baseado em sua pontuação.

Como fazer no Scratch:

1. Crie um cenário inicial e adicione um sprite de professor(a).
2. Programe perguntas com opções de resposta como sprites clicáveis.
3. Utilize variáveis para registrar o nome do aluno e a pontuação.
4. Programe blocos condicionais para exibir mensagens de feedback com base na pontuação final.

Resultado:

Figura 10 – Atividade 8



Fonte: (SOUZA, 2019)

4.3.9 Atividade 9: Movimentos no Plano Cartesiano

Objetivo: Ensinar o uso do plano cartesiano no Scratch para posicionamento de personagens. Os alunos desenvolvem habilidades para movimentar sprites usando coordenadas e integram laços de repetição e condicionais para criar padrões e interação.

Descrição: Crie um programa onde o personagem se move até a posição $x=-200$, $y=0$, troca de traje e repete esse movimento 5 vezes.

Como fazer no Scratch:

1. Use blocos de movimento para alterar a posição do personagem.
2. Adicione um laço para repetir o movimento 5 vezes.
3. Inclua um bloco de interação onde o personagem pergunta a idade do usuário e exibe uma mensagem baseada na resposta.

Resultado:

Figura 11 – Atividade 9



Fonte: (ANASTACIO, 2020)

4.3.10 Atividade 10: Calculando a Média

Objetivo: Introduzir o conceito de variáveis e operações matemáticas no Scratch. Os alunos aprendem a armazenar, manipular e exibir dados inseridos pelo usuário, consolidando a lógica por trás do uso de variáveis.

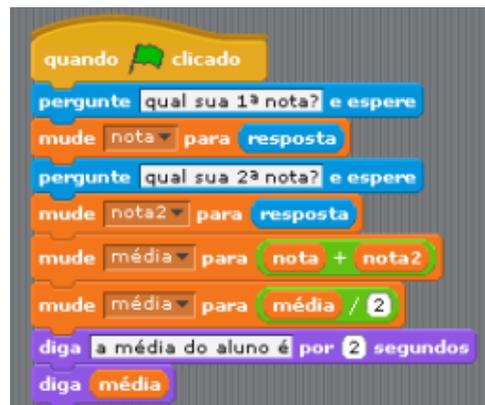
Descrição: Desenvolva um programa que calcula a média de duas notas inseridas pelo usuário e exibe o resultado.

Como fazer no Scratch:

1. Crie duas variáveis para armazenar as notas.
2. Use o bloco “perguntar” para coletar as notas do usuário.
3. Calcule a média e exiba o resultado.

Resultado:

Figura 12 – Atividade 10



Fonte: (ANASTACIO, 2020)

4.3.11 Atividade 11: Operações Matemáticas

Objetivo: Ensinar como utilizar blocos condicionais para realizar cálculos dinâmicos com base na escolha do usuário. Os alunos aprendem a planejar diferentes fluxos dentro de um programa, consolidando lógica condicional e o uso de operadores matemáticos.

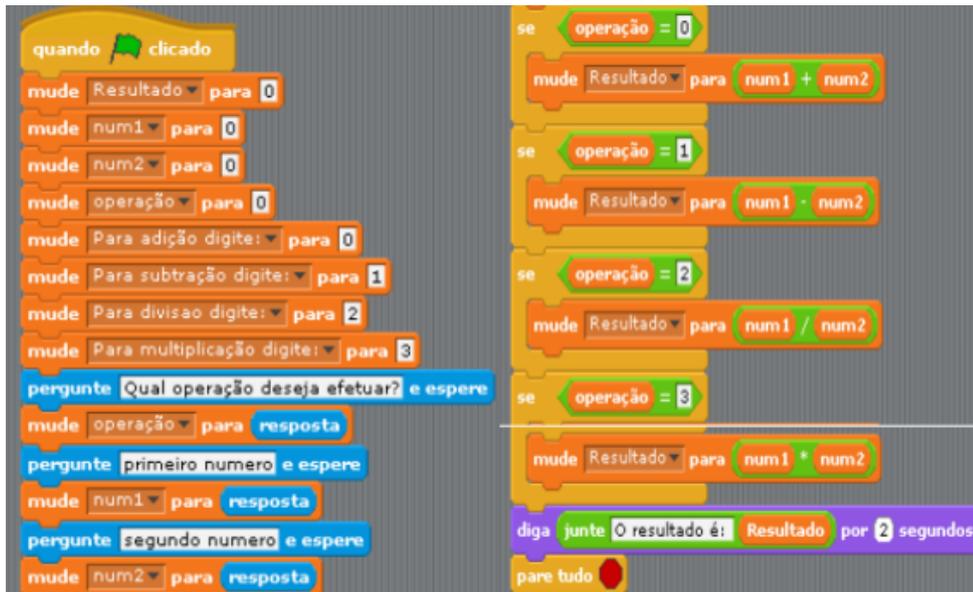
Descrição: Desenvolva um algoritmo onde o usuário escolhe uma operação (adição, subtração, multiplicação ou divisão) e insere dois valores. O programa realiza o cálculo e exibe o resultado.

Como fazer no Scratch:

1. Use o bloco “perguntar” para coletar a escolha da operação e os valores.
2. Adicione condicionais para realizar diferentes cálculos dependendo da operação escolhida.
3. Exiba o resultado final em um balão de fala.

Resultado:

Figura 13 – Atividade 11



Fonte: (ANASTACIO, 2020)

4.3.12 Atividade 12: Par ou Ímpar

Objetivo: Trabalhar com conceitos básicos de divisibilidade e lógica condicional. Os alunos aprendem a usar operadores matemáticos e blocos de decisão para criar programas que validam números inseridos pelo usuário.

Descrição: Desenvolva um programa que verifica se um número inserido pelo usuário é par ou ímpar.

Como fazer no Scratch:

1. Use o bloco “perguntar” para receber o número do usuário.
2. Adicione um bloco condicional para verificar se o número é divisível por 2.
3. Exiba a mensagem “par” ou “ímpar” de acordo com a validação.

Resultado:

Figura 14 – Atividade 12



Fonte: (ANASTACIO, 2020)

4.3.13 Atividade 13: Identificando o Maior Número

Objetivo: Ensinar o uso de variáveis, operadores lógicos e estruturas condicionais no Scratch. Os alunos aprenderão a solicitar entradas do usuário, comparar valores e exibir o maior número inserido.

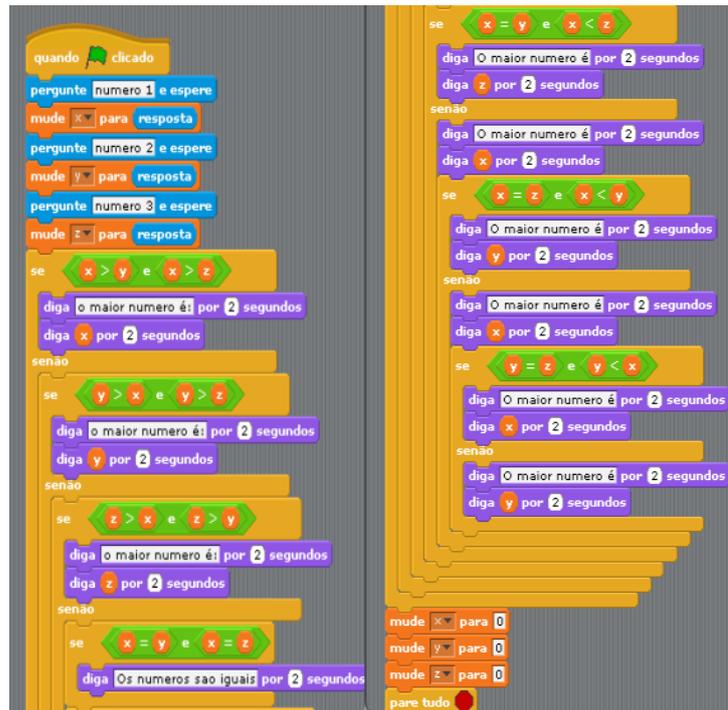
Descrição: Crie um algoritmo que permita ao usuário inserir três números e, em seguida, exiba o maior entre eles.

Como fazer no Scratch:

1. Crie três variáveis: "Número 1", "Número 2" e "Número 3".
2. Utilize blocos de entrada para que o usuário informe os três números.
3. Utilize estruturas condicionais para determinar qual dos três números é o maior.
4. Se "Número 1" for maior que "Número 2" e "Número 3", exiba "O maior número é: Número 1".
5. Se "Número 2" for maior que "Número 1" e "Número 3", exiba "O maior número é: Número 2".
6. Caso contrário, exiba "O maior número é: Número 3".
7. Se houver números iguais, exiba uma mensagem indicando isso.
8. Utilize blocos de aparência para exibir a resposta ao usuário.

Resultado:

Figura 15 – Atividade 13



Fonte: (ANASTACIO, 2020)

4.3.14 Atividade 14: Calculando a Média e Verificando Aprovação

Objetivo: Ensinar o uso de variáveis, operadores matemáticos e estruturas condicionais no Scratch. Os alunos aprenderão a coletar valores inseridos pelo usuário, calcular médias e determinar a situação de um aluno com base em regras de aprovação.

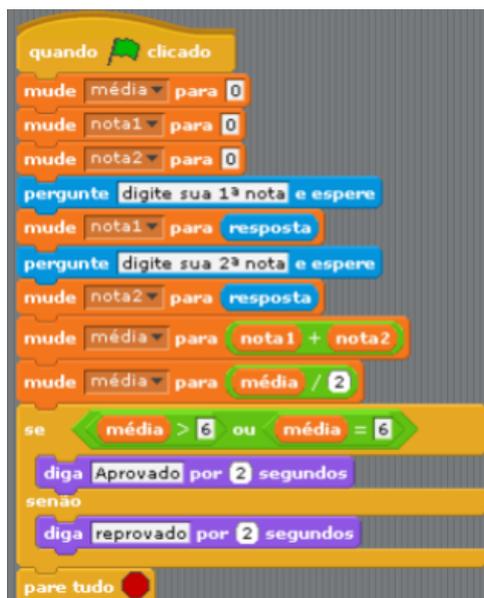
Descrição: Crie um algoritmo que solicite três notas do usuário, calcule a média e determine a situação do aluno com base nas regras de aprovação.

Como fazer no Scratch:

1. Crie duas variáveis: "Nota 1" e "Nota 2".
2. Utilize blocos de entrada para que o usuário insira os valores das notas.
3. Crie a variável "Média" e programe o cálculo: $Média = (Nota\ 1 + Nota\ 2) / 2$
4. Se a média for maior ou igual a 6, exiba "Aprovado".
5. Caso contrário, exiba "Reprovado".

Resultado:

Figura 16 – Atividade 14



Fonte: (ANASTACIO, 2020)

4.3.15 Atividade 15: Calculando a Média e Verificando Aprovação com Recuperação

Objetivo: Ensinar o uso de variáveis, operadores matemáticos e estruturas condicionais no Scratch. Os alunos aprenderão a coletar valores inseridos pelo usuário, calcular médias e determinar a situação de um aluno com base em regras de aprovação.

Descrição: Crie um algoritmo que solicite três notas do usuário, calcule a média e determine a situação do aluno com base nas regras de aprovação.

