

Capítulo

3

Desenvolvendo o pensamento computacional utilizando Scratch e lógica matemática

Gilson Pedroso dos Santos e Ronilson dos Santos Bezerra

Abstract

This work discusses the importance of computational thinking and Scratch, a tool that enables the creation of an environment where it is possible, among other things, to learn and teach mathematical logic, using logical, arithmetic and relational operators. In the development of this, a bibliographical research was done on Computational Thinking, Scratch, Mathematical Logic, and interactive tutorials have been created for the teaching of the basic use of Scratch and the teaching of mathematical logic. The results converge to the reflection of the use of the Scratch tool in teaching environments, of computational thinking, as one of the main skills that the individual must possess in the current globalized society and how logic can be learned and taught through a computational resource.

Resumo

Este trabalho discute a importância do pensamento computacional e do Scratch, ferramenta esta que possibilita a criação de um ambiente onde é possível, dentre outras coisas, aprender e ensinar sobre lógica matemática, utilizando operadores lógicos, aritméticos e relacionais. No desenvolvimento deste, foi realizada uma pesquisa bibliográfica sobre Pensamento Computacional, Scratch, Lógica Matemática, além de ter sido criados tutoriais interativos para o ensino do uso básico de Scratch e do ensino de lógica matemática. Os resultados convergem para a reflexão do uso da ferramenta Scratch em ambientes de ensino, do pensamento computacional, como uma das principais habilidades que o indivíduo deve possuir na atual sociedade globalizada e como a lógica pode ser aprendida e ensinada através de um recurso computacional.

3.1. Introdução

O presente trabalho trata sobre a utilização da ferramenta Scratch no ensino da lógica matemática e no desenvolvimento do pensamento computacional.

Possui como objetivo discutir a importância do pensamento computacional e a sua aplicabilidade no ensino, bem como apresentar uma introdução ao uso da ferramenta Scratch e verificar como a lógica matemática pode ser ensinada utilizando esse recurso computacional.

De acordo com Blikstein (2008) uma pessoa do século XXI precisa dominar muito mais que as habilidades básicas, tais como ler e escrever, que eram exigidas nos séculos passados. Neste sentido, o autor destaca o Pensamento Computacional como uma habilidade indispensável para as pessoas no século XXI. No entanto este pensar não se trata apenas de saber utilizar os aplicativos de um computador, mas sim de utilizar o computador para aumento de sua capacidade cognitiva. Dessa forma, o Scratch, por ser uma ferramenta de programação em blocos, pode ajudar os estudantes deste século a desenvolverem a habilidade de “pensar computacionalmente”. Esta ferramenta, somada ao ensino da lógica matemática, pode colaborar, muito positivamente, para capacitar o estudante a ser mais proativo na busca por soluções de problemas, além de ter aumentada sua capacidade de criatividade e inventividade.

Como procedimento metodológico foram realizadas pesquisas bibliográficas sobre o pensamento computacional, Scratch e Lógica matemática, além de terem sido criados tutoriais interativos para o ensino do uso básico do Scratch e do ensino de lógica matemática utilizando esta ferramenta.

A seguir é descrito como este trabalho foi organizado. Na seção 2 foi feita uma revisão da literatura sobre os temas, onde na subseção 2.1 foi abordado sobre o Pensamento Computacional, conceito este que pode ser aplicado no processo educativo. Por sua vez na subseção 2.2 dissertou-se sobre o Scratch, na qual foi apresentada a ferramenta e suas possibilidades de uso no ensino. Na subseção 2.3 explicou-se sobre a Lógica Matemática onde foram discutidos os conceitos relacionados às operações lógicas e suas respectivas tabelas-verdade. Já na subseção 2.4 elucidou-se sobre o uso do Scratch no ensino da lógica para o desenvolvimento do pensamento computacional. Na seção seguinte (3) foi desenvolvida a proposta de atividade, utilizando o Scratch no ensino da lógica para o desenvolvimento do pensamento computacional, Neste foram propostas atividades que podem ser feitas no ambiente Scratch com a finalidade de se desenvolver a habilidade do “pensar computacionalmente” e como se ensinar e aprender lógica utilizando Scratch. Nas considerações finais (seção 4) mencionou-se sobre a importância da atividade proposta para o desenvolvimento do Pensamento Computacional.

3.2. Referencial Teórico

3.2.1. Pensamento Computacional

Esta subseção traz alguns conceitos sobre o pensamento computacional e aborda sua aplicabilidade no ensino.

Nas últimas décadas a computação, enquanto ciência, foi ganhando espaço e se disseminando na sociedade, especialmente à medida que os computadores deixaram de

ser exclusividade de forças militares e de grandes empresas, passando a fazer parte do cotidiano das pessoas, nas mais diversas formas e modelos de dispositivos, como por exemplo: desktops, notebooks, caixas de autoatendimento de bancos, celulares, computadores de bordo dos veículos, além de muitos outros.

A popularização da computação, somada ao crescente uso da internet, mudou completamente a forma como as pessoas se relacionam com as informações. Enquanto antes as pessoas dependiam dos jornais impressos, rádio ou televisão para que se mantivessem informados, atualmente a informação pode ser obtida a partir de uma infinidade de dispositivos conectados à internet, tais como celulares, *tablets*, *notebooks*, relógios digitais e outros. Com isto, mudou completamente a forma e a velocidade com que as pessoas têm acesso às informações que desejam.

Diante disto, nos últimos anos tem crescido as discussões sobre a inserção das Tecnologias da Informação e da Comunicação (TIC) no processo de ensino-aprendizagem, devido à necessidade do cidadão do século XXI saber interagir com as tecnologias que estão a sua volta. Todavia, França et al. (2014 apud Nunes, 2008) esclarece que é necessário diferenciar o ensino da informática do ensino da computação – apesar de em alguns países estas duas palavras possuírem o mesmo significado. Além disso, os autores ressaltam que o ensino sobre a utilização de aplicativos não pode ser confundido com o ensinar computação.

Blikstein (2008), ao falar sobre a computação no processo educativo, destaca que o mundo atual exige muito mais do que ler, escrever, somar e subtrair. De acordo com o autor é preciso ter diversas habilidades e conhecimentos para o exercício pleno da cidadania no século XXI. Dentre as habilidades necessárias o autor destaca o “pensamento computacional”. Para Wing (2006) o pensamento computacional é uma habilidade fundamental para todos e não apenas para os cientistas da computação. De acordo com a autora os computadores facilitam a disseminação do pensamento computacional e essa temática deve ser acrescentada à grade curricular da educação básica.

O pensamento computacional trata-se não apenas de saber navegar na internet ou utilizar um aplicativo, mas sim a habilidade de saber usar o computador como um instrumento de aumento de poder cognitivo e operacional humano, possibilitando o aumento da produtividade, inventividade e criatividade humana (BLIKSTEIN, 2008). Wing (2006) destaca como exemplos o uso do pensamento computacional em cálculos estatísticos permitindo resoluções de problemas em grande escala, tamanho e dimensões inimagináveis até pouco tempo.

O “pensar computacionalmente”, de acordo com Blikstein (2008) pode ser dividido em duas etapas: a) identificar as tarefas cognitivas que podem ser realizadas de forma mais rápida e eficiente com o uso da computação; b) saber programar um computador para que este realize as tarefas cognitivas (identificadas na “etapa a”). Sendo assim aquilo que não é necessariamente humano é transferido para que o computador execute, permitindo que programas de computadores possam auxiliar nos processos produtivos, pesquisas científicas e até mesmo otimização de rotinas pessoais, evitando perda de tempo humano com tarefas repetitivas.