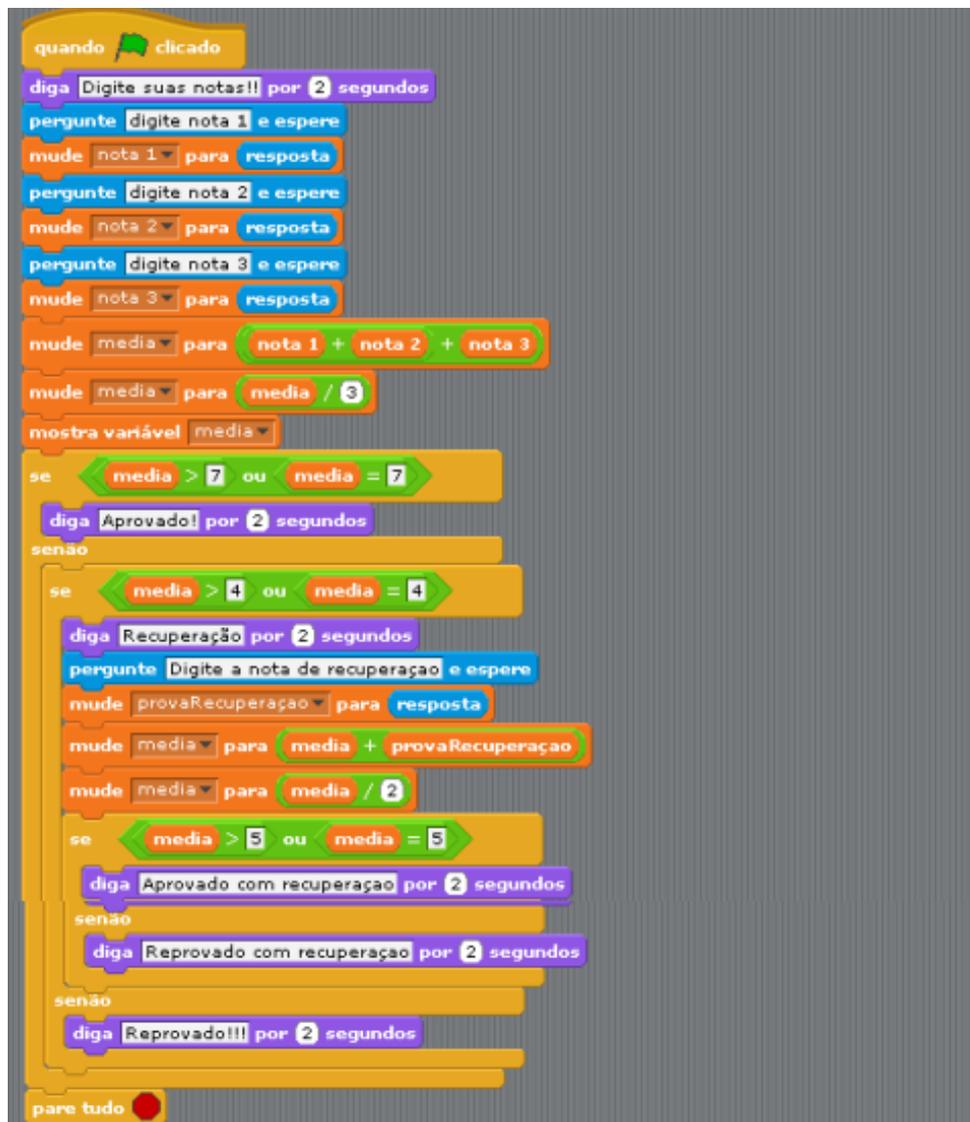
Como fazer no Scratch:

1. Crie três variáveis: "Nota 1", "Nota 2" e "Nota 3".
2. Utilize blocos de entrada para que o usuário insira os três valores.
3. Crie a variável "Média" e programe o cálculo: $Média = (Nota\ 1 + Nota\ 2 + Nota\ 3) / 3$
4. Se a média for maior ou igual a 7, exiba "Aprovado".
5. Se a média for maior ou igual a 4 e menor que 7, exiba "Em Exame" e permita que o usuário insira uma nova nota do exame.
6. Se a média for menor que 4, exiba "Reprovado".
7. Se o aluno precisar de exame, crie a variável "Nota do Exame" e solicite um novo valor.
8. Calcule a nova média: $Nova\ Média = (Média + Nota\ do\ Exame) / 2$
9. Se a nova média for maior ou igual a 5, exiba "Aprovado com Exame".

10. Caso contrário, exiba "Reprovado".

Resultado:

Figura 17 – Atividade 15



Fonte: (ANASTACIO, 2020)

4.3.16 Atividade 16: Sorteio e Comparação de Números

Objetivo: Ensinar o uso de variáveis, operadores matemáticos, geração de números aleatórios e estruturas condicionais no Scratch. Os alunos aprenderão a coletar valores inseridos pelo usuário, gerar números aleatórios e comparar os valores para determinar quantos foram acertados.

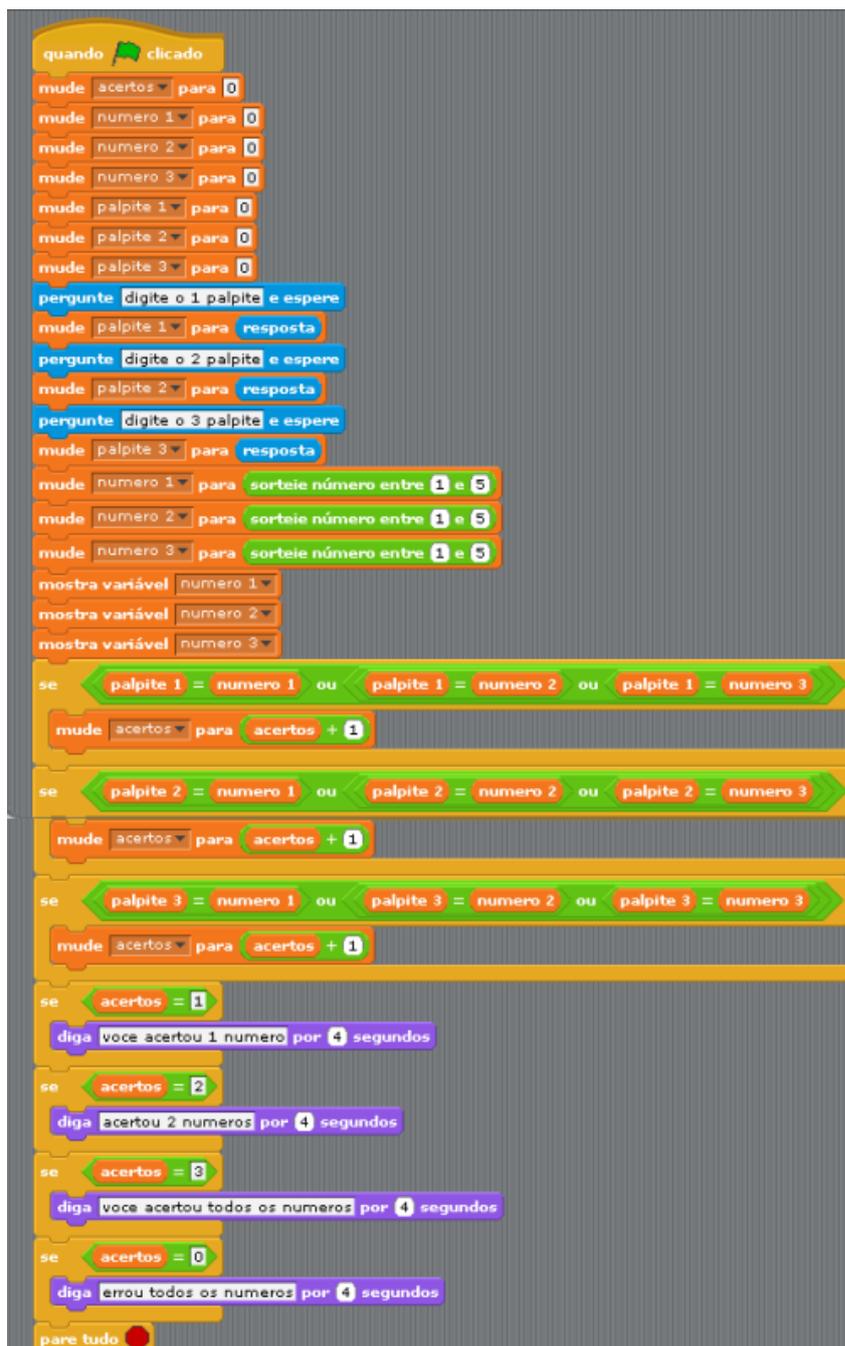
Descrição: Crie um algoritmo no qual o usuário insere 3 números inteiros entre 0 e 10. Em seguida, o programa sorteia 3 números aleatórios e verifica quantos números o usuário acertou.

Como fazer no Scratch:

1. Crie três variáveis: "Número 1", "Número 2" e "Número 3".
2. Utilize blocos de entrada para que o usuário insira os três números.
3. Crie três variáveis: "Sorteado 1", "Sorteado 2" e "Sorteado 3".
4. Use o bloco de geração aleatória para definir cada número entre 0 e 10.
5. Crie uma variável "Acertos" iniciada em zero.
6. Compare cada número digitado com os números sorteados e incremente a variável "Acertos" sempre que houver um número igual.
7. Exiba os números sorteados e quantos números o usuário acertou.

Resultado:

Figura 18 – Atividade 16



Fonte: (ANASTACIO, 2020)

4.3.17 Atividade 17: Identificação do Maior e Menor Número

Objetivo: Ensinar o uso de variáveis, operadores matemáticos e estruturas de repetição e condição no Scratch. Os alunos aprenderão a coletar vários valores inseridos pelo usuário, processar os dados para determinar o maior e o menor número e encerrar a entrada com um valor de parada.

Descrição: Crie um algoritmo que permita ao usuário inserir vários números inteiros. O pro-

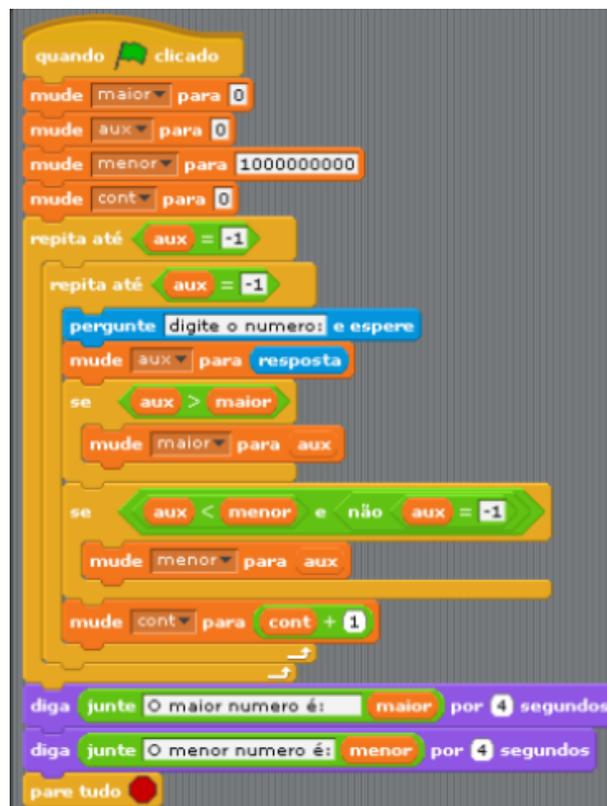
cesso continuará até que o usuário digite -1, momento em que o programa exibirá o maior e o menor número digitado.

Como fazer no Scratch:

1. Crie as variáveis "Número", "Maior" e "Menor".
2. Defina "Maior" como um valor muito pequeno (exemplo: -9999) e "Menor" como um valor muito grande (exemplo: 9999) para garantir que o primeiro número digitado seja considerado corretamente.
3. Solicite ao usuário que insira um número inteiro.
4. Enquanto o número não for -1, o programa deve continuar pedindo mais números.
5. Se for -1, a entrada de dados termina e os resultados são exibidos.
6. Se o número digitado for maior que "Maior", atualize a variável "Maior".
7. Se o número digitado for menor que "Menor", atualize a variável "Menor".
8. Após a entrada de -1, exiba os valores armazenados nas variáveis "Maior" e "Menor".

Resultado:

Figura 19 – Atividade 17



Fonte: (ANASTACIO, 2020)

4.3.18 Atividade 18: Cálculo da Média de 4 Notas e Verificação de Aprovação

Objetivo: Ensinar o uso de variáveis, operadores matemáticos e estruturas de repetição e condição no Scratch. Os alunos aprenderão a coletar notas inseridas pelo usuário, calcular a média e determinar se o aluno foi aprovado ou reprovado.

Descrição: Crie um algoritmo que solicite quatro notas do usuário, calcule a média e exiba se o aluno está aprovado ou reprovado com base na média mínima de 6.0.

Como fazer no Scratch:

1. Crie as variáveis "Nota", "Soma" e "Média".
2. Defina "Soma" e "Média" como 0 no início do programa.
3. Utilize um bloco de repetição para solicitar 4 notas do usuário.
4. A cada iteração, peça ao usuário para inserir uma nota e some esse valor à variável "Soma".
5. Após a repetição, calcule a média dividindo "Soma" por 4.
6. Se a média for maior ou igual a 6, exiba "Aprovado".
7. Caso contrário, exiba "Reprovado".
8. Após a exibição do resultado, redefina as variáveis para 0 para permitir novas execuções.

Resultado:

Figura 20 – Atividade 18



Fonte: (ANASTACIO, 2020)

5 Considerações Finais

O presente trabalho teve como objetivo principal desenvolver um caderno de atividades pedagógicas utilizando a plataforma Scratch, com o propósito de introduzir conceitos básicos de programação a crianças do 6º ao 9º ano do ensino fundamental. A proposta se destaca por aliar fundamentos teóricos e práticos, oferecendo aos alunos uma experiência de aprendizado lúdica e interativa, enquanto fornece aos professores e facilitadores um material estruturado e adaptável.

Por meio da elaboração do caderno, buscou-se explorar os benefícios do ensino de programação em contextos educacionais, destacando o Scratch como uma ferramenta acessível e eficaz para este propósito. As atividades foram cuidadosamente selecionadas e adaptadas, abrangendo tópicos que vão desde conceitos introdutórios, como sequência lógica, até elementos mais avançados, como variáveis e condicionais. Além disso, a alternância entre atividades guiadas e autônomas promove o desenvolvimento da autonomia dos alunos, incentivando-os a explorar e criar projetos próprios.

Os resultados esperados com a implementação deste caderno em projetos de extensão educacional incluem:

1. Ampliação do acesso ao ensino de programação para crianças 6º ao 9º ano do ensino fundamental, promovendo a inclusão digital e o empoderamento por meio da tecnologia;
2. Desenvolvimento de habilidades cognitivas, como pensamento lógico, criatividade e resolução de problemas, essenciais para a formação dos alunos em um mundo cada vez mais tecnológico;
3. Apoio aos professores e facilitadores, proporcionando-lhes um material prático e didático para aplicação em diferentes contextos educacionais.

Como continuidade deste trabalho, sugere-se a aplicação prática do caderno em projetos de extensão, possibilitando uma análise mais aprofundada dos impactos educacionais e sociais da metodologia proposta. A realização de oficinas e treinamentos para professores também pode ser um caminho para expandir o alcance e a eficácia do material desenvolvido.

Conclui-se, portanto, que este TCC contribui para o campo educacional ao disponibilizar um recurso pedagógico que une acessibilidade, criatividade e aprendizado prático. Com base nos autores referenciados neste trabalho, a utilização do Scratch no ensino de programação demonstrou ser uma abordagem eficiente e inclusiva, alinhada às demandas de uma sociedade em constante transformação tecnológica. Espera-se que o caderno de atividades inspire futuras iniciativas e colabore para a formação de uma geração mais preparada para os desafios do mundo digital.

Referências

- AMBRÓSIO, A. P.; COSTA, F. M. O uso de pbl para o ensino de algoritmos e programação de computadores. In: *Anais do 6º Congresso Internacional PBL*. São Paulo: USP/PAN-ABP, 2010. v. 1, p. 2–11. Citado nas páginas 17, 19 e 22.
- ANASTACIO, P. R. *O Uso do Scratch no Ensino de Programação*. 2020. Dissertação (Mestrado Profissional em Ensino) — Universidade Estadual do Norte do Paraná, 2020. Disponível em: <<https://uenp.edu.br/mestrado-ensino-publicacoes/ppgen-dissertacoes-defendidas/ppgen-dissertacoes-defendidas-3-turma-2018-2019/16449-paulo-roberto-anastacio/file>>. Citado nas páginas 19, 22, 23, 24, 26, 27, 30, 32, 34, 36, 38, 39, 40, 41, 42, 43, 44, 46, 47 e 48.
- ANJOS, S. M. dos et al. *Tecnologia na educação: uma jornada pela evolução histórica, desafios atuais e perspectivas futuras*. Quipá, 2024. ISBN 978-65-5376-296-1. Disponível em: <<https://educapes.capes.gov.br/handle/capes/742072>>. Citado na página 15.
- BRACKMANN, C. P. *Desenvolvimento do pensamento computacional através de atividades desplugadas na educação básica*. 2017. Tese (Tese de Doutorado) — Universidade Federal do Rio Grande do Sul, Porto Alegre, Brasil, 2017. Disponível em: <<http://hdl.handle.net/10183/172208>>. Citado na página 16.
- BRENNAN, K.; RESNICK, M. New frameworks for studying and assessing the development of computational thinking. In: *Proceedings of the 2012 annual meeting of the American Educational Research Association*. [S.l.: s.n.], 2012. v. 1, p. 1–25. Citado nas páginas 13, 14, 17, 22 e 23.
- CORRÊA, B. S. *Programando com Scratch no Ensino Fundamental: uma possibilidade para a construção de conceitos matemáticos*. 2021. Dissertação (Dissertação de Mestrado) — Universidade Federal do Rio Grande do Sul, 2021. Disponível em: <<http://hdl.handle.net/10183/222451>>. Citado nas páginas 18 e 23.
- COSTA, H. M. d. *O uso de programação com o Scratch na sala de aula de matemática no ensino básico: um estudo de atividades com possibilidades de utilização no Ensino Fundamental II*. 2022. Dissertação (Dissertação (Mestrado)) — Universidade Federal de Juiz de Fora (UFJF), ICE – Instituto de Ciências Exatas, Programa de Pós-graduação em Educação Matemática, Juiz de Fora, 2022. Disponível em: <<https://repositorio.ufjf.br/jspui/handle/ufjf/15262>>. Citado nas páginas 20 e 23.
- FARIAS, C. M. d.; OLIVEIRA, A. S. d.; SILVA, E. D. d. A. Uso do scratch na introdução de conceitos de lógica de programação: relato de experiência. In: *Anais do XXVI Workshop sobre Educação em Computação*. Porto Alegre, RS, Brasil: SBC, 2018. ISSN 2595-6175. Citado na página 14.
- FARIAS, C. M. de; OLIVEIRA, A. S. de; SILVA, E. D. de A. Uso do scratch na introdução de conceitos de lógica de programação: relato de experiência. In: *Anais do XXVI Workshop sobre Educação em Computação*. Porto Alegre, RS, Brasil: SBC, 2018. ISSN 2595-6175. Disponível em: <<https://sol.sbc.org.br/index.php/wei/article/view/3491>>. Citado na página 16.
- GIL, A. C. *Métodos e Técnicas de Pesquisa Social*. São Paulo: Atlas, 2008. Citado na página 21.

- GOMES, A. d. J. *Dificuldades de aprendizagem de programação de computadores: contributos para a sua compreensão e resolução*. 2010. Tese (Doutorado) — Universidade de Coimbra, Coimbra, 2010. Tese de Doutoramento em Engenharia Informática. Disponível em: <<http://hdl.handle.net/10316/14586>>. Citado na página 16.
- HOED, R. M. *Análise da evasão em cursos superiores: o caso da evasão em cursos superiores da área de computação*. dez 2016. Dissertação (Mestrado) — Universidade de Brasília, Brasília, dez 2016. Dissertação (Mestrado Profissional em Computação Aplicada). Disponível em: <<http://repositorio2.unb.br/jspui/handle/10482/22575>>. Citado na página 16.
- INEP. *Notas Estatísticas: Censo da Educação Superior 2020*. 2022. Acesso em: 23 ago. 2024. Disponível em: <https://download.inep.gov.br/publicacoes/institucionais/estatisticas_e_indicadores/notas_estatisticas_censo_da_educacao_superior_2020.pdf>. Citado na página 13.
- JUNIOR, J. d. S. L. *Uma análise das dificuldades de aprendizagem da Lógica de Programação no Curso Técnico Integrado em Informática do IFBA - Campus Jacobina*. Jacobina: [s.n.], 2016. Trabalho de Conclusão de Curso (Técnico Integrado em Informática). Disponível em: <<http://repositorio.ifba.edu.br/jspui/handle/123456789/131>>. Citado na página 15.
- KHOURI, C. M. B.; SANTOS, G. N. d.; BARBOSA, M. S. S. Mapeamento sistemático em metodologias de ensino-aprendizagem de programação. *Revista de Ciência da Computação*, v. 2, n. 1, p. 13–27, maio 2020. Disponível em: <<https://periodicos2.uesb.br/index.php/recic/article/view/6669>>. Citado na página 15.
- LAKATOS, E. M.; MARCONI, M. d. A. *Fundamentos de Metodologia Científica*. São Paulo: Atlas, 2003. Citado na página 21.
- LIMA, J. A. *Níveis de apropriação das TDIC pelos professores*. 2019. 146f p. Dissertação (Dissertação (Mestrado em Educação)) — Pontifícia Universidade Católica de São Paulo, São Paulo, 2019. Citado na página 13.
- MEDEIROS, A. C. S. de. *Scratch: da lógica de programação à química dos hidrocarbonetos*. 2021. Dissertação (Dissertação de Mestrado) — Universidade Estadual da Paraíba, 2021. Disponível em: <<http://hdl.handle.net/10183/222451>>. Citado nas páginas 18, 19 e 23.
- MIT Media Lab. *Guia do Educador Scratch: Crie uma História*. [S.l.], 2023. Acesso em: 17 fev. 2025. Disponível em: <<https://resources.scratch.mit.edu/www/guides/pt-br/StoryGuide.pdf>>. Citado na página 26.
- PAPERT, S. *Mindstorms: Children, Computers, and Powerful Ideas*. New York: Basic Books, 1980. Citado na página 17.
- RESNICK, M. et al. Scratch: programming for all. *Commun. ACM*, Association for Computing Machinery, New York, NY, USA, v. 52, n. 11, p. 60–67, nov 2009. ISSN 0001-0782. Disponível em: <<https://doi.org/10.1145/1592761.1592779>>. Citado nas páginas 13, 14 e 22.
- REZENDE, C. M. C.; BISPO, E. L. Comparação entre o uso de pseudocódigo e a programação visual no ensino de programação: Uma avaliação a partir da ferramenta scratch. In: *Conferência Ibérica de Sistemas e Tecnologias de Informação (CISTI)*. Caceres: CISTI, 2018. p. 1–5. Citado nas páginas 16, 19 e 22.
- SALAZAR, R.; ODAKURA, V.; BARVINSKI, C. Scratch no ensino superior: motivação. In: CBIE-LACLO. *Anais do XXVI Simpósio Brasileiro de Informática na Educação (SBIE 2015)*. Dourados, MS, Brasil, 2015. Citado nas páginas 17 e 19.

SCRATCH. *SCRATCH. Imagine, program, share.* 2017. Acesso em: 11 Jun 2019. Disponível em: <<http://scratch.mit.edu>>. Citado na página 16.

SILVA, J. C. da. Ensino de programação para alunos do ensino básico: Um levantamento das pesquisas realizadas no Brasil. In: UNIVERSIDADE FEDERAL DA PARAÍBA (UFPB). *Anais de Conferência*. Rio Tinto, PB, Brasil, 2024. Citado na página 21.

SILVA, M. L. A. da. *A ferramenta Scratch: uma proposta lúdica de ensino para aprendizagem de estudantes com transtorno do espectro autista.* 2022. Dissertação (Dissertação de Mestrado) — Universidade Pitágoras Unopar Anhangüera, Londrina, 2022. Orientadora: Prof.a Dr.a Luciane Guimarães Batistella Bianchini. Citado na página 18.

SOUZA, M. F. d. *Ensino e Tecnologias: aprendizagem significativa por meio da plataforma Scratch.* 2019. 100 p. Dissertação (Dissertação (Mestrado)) — Colégio Pedro II, Pró-Reitoria de Pós-Graduação, Pesquisa, Extensão e Cultura, Programa de Mestrado Profissional em Práticas de Educação Básica, Rio de Janeiro, 2019. Citado nas páginas 17, 20, 23, 26, 29, 31, 33, 35 e 37.

ZANETTI, H. A. P. et al. *Proposta de ensino de programação para crianças com scratch e pensamento computacional.* 2021. Citado nas páginas 18 e 22.