Para França et al. (2014 apud Nunes, 2008) ensinar o aluno a “pensar computacionalmente” torna possível “a libertação autêntica” defendida por Freire

(1987) na obra *Pedagogia do oprimido*, onde o autor afirma que existirá a “ação e a reflexão dos homens sobre o mundo, para transformá-lo” (FREIRE, 1987).

No entanto, apesar da importância deste tema, França e Tedesco (2015) destacam que diferente dos Estados Unidos e alguns países da Europa que têm buscado inserir o ensino da computação na grade curricular, o mesmo ainda não acontece no Brasil, onde esta disciplina é ofertada apenas para cursos voltados especificamente às áreas de tecnologias (seja curso técnico ou superior), salvo algumas iniciativas envolvendo estudantes de níveis fundamental e médio.

Diante disto França e Tedesco (2015) ponderam, que apesar do crescente interesse dos pesquisadores de vários países, inclusive do Brasil, na disseminação do pensamento computacional, o ensino de habilidades computacionais na educação básica do Brasil ainda pode ser considerado um desafio, ao mesmo tempo em que é uma oportunidade aos educadores.

Todavia, apesar do desafio, França e Tedesco (2015) destacam importantes iniciativas e relatos de experimentos, voltados para a área e realizados em vários estados brasileiros, que vem dando certo, além de, atualmente, servirem de fontes inspiradoras para que muitos educadores prossigam com a propagação deste importante tema em suas escolas.

Dentre os trabalhos, citados por França e Tedesco (2015), destacam-se: olimpíadas voltadas para programação, curso de lógica de programação utilizando o ambiente do App Inventor (www.appinventor.mit.edu), projeto voltado para o desenvolvimento do raciocínio lógico-matemático, peças teatrais para facilitar a compreensão dos princípios do pensamento computacional e construção de jogos digitais.

Nas várias abordagens utilizadas no ensino do pensamento computacional, utilizando como meio a lógica de programação para promover esta habilidade, ganha destaque alguns ambientes visuais de programação, tais como Scratch, App Inventor, AgentSheets e RoboMind (FRANÇA E TEDESCO, 2015). No presente trabalho optou-se pela ferramenta Scratch por ser uma das mais conhecidas, com uma vasta documentação – o que facilita para os iniciantes, é gratuito, tem código aberto, além de dispensar a necessidade de internet para utilizá-lo (a não ser para fazer o *download* do software).

3.2.2. Scratch

Esta subseção trata sobre a ferramenta Scratch e seu uso na disseminação do pensamento computacional.

Para Omine (2014), conhecer a computação e programação permite ao indivíduo compreender melhor como os sistemas computacionais, com os quais ele interage, são construídos, possibilitando um conhecimento melhor das intenções por trás deste sistema. Isso reforça a afirmação de Omine (apud Lemos, 2013) que afirma que o conhecimento de programação é tão essencial, até mesmo para a manutenção da democracia. Segundo o autor “a possibilidade de programar e realizar simulações utilizando a computação, testando hipóteses, permite ao homem lidar com problemas mais complexos, diversificando sua ação no mundo”.

Todavia, ensinar programação para quem ainda não tem nenhum conhecimento de lógica de programação ou algoritmos não é tarefa fácil. Diante disto, pesquisadores, desde o início dos anos 1960, vêm buscando formas de desenvolver linguagens e ambientes de programação que sejam mais acessíveis para um maior número de pessoas. Isto porque uma das grandes dificuldades encontradas por muitos usuários é a compreensão da sintaxe, devido à necessidade de compreender perfeitamente a linguagem que o computador entende e também o idioma utilizado, uma vez que algumas sintaxes não são claras o suficiente para um iniciante em programação. Diante disto, nestes últimos anos tem-se pesquisado formas de como deixar a linguagem de programação o mais próximo da linguagem natural (SCAICO et al.. apud PAUSH E KELLEHER, 2005).

É neste contexto que o Scratch destaca-se como um importante ambiente de auxílio ao ensino da programação, por utilizar o conceito de desenvolvimento orientado ao design, permitindo com que mesmo usuários iniciantes (ou até mesmo, crianças) consigam criar um primeiro projeto em alguns minutos. Isto porque utilizando o Scratch o usuário não precisa, inicialmente, dominar a sintaxe ou idioma de uma linguagem de programação. Em vez disto, devido à metodologia “arrastar e soltar” – também conhecida como programação em blocos - oferecida pela ferramenta, o usuário pode ir arrastando blocos de códigos e criando o seu projeto, seja jogos, animações, histórias e outros.

Scratch é uma linguagem de programação criada e mantida *pele Lifelong Kindergarten* do grupo *Massachusetts Institute of Technology (MIT) Media Lab*. Surgida em 2007 a plataforma, que é disponibilizada no link www.scratch.mit.edu (ver Figura 1), permite que os usuários criem e compartilhem mídias interativas, tais como histórias, jogos e etc. (LIFELONG KINDERGARTEN GROUP, 2017).

Inicialmente concebido com a ideia de atender pessoas na faixa etária de 8 a 16 anos, o Scratch logo passou a ser utilizado por pessoas de todas as idades. Atualmente, diversos educadores do nível fundamental, médio e até mesmo dos cursos técnicos ou nível superior têm se utilizado desta ferramenta em suas práticas pedagógicas. A proposta do Scratch é que os alunos aprendam a programar suas próprias histórias e aprendam a pensar de forma criativa e raciocinar sistematicamente à medida que desenvolvem estas histórias (LIFELONG KINDERGARTEN GROUP, 2017).

Além disso, os educadores que fazem uso do Scratch podem contar com o apoio de comunidades de educadores - como a *ScratchEd* desenvolvida e mantida pela *Haward Graduate School of Education* - que estão dispostos a compartilhar suas histórias, trabalhando de uma forma colaborativa, facilitando o conhecimento de novos recursos e da absorção de novas ideias que podem ser implementadas com o Scratch. Por ser utilizado em mais de 150 países e disponível em mais de 40 línguas esta ferramenta tornou-se muito acessível, impactando no número de projetos que são criados diariamente e que ficam disponíveis para que outros usuários façam o “reúso” do conhecimento ali contido. Atualmente, há uma vasta documentação sobre o Scratch, facilitando o processo de aprendizagem dos educadores que iniciam o uso desta ferramenta (LIFELONG KINDERGARTEN GROUP, 2017).



Figura 1: Página inicial do website do Scratch. Fonte: <https://scratch.mit.edu>. Acesso em 01 de Julho de 2017.

O Scratch permite que seus usuários absorvam vários conhecimentos enquanto criam seus projetos, entre os quais - tomando como base o artigo “*Learning with Scratch*”¹ - destacam-se:

- **Conhecimento matemático e computacional** – isto porque enquanto criam seus projetos os usuários vão absorvendo os conceitos básicos da computação, como iteração e condicionais, além de outros conceitos matemáticos como coordenadas, variáveis e números aleatórios, por exemplo.
- **Processo de “Design”** – um usuário do Scratch cria uma nova ideia, faz um protótipo, realiza testes, depura quando surgem erros, recebe comentários de outros usuários e isto permite que o usuário revise seu projeto inúmeras vezes durante este processo de criação. Estes passos continuam em uma espécie de ciclo contínuo (Figura 2) onde uma ideia pode dar origem a um projeto, e este leva a outros projetos e assim sucessivamente, fazendo com o que o usuário de Scratch esteja em um constante processo de aprendizagem e construção do conhecimento. Este processo cíclico auxilia no desenvolvimento de cidadãos com pensamento crítico, que possuem habilidade de comunicar-se, com capacidade de realizar análise sistemática, colaborar efetivamente e aprender de forma contínua.
- **Maior “fluência” com a tecnologia digital** – nesta concepção ser fluente em tecnologia digital é não apenas saber utilizar um computador, digitar um texto

¹ Learning with Scratch – the Scratch Documentation Website - MIT. Traduzido por Teresa Martinho Marques. Disponível em: http://repositorio.ul.pt/bitstream/10451/847/49/20148_ulsd_dep.17852_tm_anexo38a.pdf. Acesso em 01 de Junho de 2017.

ou acessar a internet. Tomando como exemplo um idioma, por exemplo, para se tornar fluente não basta saber ler naquele idioma, mas é preciso, também, saber se expressar utilizando o mesmo. Portanto, ser fluente digitalmente é não apenas interagir com o computador, mas também criar algo com ele. Portanto, os ideais do Scratch coadunam perfeitamente com o conceito do Pensamento Computacional conforme exposto na seção 0.

O ciclo da criação do projeto no Scratch (mencionado anteriormente e ilustrado na Figura 2) correlaciona-se com o ciclo da aprendizagem ou ciclo *descrição-execução-reflexão-depuração* abordado por Valente (2005) quando afirma que:

Inicialmente, o computador era entendido como meio para representar o conhecimento do aprendiz, explicitando o raciocínio usado na resolução de problemas ou projetos. No entanto, fica patente que, além de representado, o “raciocínio” pode ser também executado por intermédio da máquina, oferecendo resultados que servirão para melhorar esta representação e, conseqüentemente, o conhecimento subjacente. Com isto, surgiu a ideia de ciclo de ações descrição-execução-reflexão-depuração.

Além de possuir todas essas potencialidades para o ensino-aprendizagem, o Scratch também conta com uma enorme quantidade de tutoriais e projetos (uma vez que os próprios usuários podem compartilhar suas criações) e estes ficam disponíveis, gratuitamente, em seu website. Tais projetos podem ser utilizados para o aprendizado autônomo ou mesmo como fonte de inspiração para criação de outros projetos.

Esta quantidade de projetos disponíveis aumenta a cada dia, assim como a interação dos usuários, comentando os projetos existentes, em uma verdadeira rede colaborativa. De acordo com dados estatísticos, obtidos no referido site, só em Maio de 2017 foram criados mais de 1 milhão de novos projetos de Scratch e mais de 3 milhões de comentários foram postados nos projetos existentes. Esses números são ascendentes desde 2007 (quando o projeto nasceu), conforme pode ser visto no Gráfico 1. É importante ressaltar que a maioria dos novos usuários são adolescentes, com faixa etária entre 9 e 17 anos, conforme pode ser observado no Gráfico 2.

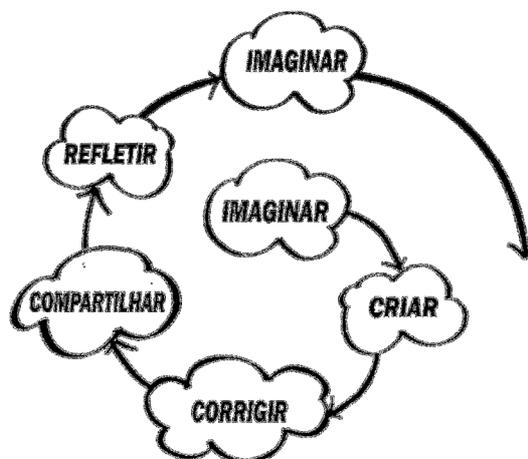


Figura 2: Ciclo de criação no Scratch1

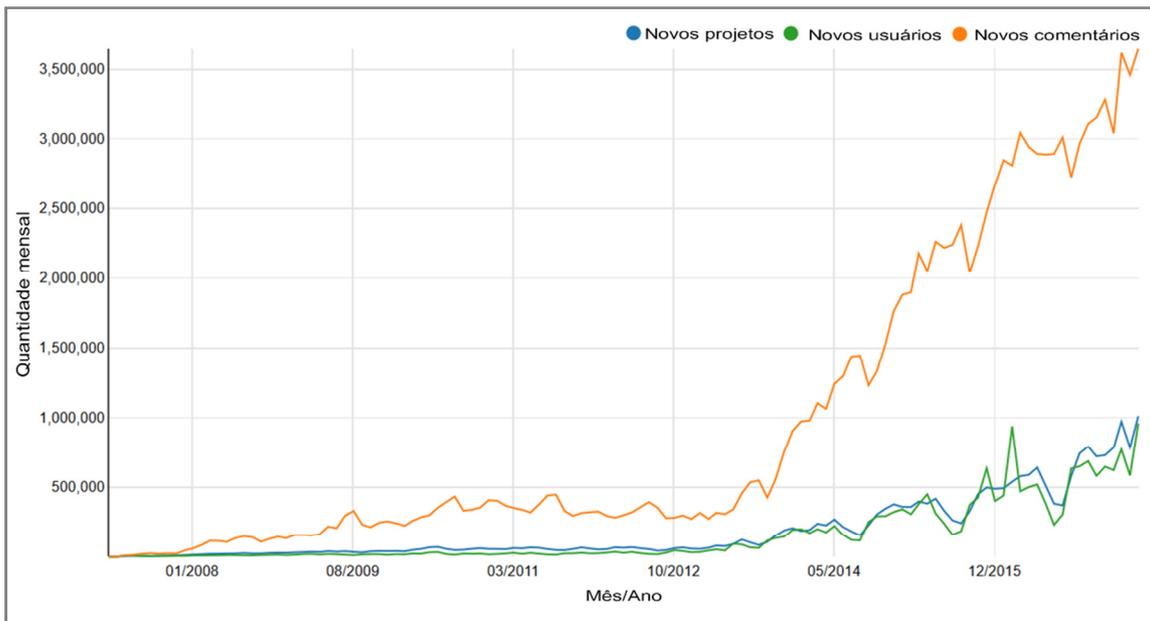


Gráfico 1: Atividades mensais no website no Scratch. Traduzido pelos autores. Fonte: <https://scratch.mit.edu/statistics/>. Acesso em 01 de Julho de 2017.

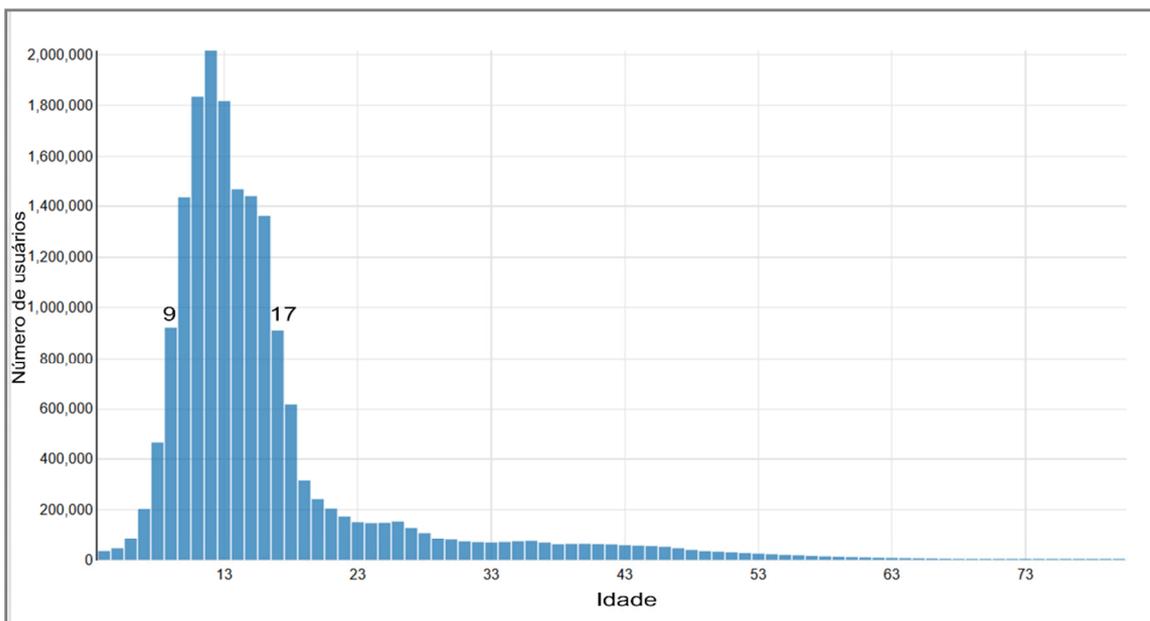


Gráfico 2: Distribuição de idade dos novos usuários de Scratch. Traduzido pelos autores. Fonte: <https://scratch.mit.edu/statistics/>. Acesso em 01 de Julho de 2017.