Apêndices

APÊNDICE A – Caderno de Atividades

Caderno de atividades Scratch

O Scratch é uma linguagem de programação visual desenvolvida pelo MIT Media Lab, projetada para introduzir conceitos de programação de forma simples, lúdica e acessível. Utilizando uma interface baseada em blocos coloridos que se encaixam, o Scratch permite criar animações, jogos e histórias interativas sem a necessidade de conhecimentos prévios em programação. É amplamente utilizado em ambientes educacionais por sua abordagem intuitiva e por estimular o pensamento lógico, a criatividade e a resolução de problemas (REZENDE; BISPO, 2018).

Este caderno de atividades foi elaborado com o objetivo de servir como um guia prático e dinâmico para introduzir conceitos básicos de programação para crianças do 6º ao 9º ano do ensino fundamental. Ele não é apenas uma coleção de exercícios, mas um recurso que propõe uma jornada de aprendizado interativa e envolvente, onde a exploração, a curiosidade e a diversão são pilares fundamentais. Através deste material, os alunos poderão:

- Aprender os conceitos essenciais de programação de maneira gradual.
- Desenvolver habilidades como lógica, resolução de problemas e criatividade.
- Aplicar conhecimentos em atividades práticas, que conectam teoria e prática.

Instruções para Facilitadores: Guia Prático para Usar o Scratch

Este guia destina-se a **facilitadores** (professores ou monitores) que irão utilizar o Scratch como ferramenta de ensino. Abaixo, detalhamos como começar a trabalhar com a plataforma, desde a criação de uma conta até o uso básico, para que você esteja preparado para apoiar seus alunos.

1. Criando uma Conta no Scratch

1. Acesse a plataforma:

- Entre no site oficial do Scratch <https://scratch.mit.edu>.

2. Clique em "Junte-se ao Scratch":

- O botão está localizado no canto superior direito da página inicial.

3. Escolha um nome de usuário e senha:

- Oriente os alunos a escolherem um nome simples e seguro, evitando informações pessoais.

4. Forneça um e-mail válido:

- O e-mail é necessário para ativar a conta e recuperar informações, se necessário.

5. Complete o cadastro:

- Siga as etapas solicitadas, como idade e país, e finalize clicando no link de ativação enviado ao e-mail cadastrado.

Dica: Se estiver trabalhando com alunos menores de 13 anos, considere criar uma conta geral para a turma ou utilizar contas de facilitadores para gerenciar os projetos.

2. Explorando a Interface do Scratch

Após fazer login, o facilitador pode explorar os seguintes recursos principais:

1. Página Inicial:

- Navegue entre projetos compartilhados pela comunidade.
- Explore os tutoriais disponíveis na aba "Ideias".

2. Editor de Projetos:

- Clique em "Criar" para abrir o editor onde os alunos desenvolverão seus projetos.
 - O editor está dividido em três seções principais:
 - **Palco:** Área onde o projeto é exibido em tempo real.
 - **Blocos de Código:** Divididos em categorias como movimento, aparência, som, controle e eventos.
 - **Área de Código:** Espaço para arrastar e organizar os blocos.
-

3. Primeiros Passos no Scratch

1. Movendo o Personagem (Sprite):

- Por padrão, o Scratch adiciona um personagem chamado "Gato". Oriente os alunos a clicar no bloco azul "Mover 10 passos" e soltá-lo na área de código.
- Clique no bloco para ver o sprite se mover no palco.

2. Alterando o Sprite:

- Clique no ícone de "Escolher Sprite" (localizado no canto inferior direito) para selecionar ou desenhar um novo personagem.

3. Criando uma Animação Simples:

- Adicione blocos de "Movimento" e "Aparência" para fazer o sprite andar e mudar de cor. Por exemplo:
 - "Mover 10 passos".
 - "Esperar 1 segundo".
 - "Mudar para a fantasia seguinte".

4. Adicionando um Fundo:

- Clique em "Escolher um Plano de Fundo" e selecione um cenário para o projeto.

5. Usando Eventos:

- Para iniciar uma ação com um clique, adicione o bloco amarelo "Quando a bandeira verde for clicada".

4. Recursos para Ensinar o Básico

● Tutoriais Integrados:

- No menu "Ideias", explore vídeos e guias passo a passo sobre como criar jogos, histórias interativas e animações.

● Desafios Iniciais:

- Oriente os alunos a fazer um projeto simples, como mover o personagem para frente e para trás ou mudar de cor ao pressionar uma tecla.

5. Salvando e Compartilhando Projetos

1. Salvando:

- Certifique-se de que os alunos estão conectados para salvar seus projetos automaticamente.

2. Compartilhando:

- Clique em “Compartilhar” no topo da tela do projeto para torná-lo público e permitir que outros usuários o vejam e remixem.
-

6. Boas Práticas no Scratch

● Estimule a Exploração:

- Incentive os alunos a experimentar diferentes blocos e descobrir novos efeitos.

● Destaque o Processo:

- Valorize o esforço e a criatividade, não apenas o produto final.

● Promova a Colaboração:

- Alunos podem remixar projetos uns dos outros para aprender em conjunto.

Planejamento para Aula

Aula e Atividade	Conceitos Teóricos	Tempo em Minutos	Modalidade
1. Movimentação Básica	Sequência Lógica, Laços	30-40	Com Professor
2. Animar seu Nome	Sequência Lógica, Laços	40-50	Com professor
3. Desenhando Círculo	Laços , Sequência Lógica	40-50	Aluno Sozinho
4. Veloz e Furioso	Condicionais, Variáveis	50-60	Com Professor
5. Desenhando Quadrado	Laços, Sequência Lógica	40-50	Aluno Sozinho
6. Astronauta Artista	Laços, Condicionais	50-60	Com Professor
7. Jogo de Perguntas	Condicionais, Validação	30-40	Aluno Sozinho
8. Desafio de Lógica	Variáveis, Condicionais	50-60	Com Professor
9. Plano Cartesiano	Coordenadas, Laços	40-50	Aluno Sozinho
10. Calculando Média	Variáveis, Operações	30-40	Com Professor
11. Operações Matemáticas	Condicionais, Operadores	40-50	Aluno Sozinho
12. Par ou Ímpar	Condicionais, Operadores	30-40	Com Professor
13. Identificando o Maior Número	Condicionais	50-60	Aluno Sozinho
14. : Calculando a Média	Variáveis, Condicionais	60-70	Com Professor
15.: Calculando a Média Avançado	Variáveis, Condicionais	40-50	Aluno Sozinho
16. : Sorteio e Comparação de Números	Variáveis, Condicionais	50-60	Com Professor
17. Identificação do Maior e Menor Número	Condicionais, Laços	50-60	Aluno Sozinho
18. Cálculo da Média de 4 Notas	Variáveis, Laços	60-70	Com Professor

Atividade 1 - Movimentação Básica

Objetivo: Ensinar aos alunos os conceitos básicos de programação no Scratch, introduzindo-os à interface da plataforma e ao uso de blocos de movimentação, laços

de repetição e interação com o palco. O foco está em desenvolver um entendimento inicial sobre como criar movimento contínuo e como utilizar bordas como elementos interativos.

Descrição: Desenvolva um programa que faça o personagem mover 10 passos, esperar um segundo e trocar de traje. Caso toque na borda, ele deve virar e continuar andando.

1. Arraste o bloco “mova 10 passos” para o palco de programação.
2. Adicione o bloco “espere 1 segundo”.
3. Use o bloco “se tocar na borda, vire” para corrigir a direção do personagem.
4. Encaixe os blocos dentro de um laço “para sempre”.



Fonte: (ANASTACIO, 2020)

Atividade 2 - Animar seu Nome

Objetivo: Introduzir a interface do Scratch, familiarizar o aluno com a área de programação e ensinar o uso básico de blocos de movimento, aparência e controle para criar uma animação personalizada.

Como fazer no Scratch:

1. Insira as letras do seu nome como sprites no palco.
2. Para cada letra, programe blocos para movê-la, alterar sua cor e tamanho em intervalos.
3. Adicione um bloco de repetição para que as ações sejam contínuas.
4. Execute o programa clicando na bandeira verde e observe a animação do seu nome.

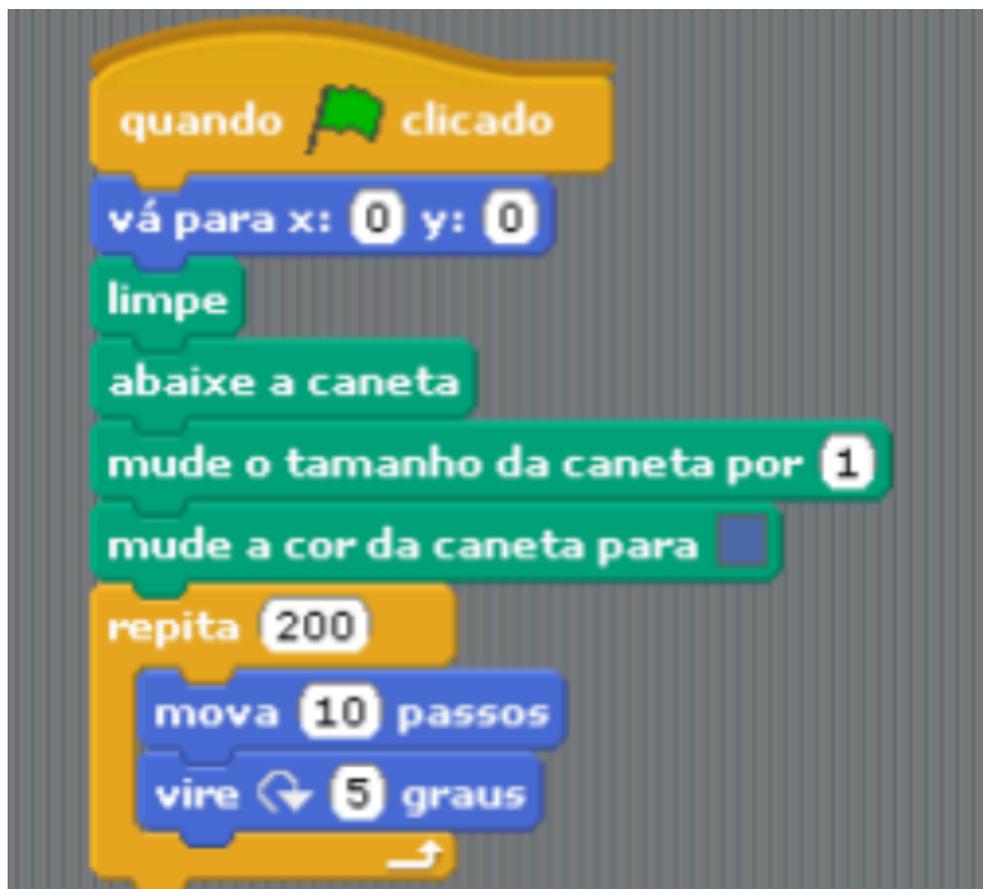


Fonte: (SOUZA, 2019)

Objetivo: Explorar conceitos matemáticos básicos, como ângulos e repetições, enquanto os alunos aprendem a utilizar laços e ferramentas de desenho no Scratch para criar formas geométricas simples, promovendo o raciocínio lógico e a relação entre código e geometria.

Descrição: Desenvolva um programa que mova o personagem e desene um círculo.

1. Use o bloco de movimento para avançar 10 passos e girar 5 graus.
2. Ative a ferramenta “caneta” para que o personagem desene no palco.
3. Repita o processo até completar um círculo.



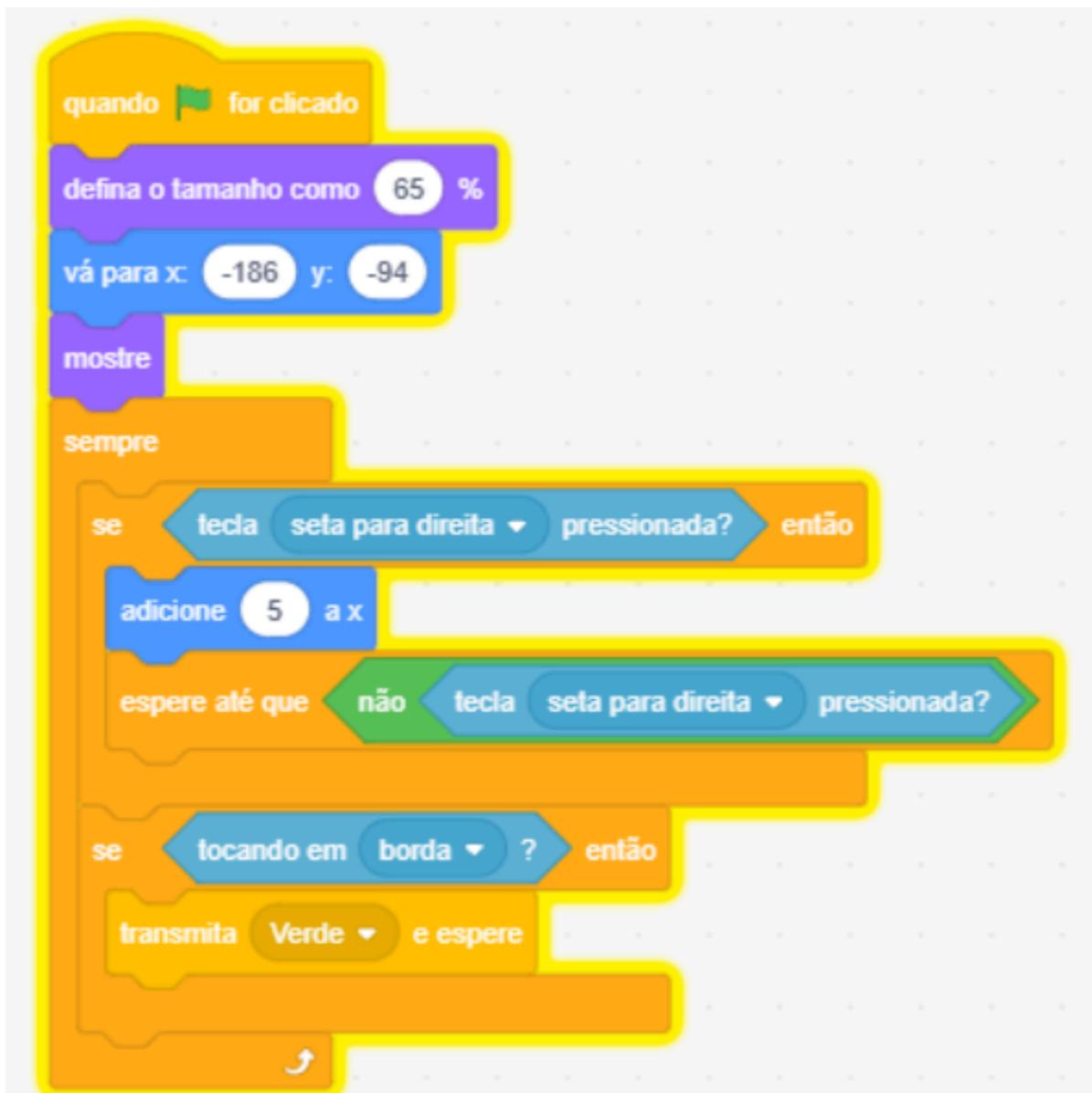
Fonte: (ANASTACIO, 2020)

Atividade 4 - Veloz e Furioso

Objetivo: Introduzir a programação de jogos simples no Scratch, com foco no uso de eventos e controle de movimentação.

Como fazer no Scratch:

1. Adicione dois carros como sprites e configure um cenário de corrida.
2. Programe cada carro para se mover utilizando teclas específicas (ex.: seta para direita e barra de espaço).
3. Utilize blocos condicionais para determinar o vencedor ao alcançar o final da pista.
4. Exiba uma mensagem de vitória para o carro que chegar primeiro.



Fonte: (SOUZA, 2019)

Atividade 5 - Desenhando um Quadrado

Objetivo: Aproximar os alunos da lógica por trás da programação estruturada, ensinando a criar formas geométricas usando ângulos e laços de repetição. Os alunos aprendem a planejar sequências de passos e aplicar noções de geometria, como ângulos retos.

Descrição: Desenvolva um programa que desenhe um quadrado ao ser executado.

1. Adicione um laço de repetição “repita 4 vezes”.
2. Programe o personagem para se mover 100 passos e girar 90 graus após cada movimento.
3. Utilize a ferramenta “caneta” para desenhar as linhas.



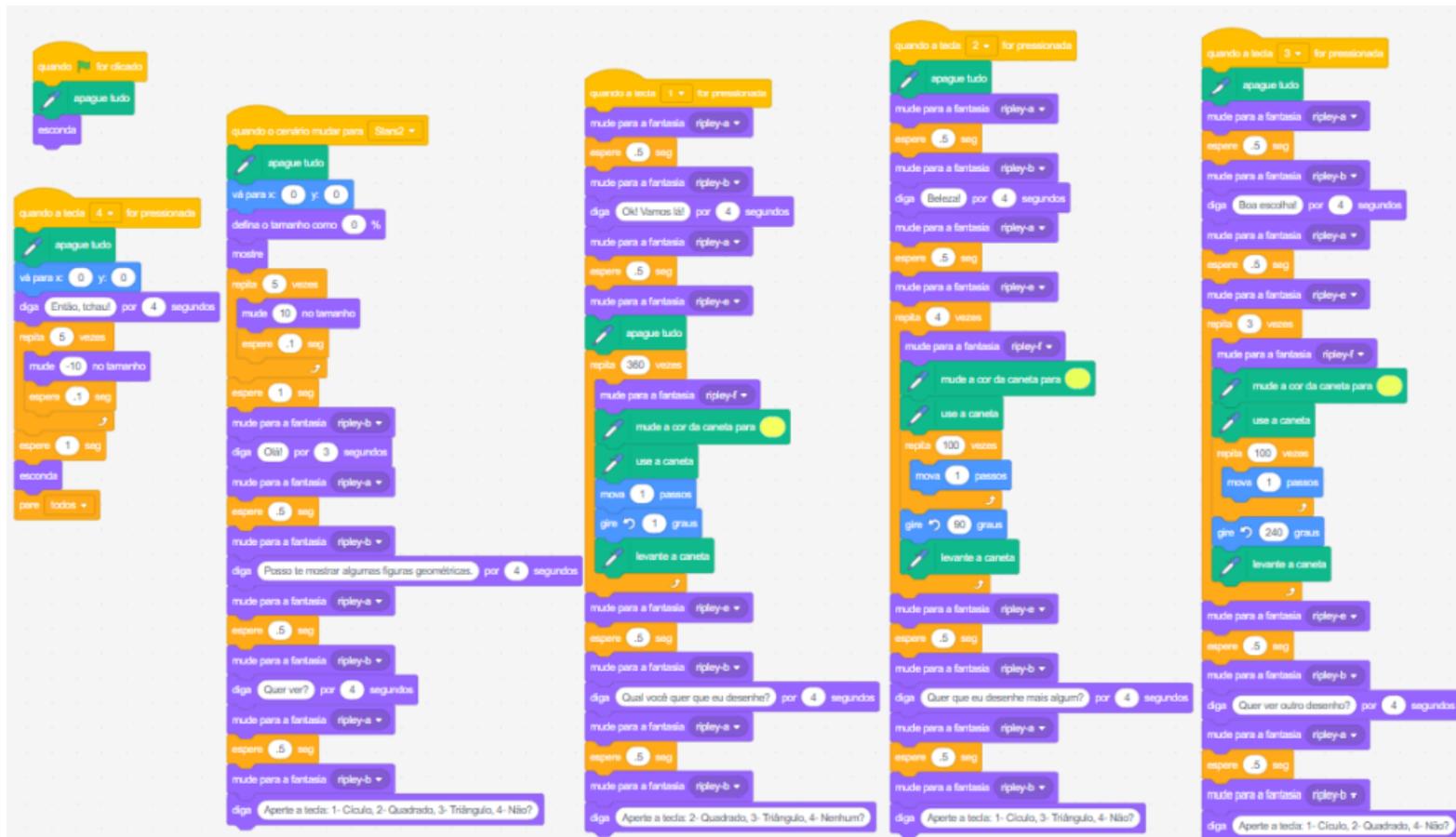
Fonte: (ANASTACIO, 2020)

Atividade 6 - Astronauta Artista

Objetivo: Ensinar o uso de extensões no Scratch (como a ferramenta “Caneta”) para criar desenhos geométricos, estimulando a lógica matemática e a criatividade.

Como fazer no Scratch:

1. Adicione o sprite de um astronauta e configure um cenário espacial.
2. Habilite a extensão “Caneta” no Scratch.
3. Programe o astronauta para desenhar formas geométricas básicas (círculos, triângulos, quadrados) ao pressionar diferentes teclas.
4. Adicione efeitos visuais e de som para enriquecer o projeto.



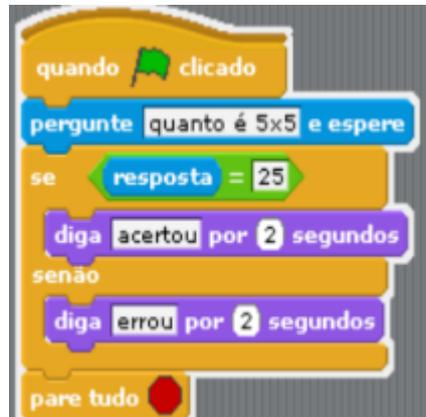
Fonte: (SOUZA, 2019)

Atividade 7 - Jogo de Perguntas

Objetivo: Introduzir o conceito de entrada e validação de dados pelo usuário, utilizando blocos de "perguntar" e "se... então". Os alunos desenvolvem habilidades para criar interatividade em projetos e entender lógica condicional básica.

Descrição: Desenvolva um programa que pergunte “Quanto é 5 x 5?” ao usuário. Analise a resposta e informe se está correta.

1. Use o bloco “perguntar” para coletar a resposta.
2. Adicione um bloco condicional “se... então” para comparar a resposta com o valor esperado.
3. Exiba uma mensagem indicando se a resposta está certa ou errada.



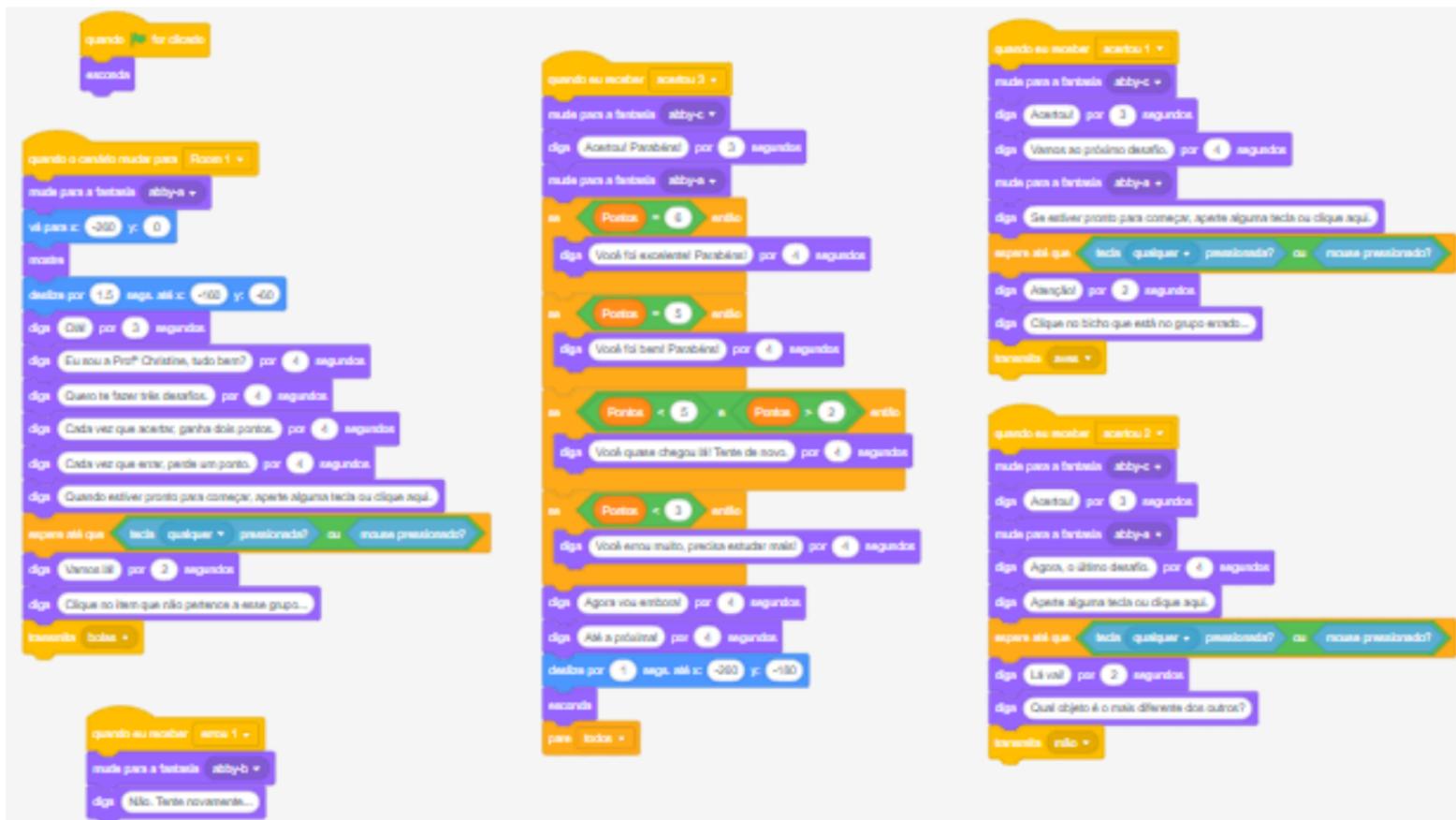
Fonte: (ANASTACIO, 2020)

Atividade 8 - Desafio de Lógica

Objetivo: Trabalhar o raciocínio lógico e ensinar o uso de variáveis no Scratch para criar um quiz interativo com feedback automático.