Por ser embasada nas ideias construcionistas – uma vez que a ferramenta foi desenvolvida a partir da linguagem Logo (criada por Seymour Papert), que utilizava este conceito - o Scratch permite que o usuário não apenas aprenda uma lógica de programação, mas também desenvolva habilidades como o pensamento criativo, análise crítica e aprendizagem contínua. Devido a estas características o Scratch tem se mostrado um importante aliado dos pesquisadores e docentes que desejam não apenas ensinar uma linguagem de programação aos seus alunos, mas também – e

principalmente – fazer com que seus alunos possam desenvolver a habilidade de “pensar computacionalmente”.

Além do mais, o Scratch propicia que o educador se utilize de inúmeras possibilidades que permitem que as atividades sejam elaboradas de acordo com o contexto escolar, com o conteúdo programático de determinada disciplina, com a faixa etária dos estudantes etc.

3.2.2.1. Experiências no Brasil do uso do Scratch

No Brasil há diversas iniciativas de uso do Scratch em escolas, que se apropriam desta ferramenta para o ensino no nível fundamental, médio, técnico ou superior, todavia a quantidade de usuários ainda é baixa, quando comparada com outros países do mundo.

De acordo com dados oficiais, há aproximadamente 20 milhões de usuários do Scratch no mundo. Destes, apenas 1,5% (pouco mais de 286 mil) são brasileiros (conforme Gráfico 3), enquanto que 44,80% (mais de 8,5 milhões) são dos Estados Unidos, por exemplo. Ao realizar esta comparação é importante destacar que a população dos EUA é aproximadamente 54% maior que a do Brasil, assim como a facilidade de acesso às tecnologias, dentre outros fatores. Entretanto, pode-se citar, por exemplo, a Colômbia que apesar de ter uma população correspondente a ¼ da população brasileira, possui mais de 230 mil usuários.

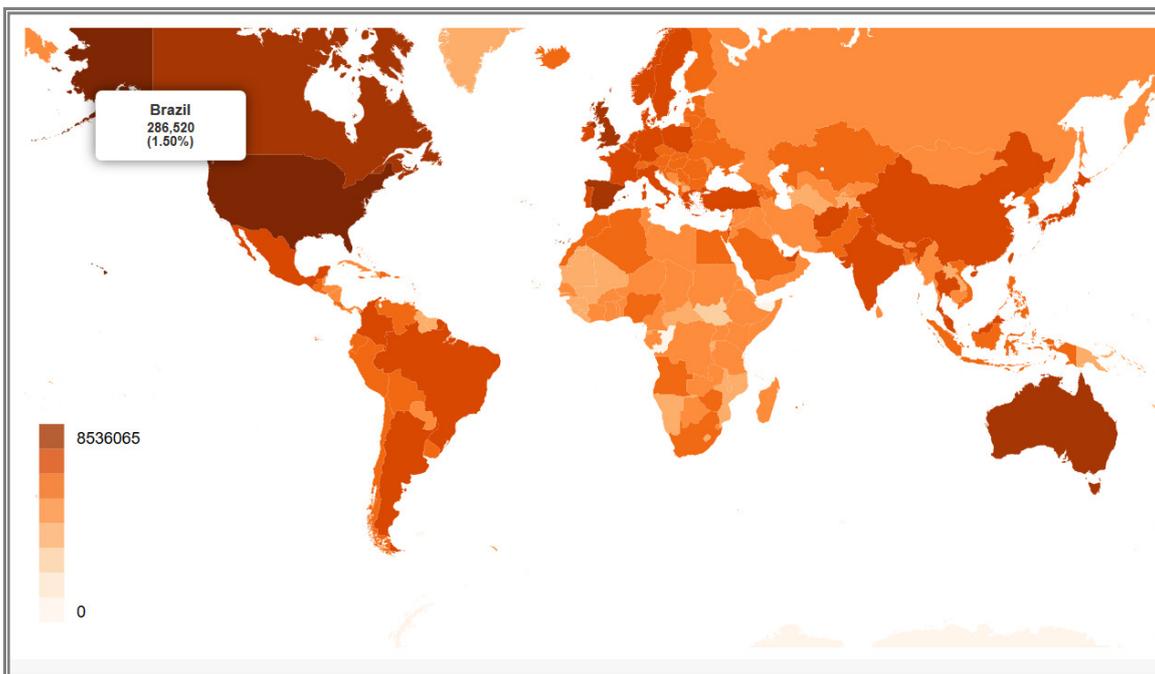


Gráfico 3: Gráfico do quantitativo de usuários do Scratch no mundo, com destaque para o quantitativo de usuários no Brasil. Fonte: <https://scratch.mit.edu/statistics/>. Acesso em 01 de Julho de 2017.

Ainda assim, o Brasil destaca-se por estar entre os quatro países da América Latina (os outros três são México, Colômbia e Argentina) com maior quantidade de usuários. Apesar disso, os números supracitados mostram que há muito espaço para expansão do uso desta ferramenta no Brasil.

Dentre as inúmeras formas pedagógicas de utilizar o Scratch, Sobreira, Takinani e Santos (2013) destacam algumas mais conhecidas, as quais são listadas nas subseções seguintes, com destaques (e comentários) destes autores sobre alguns pontos mais relevantes.

3.2.2.1.1. Animações

Estas permitem a criação de histórias com animações, diálogos, efeitos sonoros e movimentos. Considera-se que esta atividade é ideal para iniciação do ensino da programação para estudantes de nível fundamental, especialmente porque utilizam recursos básicos do Scratch, e por isto são mais fáceis, além de possibilitar a observação dos resultados rapidamente. Além disso, no website do Scratch (conforme mencionado na subseção 0) há uma grande quantidade de tutoriais que facilitam esta iniciação.

3.2.2.1.2. Jogos de tabuleiros

Proporcionam a criação de jogos a partir da criatividade do programador. Este tipo de atividade torna possível a exploração de outros recursos da ferramenta Scratch (ainda não utilizados na atividade anterior), envolvendo o uso de recursos tais como sensores, operadores e variáveis.

3.2.2.1.3. Jogos ou games digitais

Oportunizam a exploração de todas as possibilidades do Scratch e explora a criatividade do programador. Segundo Sobreira, Takinani e Santos (2013) este tipo de jogo além de ser atraente ao público infantil e adolescente, oportuniza que os usuários reproduzam seus *games* favoritos ou criem novos jogos. Com isto, além de passarem a compreender o papel de produtor de games, os usuários compreendem os comandos utilizados, desde os mais básicos aos mais avançados.