Como fazer no Scratch:

1. Crie um cenário inicial e adicione um sprite de professor(a).
2. Programe perguntas com opções de resposta como sprites clicáveis.
3. Utilize variáveis para registrar o nome do aluno e a pontuação.
4. Programe blocos condicionais para exibir mensagens de feedback com base na pontuação final.



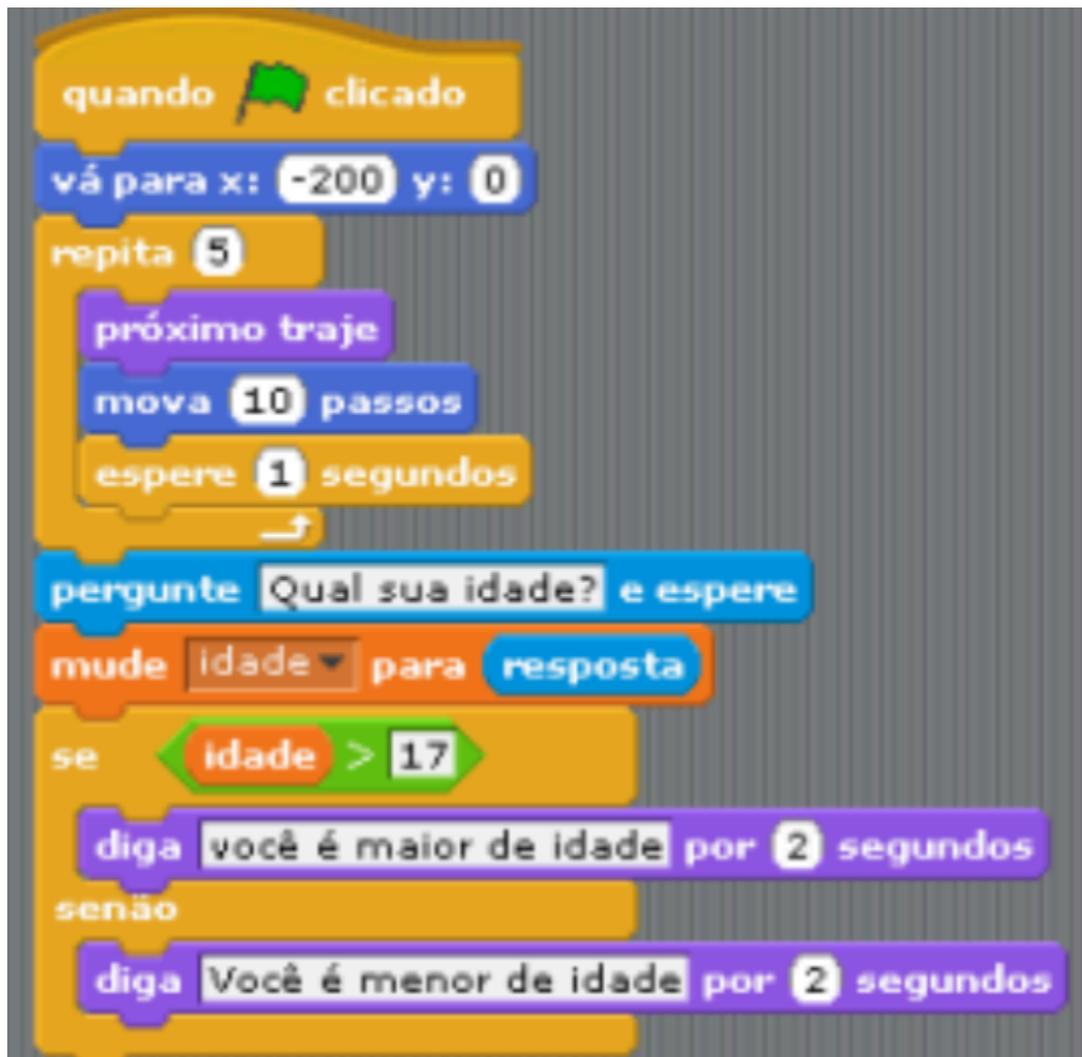
Fonte: (SOUZA, 2019)

Atividade 9 - Movimentos no Plano Cartesiano

Objetivo: Ensinar o uso do plano cartesiano no Scratch para posicionamento de personagens. Os alunos desenvolvem habilidades para movimentar sprites usando coordenadas e integram laços de repetição e condicionais para criar padrões e interação.

Descrição: Crie um programa onde o personagem se move até a posição $x=-200, y=0$, troca de traje e repete esse movimento 5 vezes.

1. Use blocos de movimento para alterar a posição do personagem.
2. Adicione um laço para repetir o movimento 5 vezes.
3. Inclua um bloco de interação onde o personagem pergunta a idade do usuário e exibe uma mensagem baseada na resposta.



Fonte: (ANASTACIO, 2020)

Atividade 10 - Calculando a Média

Objetivo: Introduzir o conceito de variáveis e operações matemáticas no Scratch. Os alunos aprendem a armazenar, manipular e exibir dados inseridos pelo usuário, consolidando a lógica por trás do uso de variáveis.

Descrição: Desenvolva um programa que calcula a média de duas notas inseridas pelo usuário e exibe o resultado.

1. Crie duas variáveis para armazenar as notas.
2. Use o bloco “perguntar” para coletar as notas do usuário.

3. Calcule a média e exiba o resultado.



Fonte: (ANASTACIO, 2020)

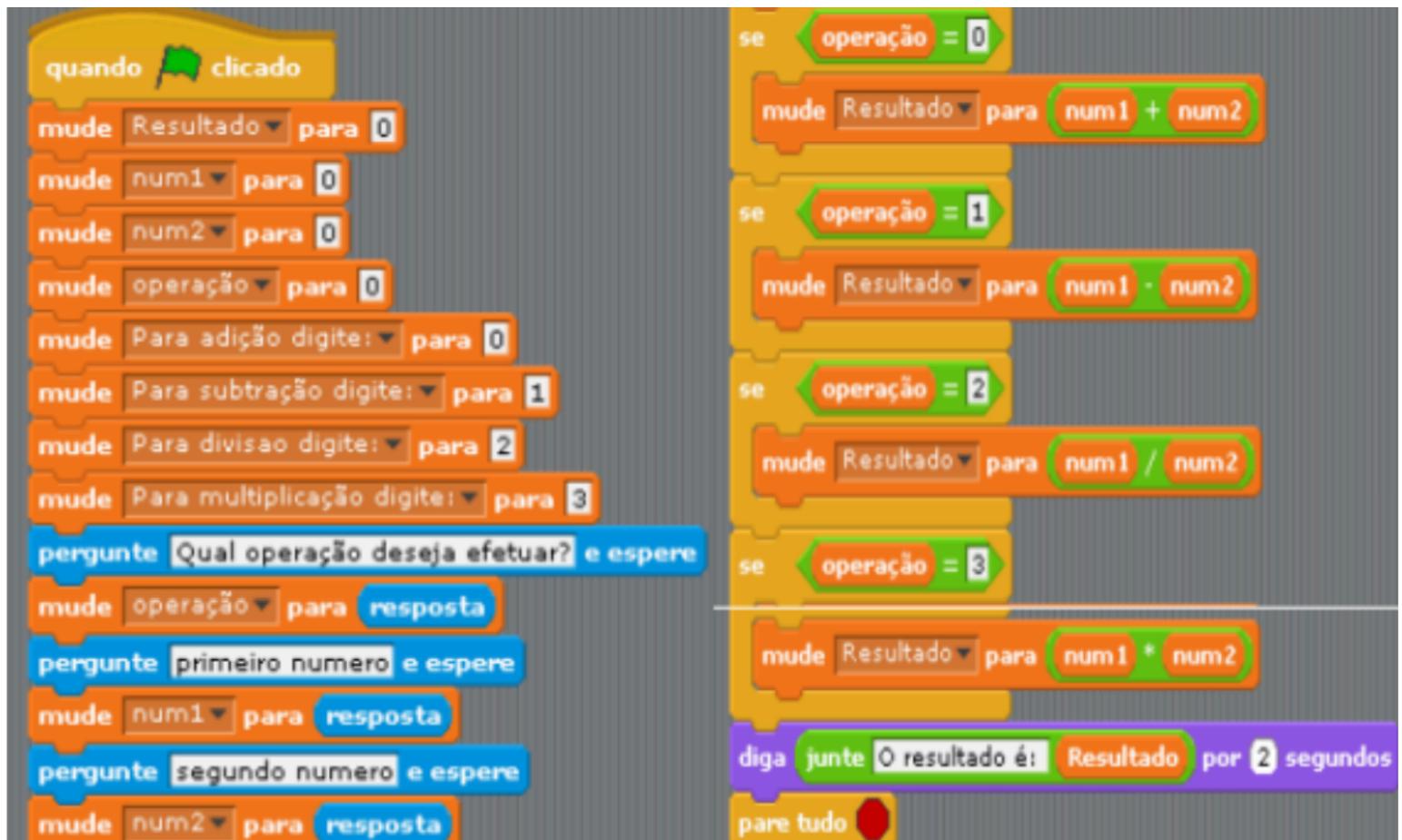
Atividade 11 - Operações Matemáticas

Objetivo: Ensinar como utilizar blocos condicionais para realizar cálculos dinâmicos com base na escolha do usuário. Os alunos aprendem a planejar diferentes fluxos dentro de um programa, consolidando lógica condicional e o uso de operadores matemáticos.

Descrição: Desenvolva um algoritmo onde o usuário escolhe uma operação (adição, subtração, multiplicação ou divisão) e insere dois valores. O programa realiza o cálculo e exibe o resultado.

1. Use o bloco “perguntar” para coletar a escolha da operação e os valores.

2. Adicione condicionais para realizar diferentes cálculos dependendo da operação escolhida.
3. Exiba o resultado final em um balão de fala.