3.2.2.1.4. Games literários

Criados com a finalidade de divulgar obras literárias – de terceiros ou dos criados pelos próprios usuários. Por este motivo é uma atividade que fomenta o desenvolvimento da capacidade de criação e interpretação do usuário. Desse modo, este tipo de atividade pode ser tornar um aliado para trabalhos de literatura – presente no currículo escolar – enquanto os alunos aprendem também a programar.

3.2.2.1.5. Quiz interativo

Permite ao aluno criar desafios a partir do conhecimento construído. Para isto são elaboradas perguntas e respostas possíveis para tornar exequível a interação das pessoas (usuários) com o jogo.

3.2.2.1.6. Comunicação com dispositivos que utilizam o Sistema Operacional Android

Permite aos usuários a integração de celulares com os jogos criados no Scratch. Isto é possível por meio do uso de algum aplicativo instalado em um dispositivo com o sistema operacional Android, como o *Scratcher Control* mostrado na Figura 3. Com isto há a possibilidade de controlar um projeto criado no Scratch por meio de uma rede de internet sem fio.



Figura 3: aplicativo Scratch Control² que permite controlar projetos no Scratch utilizando um dispositivo com Android. Fonte: Scratch Control.

3.2.2.1.7. Comunicação com a plataforma Arduino

Possibilita a programação de objetos físicos a partir do Scratch, o que se torna muito mais fácil do que ensinar ao aluno, iniciante em programação, a programar diretamente na linguagem Arduino (que é uma linguagem que tem algumas semelhanças com C, criada com base nas linguagens Wiring e Processing). Esta integração permite o uso da modificação *Scratch for Arduino*³ ou S4A como é comumente chamado.

Desenvolvido em 2010, pela equipe Smalltalk do Citilab, o S4A (Figura 4) se propõe a permitir a interação entre o Scratch com placas Arduino (plataforma de *hardware open source*)⁴, por meio de uma interface de alto nível de programação, objetivando atrair pessoas para o mundo da programação. Com isto, por meio dos blocos do Scratch é possível controlar sensores e atuadores conectados ao Arduino. Para isto faz-se necessário instalar o S4A no desktop do usuário, além de instalar um *firmware* na placa Arduino. Esses procedimentos, de instalação e configuração, estão detalhados na página oficial do SA4 (<http://s4a.cat>).

Portanto, conforme demonstrado há uma infinidade de possibilidades de utilizações do Scratch no auxílio ao ensino-aprendizagem. Dentre essas possibilidades, neste trabalho, optou-se por utilizar a ferramenta para o ensino da lógica matemática objetivando o desenvolvimento do pensamento computacional.

²Scratcher Control – Disponível no link: <https://is.gd/wJBqLR>. Acesso em 30 de Junho de 2017.

³S4A - Disponível em: http://s4a.cat/index_pt.html. Acesso em 30 de Junho de 2017.

⁴Arduino - Disponível em: <https://playground.arduino.cc/Portugues/HomePage>. Acesso em 25 de Maio de 2017.

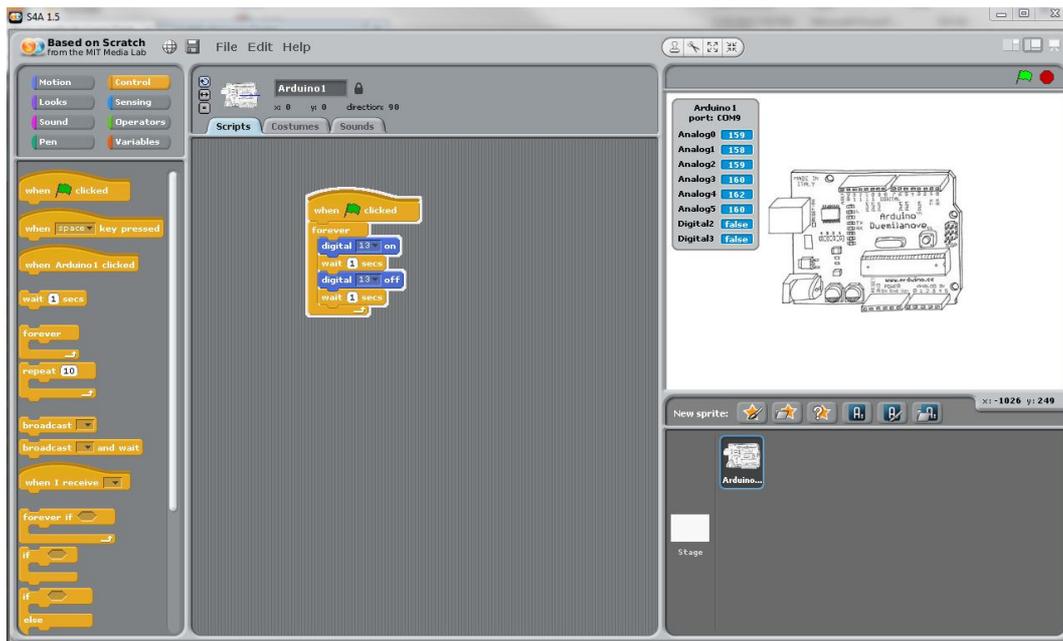


Figura 4: O S4A tem todas as características do Scratch, além de permitir a integração com Placas Arduino. Fonte: S4A.

3.2.3. Lógica Matemática

Esta seção aborda alguns conceitos relacionados à lógica matemática como operações lógicas e suas respectivas tabelas-verdade.

Uma proposição, segundo Souza (2015), é uma sentença declarativa que pode ser interpretada como verdadeira ou falsa. “A lua é retangular”, “A neve é branca” “Matemática é uma ciência” são exemplos de proposição.

A lógica matemática, ramo da matemática que estuda a lógica, é considerada uma das maneiras de representar o conhecimento. Ferreira (2011, p.5) afirma que “Lógica Matemática tem hoje aplicações concretas extremamente importantes, em diversos domínios; uma das mais notáveis é, sem dúvida, a sua utilização no planejamento dos modernos computadores eletrônicos.” Temos ainda a Lógica Proposicional ou Cálculo Proposicional onde são usadas proposições e relações lógicas entre as proposições. Estas podem ser consideradas verdadeiras ou falsas.

De Alencar Filho (2002) informa que há diversas operações lógicas como conjunção, disjunção, negação, condicional, bicondicional, dentre outras.

Tabela 1. Operações lógicas

Valor Lógico	Símbolo	Representação
Conjunção	\wedge	e
Disjunção	\vee	ou
Negação	\sim	Não
Condicional	\rightarrow	Se...então
Bicondicional	\leftrightarrow	Se e somente se

Há a conjunção entre duas proposições p e q , onde o valor lógico é verdadeiro (V) quando ambas proposições forem verdadeiras. Se pelo menos uma das proposições for falsa (F) o valor lógico será falso (F), conforme mostrado na Tabela 2.

Tabela 2. Tabela Verdade da Conjunção

p	q	$p \wedge q$
V	V	V
V	F	F
F	V	F
F	F	F

Na disjunção de duas proposições p e q , o valor lógico é verdadeiro se pelo menos uma das proposições for verdadeira e será falso somente se ambas as proposições forem falsas, conforme destacado na Tabela 3.

Tabela 3. Tabela Verdade da Disjunção

p	q	$p \vee q$
V	V	V
V	F	V
F	V	V
F	F	F

Na negação “não p ” ($\sim p$) o valor lógico será falso quando a proposição for verdadeira e terá valor lógico verdadeiro quando a proposição for falsa, conforme Tabela 4.

Tabela 4. Tabela Verdade da Negação

p	$\sim p$
V	F
F	V

Na condicional o valor lógico será falso somente quando p for verdadeira e q falsa. Nos demais casos o valor lógico será verdadeiro, conforme Tabela 5.

Tabela 5. Tabela Verdade da Condicional

p	q	$p \rightarrow q$
V	V	V
V	F	F
F	V	V
F	F	V

Na Bicondicional o valor lógico é verdadeiro quando ambas as proposições p e q forem falsas ou ambas verdadeiras. Nos demais casos possuirá valor lógico falso, conforme Tabela 6.

Tabela 6. Tabela Verdade da Bicondicional

p	q	$p \leftrightarrow q$
V	V	V
V	F	F
F	V	F
F	F	V

3.2.4. Utilizando o Scratch no ensino da lógica para o desenvolvimento do pensamento computacional

Nesta seção é apresentada a proposta da utilização do Scratch no ensino da lógica e no desenvolvimento do pensamento computacional.

No website oficial do Scratch é possível encontrar vários tutoriais e alguns materiais de apoio à iniciantes. Todavia, partes desses materiais ainda se encontram em inglês.

Assim, a proposta deste trabalho visa à utilização de um tutorial interativo sobre a introdução a utilização da ferramenta Scratch, bem como uma abordagem da lógica matemática nessa ferramenta. O tutorial interativo proposto é dividido nas seguintes partes: Página de Apresentação, Página Inicial, Como Usar, Menu Principal, O que é o Scratch, Baixando o Scratch, Instalando o Scratch, Conhecendo o Ambiente, Criando uma Animação, Ajuda.

3.2.4.1. Conhecendo o tutorial de uso do Scratch

Este tutorial⁵ foi produzido com a finalidade de ser um material de apoio para as pessoas que desejam aprender a utilizar o Scratch. Para isto, as partes do tutorial buscaram esclarecer, ao usuário, os procedimentos a serem seguidos desde a instalação da ferramenta até a criação de um primeiro projeto, além de apresentar ao usuário a ferramenta e auxiliá-lo no esclarecimento de suas dúvidas. Por meio do “Menu Principal” (Figura 5) o usuário pode ter acesso a essas opções do tutorial.

Ao escolher a opção “O que é Scratch” o usuário terá acesso à uma exposição sucinta sobre a história desta ferramenta. Esta não visa esgotar o conhecimento bibliográfico sobre esta ferramenta, mas sim despertar no usuário o desejo de conhecer um pouco mais sobre a origem da mesma.

A partir do “Menu principal” o usuário também pode escolher a opção “Baixando o Scratch” para aprender como baixar e instalar o software em um computador (Figura 6).

⁵ Disponibilizado no link < <https://goo.gl/i8e2dy> >.



Figura 5: Menu Principal. Fonte: Autores (2017).



Figura 6: Ensinando como baixar o Scratch. Fonte: Autores (2017).

Em Conhecendo o Ambiente (Figura 7), o usuário poderá conhecer a composição da interface da ferramenta Scratch, tais como: barra de menus, arquivo, editar, dicas, *about*, idioma, cenário, iniciar, parar, barra de ferramentas, lista de atores, novo ator, palco, pano de fundo, abas, blocos de comando, área de script, etc. Para cada

uma dessas partes há uma ilustração e um texto informativo sobre ela, inclusive mostrando, com detalhes, os ícones, menus e “área de script” com o objetivo de fornecer ao usuário o conhecimento necessário da ferramenta para que ela possa avançar para que, mesmo que seja um iniciante, consiga explorar os principais recursos e potenciais do Scratch.



Figura 7: Conhecendo o Scratch. Fonte: Autores (2017).

Já em Criando uma Animação (Figura 8) o usuário aprenderá em uma sequência de passos como criar a sua primeira animação na ferramenta Scratch. Esta, apesar de básica, busca mostrar ao usuário como utilizar e organizar os blocos de modo a criar uma história, uma animação, com efeitos, movimentos, cores etc.

Após o usuário obter algumas noções de como se trabalhar com o Scratch, ele será desafiado a praticar lógica matemática. Para isso, utilizaremos os operadores do Scratch, Figura 9, um conjunto de blocos que apresenta operadores aritméticos (adição, subtração, divisão e multiplicação), relacionais (igual, menor que e maior que) e lógicos (e, ou e não). Essa proposta tem objetivo de despertar no indivíduo noções básicas da lógica, exercitando a prática num ambiente computacional. Os exercícios/desafios propostos serão de nível básico a avançado, estimulando dessa forma, o usuário a raciocinar.

Criando uma Animação

Passo 1 - Clique em “Eventos” ;
Passo 2 - Clique e arraste o bloco “Quando clicar em” para a área de script;
Passo 3 - Clique em “Movimento”;
Passo 4 - Clique e arraste o bloco “Mova ... passos” para a área de script;
Passo 5 - Altere a quantidade de passos para 20;
Passo 6 - Clique em “Aparência”;
Passo 7 - Clique e arraste o bloco “diga...” para a área de script;
Passo 8 - Altere a mensagem para “Olá, Mundo!”;
Passo 9 - Clique na bandeirinha verde para visualizar a animação.

Figura 8: Criando uma Animação. Fonte: Autores (2017).



Figura 9: Operadores do Scratch. Fonte: Autores (2017).

3.3. Proposta de Atividade

Nesta seção é apresentada a proposta de atividade a ser desenvolvida, por meio da utilização de um tutorial interativo no auxílio ao ensino de lógica matemática utilizando a ferramenta Scratch.

O "Tutorial Interativo Introdução a Lógica Matemática com o Scratch⁶" foi construído com a finalidade de discutir e investigar como a lógica matemática pode ser ensinada e aprendida através do Scratch.

Ele foi produzido contendo cinquenta e uma telas e dividido nas seguintes partes: **Página de Apresentação, Início, Como Usar, Menu Principal, Conceitos, Operadores, Desafios e Ajuda.**

Na **Página de Apresentação**, Figura 10, tela 1 do tutorial, o usuário terá o primeiro contato com a ferramenta e terá informações como os nomes dos autores, bem como o nome do tutorial.



Figura 10: Página de Apresentação. Fonte: Autores (2017).

Na parte **Início**, Figura 11, o usuário pode verificar o nome do tutorial e encontrará três botões que lhe permitirão ir para Como Usar, Menu Principal ou Ajuda.

⁶ Disponibilizado no link < <https://goo.gl/41Epc5> >.



Figura 11. Início. Fonte: Autores (2017).

Na parte **Como Usar**, Figura 12, pode-se encontrar oito telas (da tela 3 a 10) onde é possível visualizar informações, através de balões explicativos, sobre como utilizar o tutorial interativo.



Figura 12. Como Usar. Fonte: Autores (2017).

Já na parte **Menu Principal**, Figura 13, tela 11, pode-se conhecer as principais funcionalidades do tutoriais, permitindo também uma melhor navegação por todo o tutorial.



Figura 13. Menu Principal. Fonte: Autores (2017).