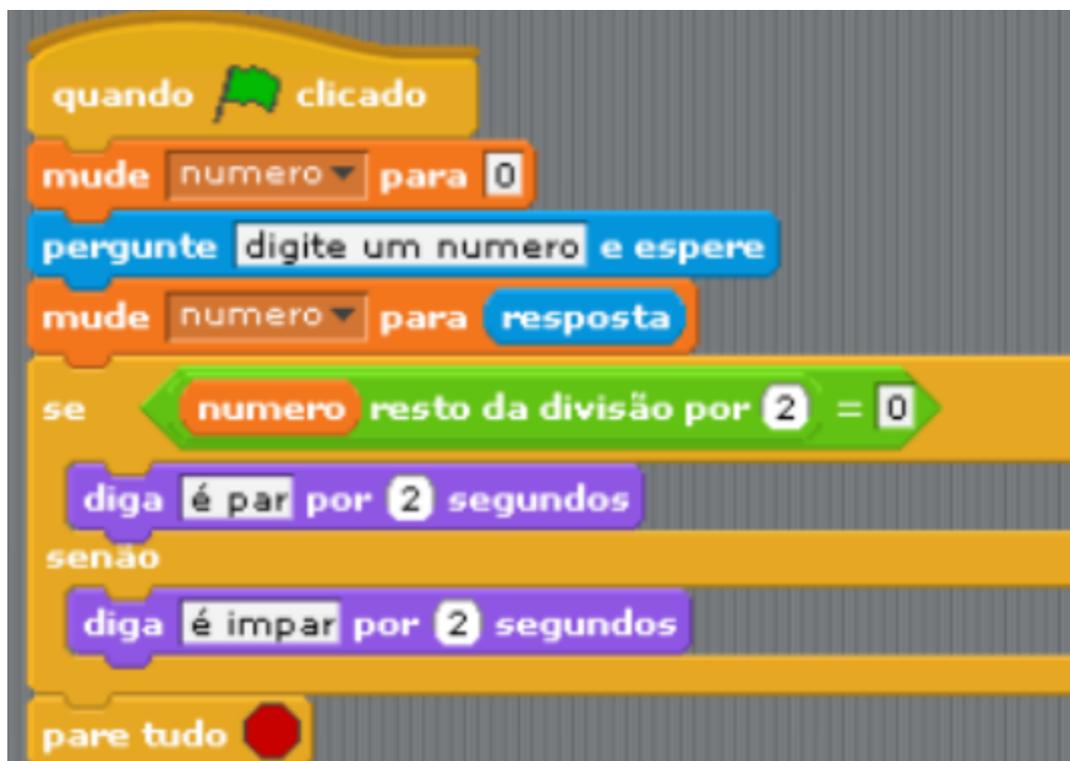
Fonte: (ANASTACIO, 2020)

Atividade 12 - Par ou Ímpar

Objetivo: Trabalhar com conceitos básicos de divisibilidade e lógica condicional. Os alunos aprendem a usar operadores matemáticos e blocos de decisão para criar programas que validam números inseridos pelo usuário.

Descrição: Desenvolva um programa que verifica se um número inserido pelo usuário é par ou ímpar.

1. Use o bloco “perguntar” para receber o número do usuário.
2. Adicione um bloco condicional para verificar se o número é divisível por 2.
3. Exiba a mensagem “par” ou “ímpar” de acordo com a validação.



Fonte: (ANASTACIO, 2020)

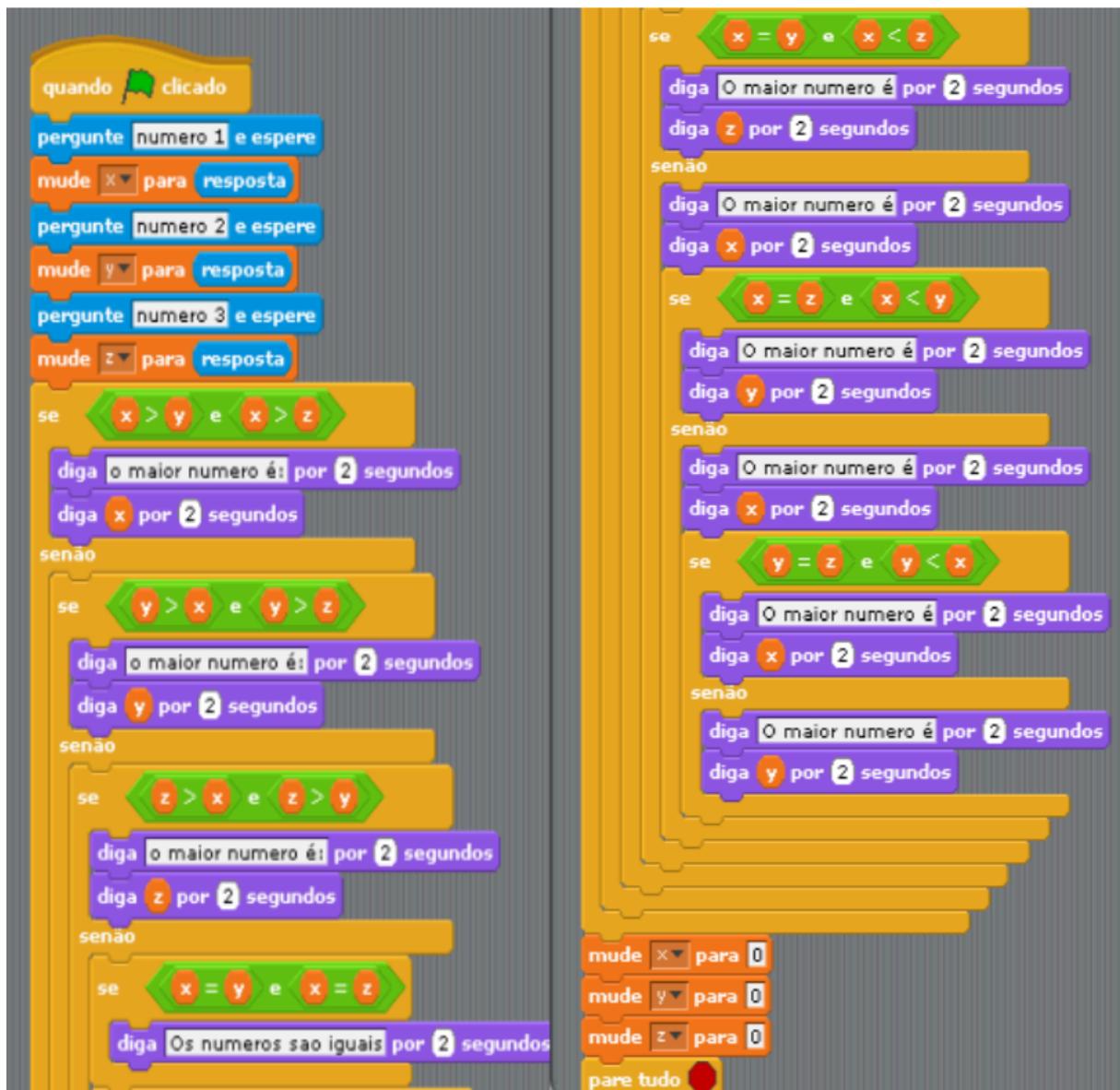
Atividade 13 - Identificando o Maior Número

Objetivo: Ensinar o uso de variáveis, operadores lógicos e estruturas condicionais no Scratch. Os alunos aprenderão a solicitar entradas do usuário, comparar valores e exibir o maior número inserido.

Descrição: Crie um algoritmo que permita ao usuário inserir três números e, em seguida, exiba o maior entre eles.

1. Crie três variáveis: "Número 1", "Número 2" e "Número 3"
2. Utilize blocos de entrada para que o usuário informe os três números.
3. Utilize estruturas condicionais para determinar qual dos três números é o maior.

4. Se "Número 1"for maior que "Número 2"e "Número 3", exiba "O maior número é: Número 1"
5. Se "Número 2"for maior que "Número 1"e "Número 3", exiba "O maior número é: Número 2"
6. Caso contrário, exiba "O maior número é: Número 3"
7. Se houver números iguais, exiba uma mensagem indicando isso.
8. Utilize blocos de aparência para exibir a resposta ao usuário.



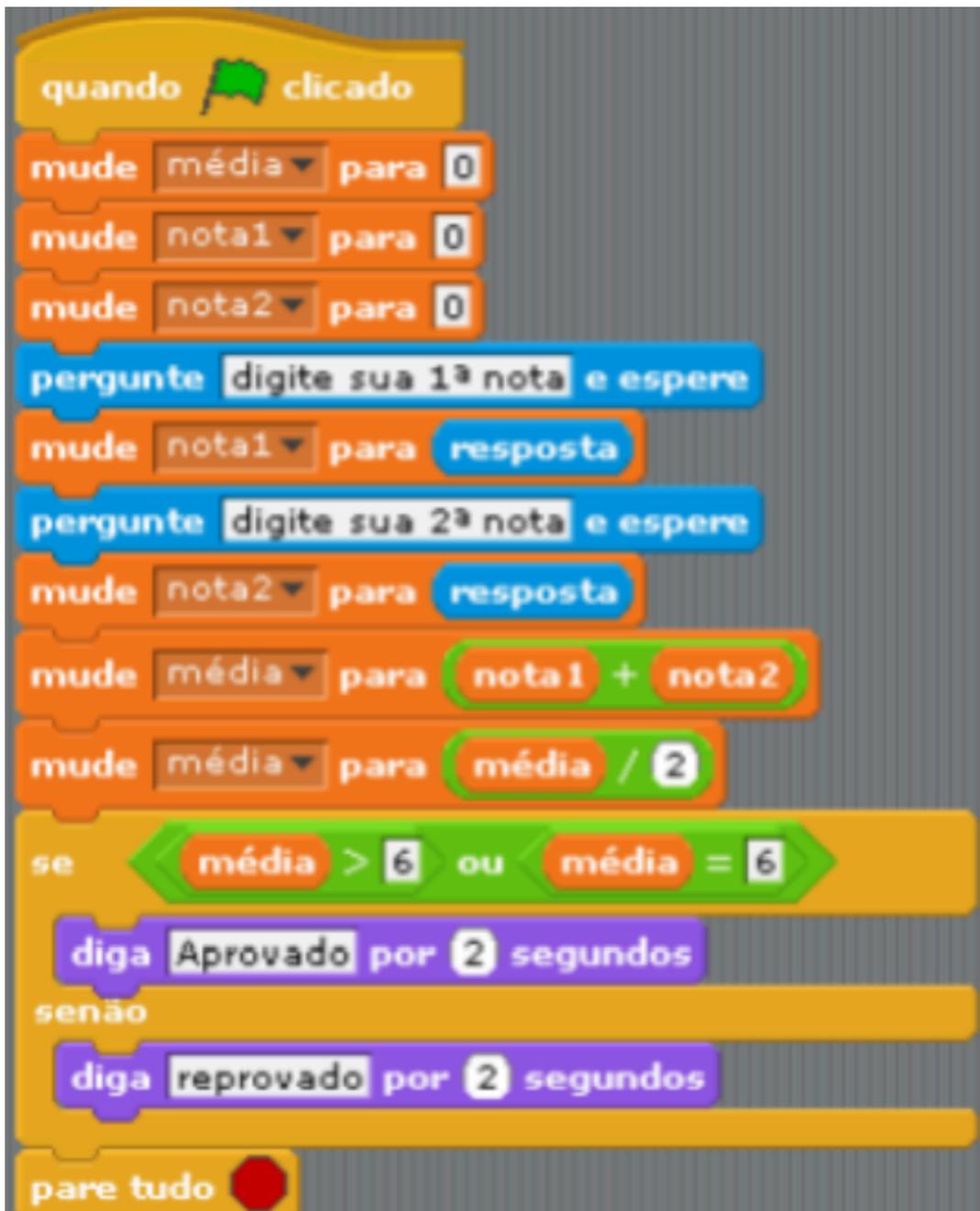
Fonte: (ANASTACIO, 2020)

Atividade 14 - Calculando a Média e Verificando Aprovação

Objetivo: Ensinar o uso de variáveis, operadores matemáticos e estruturas condicionais no Scratch. Os alunos aprenderão a coletar valores inseridos pelo usuário, calcular médias e determinar a situação de um aluno com base em regras de aprovação.

Descrição: Crie um algoritmo que solicite três notas do usuário, calcule a média e determine a situação do aluno com base nas regras de aprovação.

1. Crie duas variáveis: "Nota 1" e "Nota 2".
2. Utilize blocos de entrada para que o usuário insira os valores das notas.
3. Crie a variável "Média" e programe o cálculo: $Média = (Nota\ 1 + Nota\ 2) / 2$
4. Se a média for maior ou igual a 6, exiba "Aprovado"
5. Caso contrário, exiba "Reprovado"



Fonte: (ANASTACIO, 2020)

Atividade 15 - Calculando a Média e Verificando Aprovação com Recuperação

Objetivo: Ensinar o uso de variáveis, operadores matemáticos e estruturas condicionais no Scratch. Os alunos aprenderão a coletar valores inseridos pelo usuário, calcular médias e determinar a situação de um aluno com base em regras de aprovação.