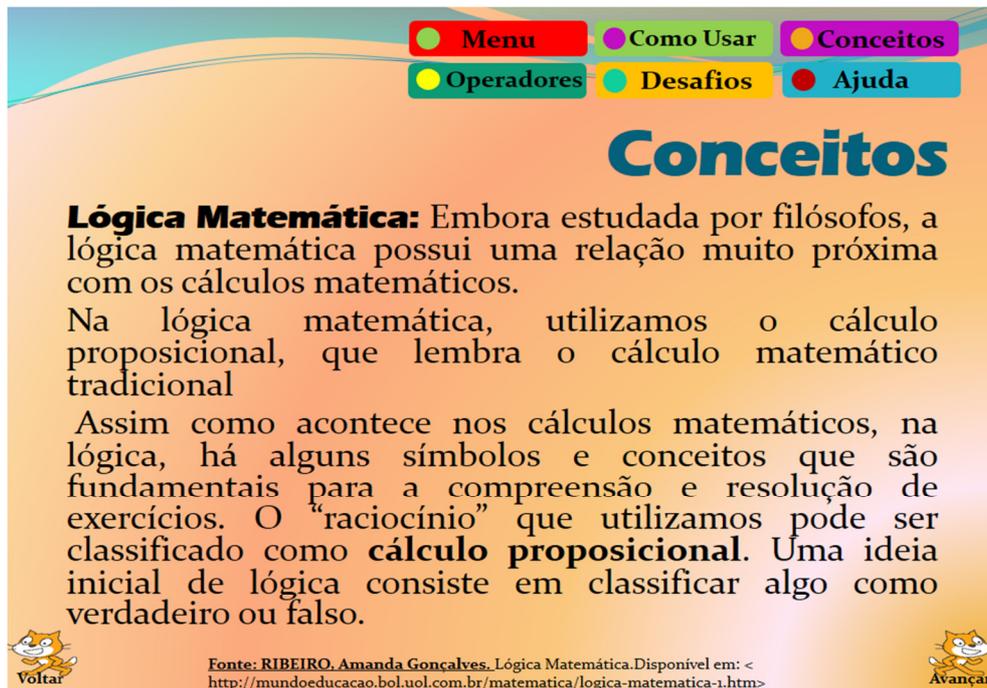
Conceitos, Figura 14, telas 12-23, é uma seção do tutorial interativo onde são apresentados diversos termos relacionados à Lógica Matemática, bem como suas definições. Dessa forma, serão conhecidos os conceitos como Lógica Matemática, Cálculo Proposicional, Argumento Dedutivo, Argumento Indutivo, Conectivos Lógicos, Proposição, Tabela Verdade, dentre outros.

Em **Operadores**, Figura 15, telas 24-35, o usuário conhecerá os operadores do Scratch: **aritméticos** (adição, subtração, divisão e multiplicação), **relacionais** (igual, menor que e maior que) e **lógicos** (e, ou e não).

Para exercitar os conteúdos apresentados no tutorial, propõem-se **Desafios**, Figura 16, telas 36-42, onde o usuário será desafiado a utilizar os operadores do Scratch para aprender um pouco mais sobre lógica matemática.

Vale ressaltar que o tutorial, além do que foi mencionado, possui botões de ação utilizados para proporcionar uma navegação mais rápida pelo tutorial, balões explicativos para informar sobre algum elemento da ferramenta, dentre outros recursos auxiliares.

A última parte é o **Ajuda**, telas 43-51, Figura 17, onde o usuário poderá obter mais informações sobre o tutorial.



The screenshot shows a navigation menu at the top with six items: Menu (green), Como Usar (purple), Conceitos (yellow), Operadores (orange), Desafios (red), and Ajuda (blue). The 'Conceitos' item is highlighted. Below the menu, the title 'Conceitos' is displayed in large blue font. The main text discusses mathematical logic, its relationship to mathematics, and the use of propositional logic in exercises. It mentions that logic is studied by philosophers and that propositional logic is used in exercises, which can be classified as propositional calculation. A source citation is provided at the bottom: 'Fonte: RIBEIRO, Amanda Gonçalves, Lógica Matemática. Disponível em: <http://mundoeducacao.bol.uol.com.br/matematica/logica-matematica-1.htm>'. The interface includes 'Voltar' and 'Avançar' buttons with cat icons.

Menu **Como Usar** **Conceitos**
Operadores **Desafios** **Ajuda**

Conceitos

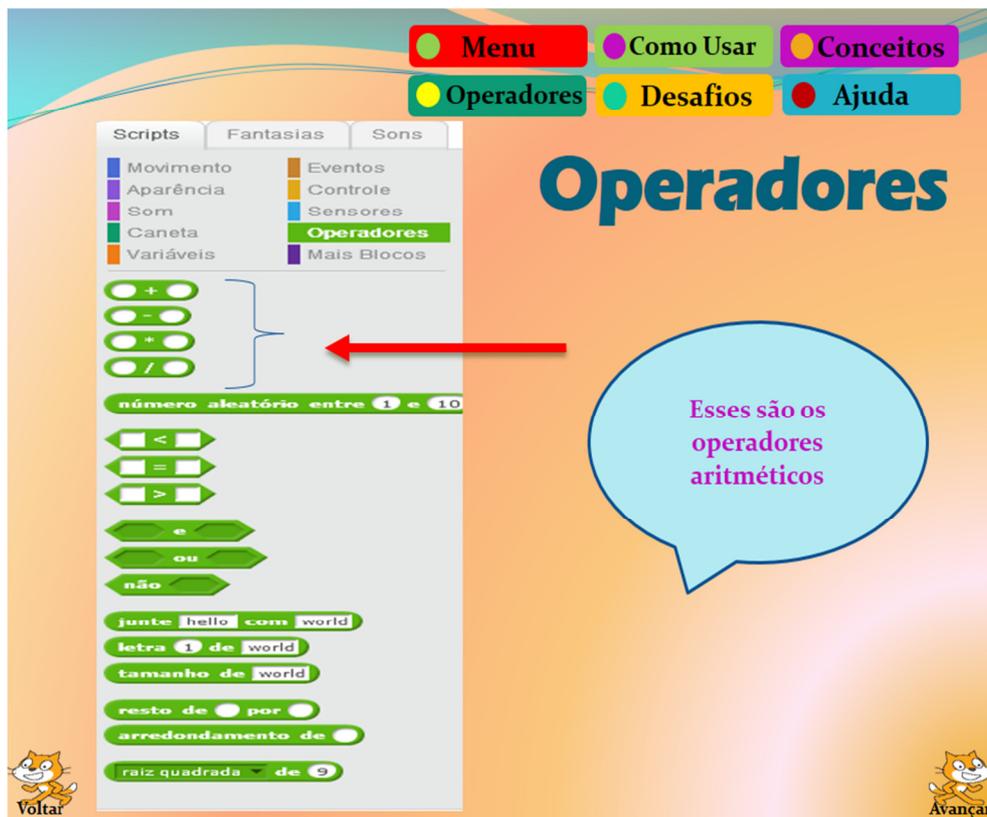
Lógica Matemática: Embora estudada por filósofos, a lógica matemática possui uma relação muito próxima com os cálculos matemáticos.

Na lógica matemática, utilizamos o cálculo proposicional, que lembra o cálculo matemático tradicional

Assim como acontece nos cálculos matemáticos, na lógica, há alguns símbolos e conceitos que são fundamentais para a compreensão e resolução de exercícios. O “raciocínio” que utilizamos pode ser classificado como **cálculo proposicional**. Uma ideia inicial de lógica consiste em classificar algo como verdadeiro ou falso.

Fonte: RIBEIRO, Amanda Gonçalves, Lógica Matemática. Disponível em: <<http://mundoeducacao.bol.uol.com.br/matematica/logica-matematica-1.htm>>

Figura 14. Conceitos. Fonte: Autores (2017).



The screenshot shows the same navigation menu as Figure 14, but with 'Operadores' highlighted. The main title is 'Operadores' in large blue font. On the left, a list of Scratch blocks is shown, with 'Operadores' highlighted in green. A red arrow points from a speech bubble to the arithmetic operator blocks. The speech bubble contains the text 'Esses são os operadores aritméticos'. The interface includes 'Voltar' and 'Avançar' buttons with cat icons.