Descrição: Crie um algoritmo que solicite três notas do usuário, calcule a média e determine a situação do aluno com base nas regras de aprovação.

1. Crie três variáveis: "Nota 1", "Nota 2" e "Nota 3".
2. Utilize blocos de entrada para que o usuário insira os três valores.
3. Crie a variável "Média" e programe o cálculo: $Média = (Nota\ 1 + Nota\ 2 + Nota\ 3) / 3$
4. Se a média for maior ou igual a 7, exiba "Aprovado"
5. Se a média for maior ou igual a 4 e menor que 7, exiba "Em Exame" e permita que o usuário insira uma nova nota do exame.
6. Se a média for menor que 4, exiba "Reprovado".
7. Se o aluno precisar de exame, crie a variável "Nota do Exame" e solicite um novo valor.
8. Calcule a nova média: $Nova\ Média = (Média + Nota\ do\ Exame) / 2$
9. Se a nova média for maior ou igual a 5, exiba "Aprovado com Exame".
10. Caso contrário, exiba "Reprovado".



Fonte: (ANASTACIO, 2020)

Atividade 16 - Sorteio e Comparação de Números

Objetivo: Ensinar o uso de variáveis, operadores matemáticos, geração de números aleatórios e estruturas condicionais no Scratch. Os alunos aprenderão a coletar

valores inseridos pelo usuário, gerar números aleatórios e comparar os valores para determinar quantos foram acertados

Descrição: Crie um algoritmo no qual o usuário insere 3 números inteiros entre 0 e 10. Em seguida, o programa sorteia 3 números aleatórios e verifica quantos números o usuário acertou.

1. Crie três variáveis: "Número 1", "Número 2" e "Número 3".
2. Utilize blocos de entrada para que o usuário insira os três números.
3. Crie três variáveis: "Sorteado 1", "Sorteado 2" e "Sorteado 3".
4. Use o bloco de geração aleatória para definir cada número entre 0 e 10.
5. Crie uma variável "Acertos" iniciada em zero.
6. Compare cada número digitado com os números sorteados e incremente a variável "Acertos" sempre que houver um número igual.
7. Exiba os números sorteados e quantos números o usuário acertou.

```
quando clicado
  mude acertos para 0
  mude numero 1 para 0
  mude numero 2 para 0
  mude numero 3 para 0
  mude palpite 1 para 0
  mude palpite 2 para 0
  mude palpite 3 para 0
  pergunte digite o 1 palpite e espere
  mude palpite 1 para resposta
  pergunte digite o 2 palpite e espere
  mude palpite 2 para resposta
  pergunte digite o 3 palpite e espere
  mude palpite 3 para resposta
  mude numero 1 para sorteie número entre 1 e 5
  mude numero 2 para sorteie número entre 1 e 5
  mude numero 3 para sorteie número entre 1 e 5
  mostra variável numero 1
  mostra variável numero 2
  mostra variável numero 3
  se palpite 1 = numero 1 ou palpite 1 = numero 2 ou palpite 1 = numero 3
    mude acertos para acertos + 1
  se palpite 2 = numero 1 ou palpite 2 = numero 2 ou palpite 2 = numero 3
    mude acertos para acertos + 1
  se palpite 3 = numero 1 ou palpite 3 = numero 2 ou palpite 3 = numero 3
    mude acertos para acertos + 1
  se acertos = 1
    diga voce acertou 1 numero por 4 segundos
  se acertos = 2
    diga acertou 2 numeros por 4 segundos
  se acertos = 3
    diga voce acertou todos os numeros por 4 segundos
  se acertos = 0
    diga errou todos os numeros por 4 segundos
  pare tudo
```

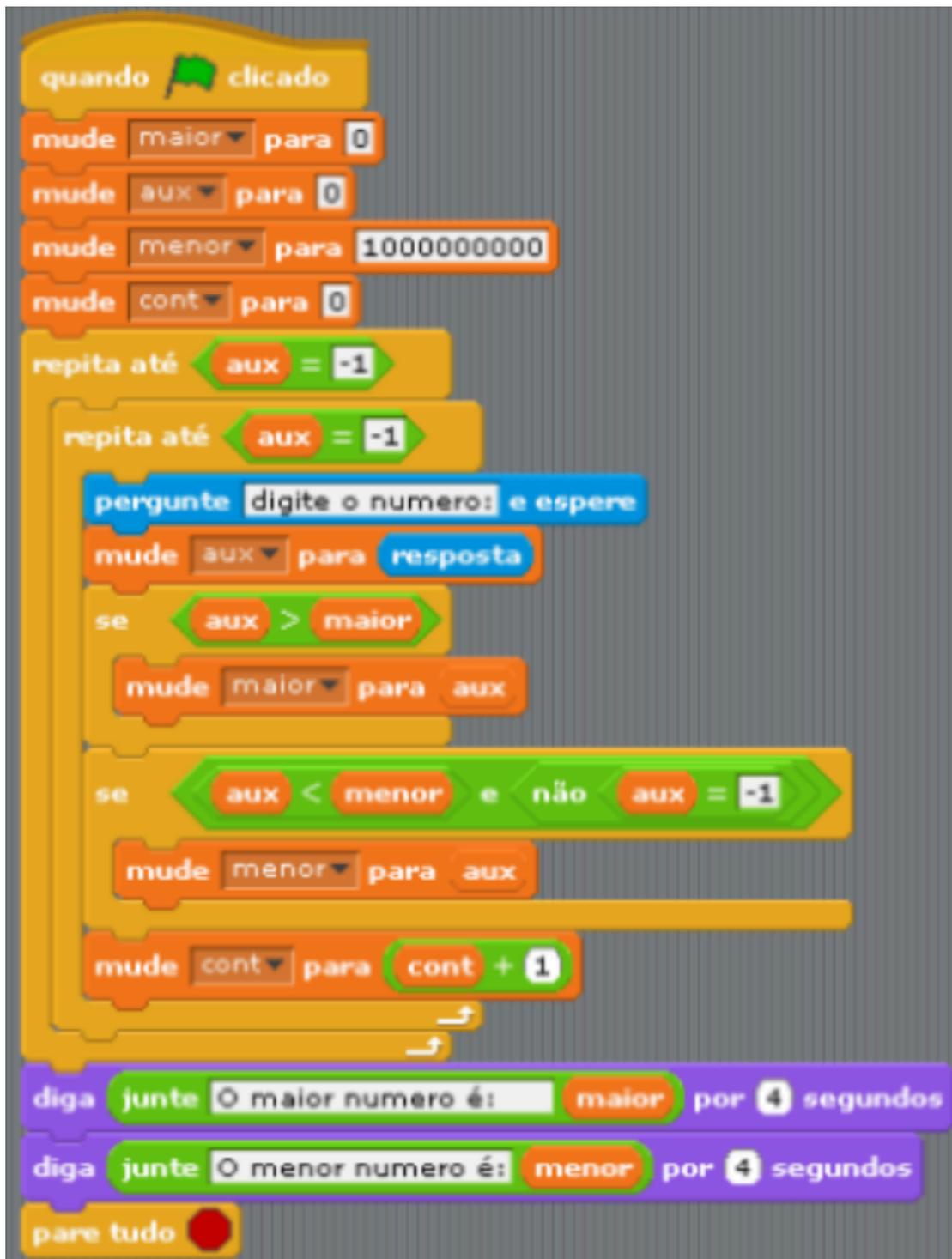
Fonte: (ANASTACIO, 2020)

Atividade 17 - Identificação do Maior e Menor Número

Objetivo: Ensinar o uso de variáveis, operadores matemáticos e estruturas de repetição e condição no Scratch. Os alunos aprenderão a coletar vários valores inseridos pelo usuário, processar os dados para determinar o maior e o menor número e encerrar a entrada com um valor de parada.

Descrição: Crie um algoritmo que permita ao usuário inserir vários números inteiros. O processo continuará até que o usuário digite -1, momento em que o programa exibirá o maior e o menor número digitado.

1. Crie as variáveis "Número", "Maior" e "Menor".
2. Defina "Maior" como um valor muito pequeno (exemplo: -9999) e "Menor" como um valor muito grande (exemplo: 9999) para garantir que o primeiro número digitado seja considerado corretamente.
3. Solicite ao usuário que insira um número inteiro.
4. Enquanto o número não for -1, o programa deve continuar pedindo mais números.
5. Se for -1, a entrada de dados termina e os resultados são exibidos.
6. Se o número digitado for maior que "Maior", atualize a variável "Maior".
7. Se o número digitado for menor que "Menor", atualize a variável "Menor".
8. Após a entrada de -1, exiba os valores armazenados nas variáveis "Maior" e "Menor".



Fonte: (ANASTACIO, 2020)

Atividade 18 - Cálculo da Média de 4 Notas e Verificação de Aprovação

Objetivo: Ensinar o uso de variáveis, operadores matemáticos e estruturas de repetição e condição no Scratch. Os alunos aprenderão a coletar notas inseridas pelo usuário, calcular a média e determinar se o aluno foi aprovado ou reprovado.

Descrição: Crie um algoritmo que solicite quatro notas do usuário, calcule a média e exiba se o aluno está aprovado ou reprovado com base na média mínima de 6.0.

1. Crie as variáveis "Nota", "Soma" e "Média".
2. Defina "Soma" e "Média" como 0 no início do programa.
3. Utilize um bloco de repetição para solicitar 4 notas do usuário.
4. A cada iteração, peça ao usuário para inserir uma nota e some esse valor à variável "Soma".
5. Após a repetição, calcule a média dividindo "Soma" por 4.
6. Se a média for maior ou igual a 6, exiba "Aprovado".
7. Caso contrário, exiba "Reprovado".
8. Após a exibição do resultado, redefina as variáveis para 0 para permitir novas execuções.

```
quando clicado
  repita 4
    pergunte digite a nota do aluno e espere
    mude nota para resposta
    mude soma para nota + soma
  mude media para soma / 4
  diga media por 2 segundos
  se <media > 6 ou <media = 6>
    diga Aprovado por 2 segundos
  senão
    diga Reprovado por 2 segundos
  mude media para 0
  mude nota para 0
  mude soma para 0
  pare tudo
```

The image shows a Scratch script designed to calculate the average of four student grades. It begins with a 'when clicked' event, followed by a 'repeat 4' loop. Inside the loop, it asks the user for a grade, updates a 'nota' variable, and adds it to a 'soma' variable. After the loop, it calculates the 'media' (average) by dividing 'soma' by 4 and displays it for 2 seconds. A conditional 'if' block checks if the average is greater than or equal to 6, displaying 'Aprovado' (Approved) or 'Reprovado' (Failed) accordingly. Finally, it resets the 'media', 'nota', and 'soma' variables to 0 and ends with a 'stop all' block.

Fonte: (ANASTACIO, 2020)

Referências

ANASTACIO, P. R. O Uso do Scratch no Ensino de Programação. 2020. Dissertação (Mestrado Profissional em Ensino) — Universidade Estadual do Norte do Paraná, 2020. Disponível

em: <<https://uenp.edu.br/mestrado-ensino-publicacoes/ppgen-dissertacoes-defendidas/ppgen-dissertacoes-defendidas-3-turma-2018-2019/16449-paulo-roberto-anastacio/file>>.

REZENDE, C. M. C.; BISPO, E. L. Comparação entre o uso de pseudocódigo e a programação visual no ensino de programação: Uma avaliação a partir da ferramenta scratch. In: Conferência Ibérica de Sistemas e Tecnologias de Informação (CISTI). Cáceres: CISTI, 2018. p. 1–5.

SCRATCH. SCRATCH. Imagine, program, share. 2017. Acesso em: 11 Jun 2019. Disponível em: <<http://scratch.mit.edu>>.

SOUZA, M. F. d. Ensino e Tecnologias: aprendizagem significativa por meio da plataforma Scratch. 2019. 100 p. Dissertação (Dissertação (Mestrado)) — Colégio Pedro II, Pró-Reitoria de Pós-Graduação, Pesquisa, Extensão e Cultura, Programa de Mestrado Profissional em Práticas de Educação Básica, Rio de Janeiro, 2019.