Menu **Como Usar** **Conceitos**
Operadores **Desafios** **Ajuda**

Operadores

Esses são os operadores aritméticos

Figura 15. Operadores. Fonte: Autores (2017).

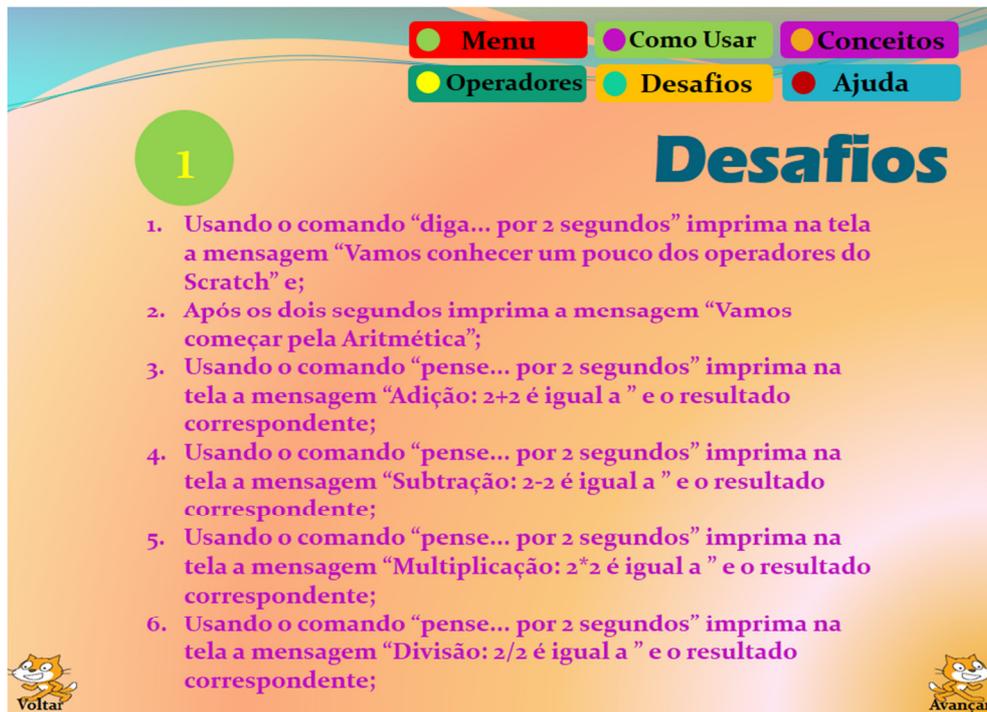


Figura 16. Desafios. Fonte: Autores (2017).



Figura 17. Ajuda. Fonte: Autores (2017).

3.3.1. Desafios

Abaixo são apresentados os desafios propostos no tutorial interativo. Dessa forma, mostra-se a seguir o comando solicitado, o que se espera do usuário ao tentar construir o código, bem como o código final esperado.

3.3.1.1. Desafio 1

Para cumprir este desafio será apresentada a animação “Gato E O Dragão⁷” (Figura 18). Esta, como vários outros projetos, está disponível no site do Scratch, podendo ser remixada, isto é, alterada.

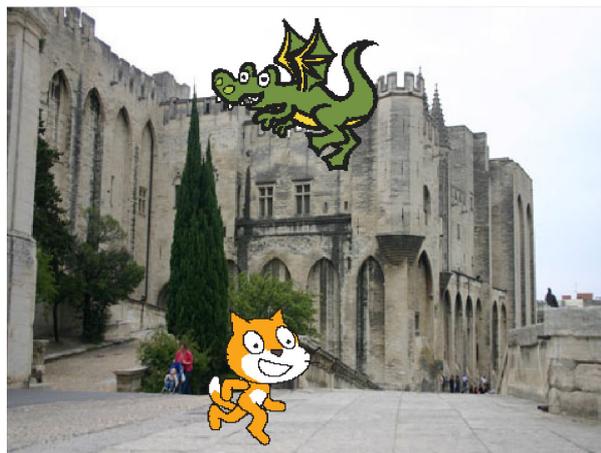


Figura 18. O “Gato E O Dragão”. Fonte: <https://scratch.mit.edu/projects/90082347>.

Essa animação contém dois panos de fundo, duas fantasias, dois atores e dois scripts (Figura 18). Quando o usuário clica na bandeirinha, ambos os atores (o gato e o dragão) irão se movimentar pelos panos de fundo.



Figura 19. A esquerda vê-se o script utilizado para o ator “gato” e a direita para o “dragão”. Fonte: <https://scratch.mit.edu/projects/90082347>.

⁷Gato E O Dragão é uma animação que pode ser encontrada em <<https://scratch.mit.edu/projects/90082347>> produzida pelo usuário DanielBR. . Acesso em 14 de Agosto de 2017.

O gato, quando se clica na bandeira verde, ele se movimenta dez passos e as duas fantasias se alternam num período de 0.2 segundos dando o efeito de movimento. Caso o gato, ao se movimentar, toque na borda, ele retornará.

O dragão também, após o usuário clicar na bandeira verde, trocará de fantasia, sendo que em uma delas ele cuspirá fogo. Esse ator também andará dez passos e quando tocar na borda ele retornará, fazendo o movimento inverso.

Além disso, o usuário deverá notar que, apesar dos scripts terem bastantes semelhanças, há uma pequena diferença no código.

Após o aluno analisar e explorar essa animação, será apresentado a ele o seguinte desafio: *com base na exploração desta, desenvolva um projeto contendo, no mínimo, mais um ator e crie um diálogo entre os atores.*

3.3.1.2. Desafio 2

Para este desafio, apresenta-se o “Jogo-Peixe⁸”. Este possui 4 atores - sendo 3 peixes e um tubarão (Figura 20) - uma variável chamada “PEIXES” e 4 scripts. Destes, um script é utilizado pelo ator tubarão e outro script por cada um dos três peixes (Figura 20). Durante este jogo os peixes surgem no cenário e se movimentam de forma aleatória. Por sua vez o tubarão é controlado pelo mouse do jogador.



Figuras 20. Jogo-Peixe. Fonte: <https://scratch.mit.edu/projects/12936663>.

Quando o usuário clica na bandeirinha o jogo é iniciado e o ator tubarão é exibido com sua primeira fantasia (de boca fechada). Logo após a variável “PEIXES” é inicializada com zero (esta variável armazenará o número de peixes consumidos pelo tubarão).

⁸Jogo-Peixe – produzido pelo usuário “gqsscratch”, este jogo é uma “Iniciativa Computação na Escola” (<http://www.computacaonaescola.ufsc.br>). O mesmo é uma adaptação do jogo publicado na história *Bit and Bytes - Go Fishing with Scratch* (www.askmagkids.com). Está disponível em <<https://scratch.mit.edu/projects/12936663>>. Acesso em 14 de Agosto de 2017.

Em seguida o tubarão passa a seguir o ponteiro do mouse, onde por meio deste o usuário pode conduzir o tubarão para que se alimente da maior quantidade de peixes possível. Quando o tubarão toca na cor “laranja” (indica que ele tocou em um peixe, pois estes são desta cor), o tubarão muda de fantasia (para a fantasia de boca aberta), espera um segundo e em seguida volta para a fantasia inicial (boca fechada).

Também no momento na inicialização do jogo (ao clicar na bandeirinha) os três peixes serão mostrados sobre o pano de fundo. Em seguida eles apontarão para a direção de 90 graus, movimentarão 3 passos e irão fazer um giro de 20 graus negativos a 20 graus positivos. Vale ressaltar que este giro é gerado aleatoriamente. Durante a movimentação, caso algum peixe toque na borda ele irá voltar.

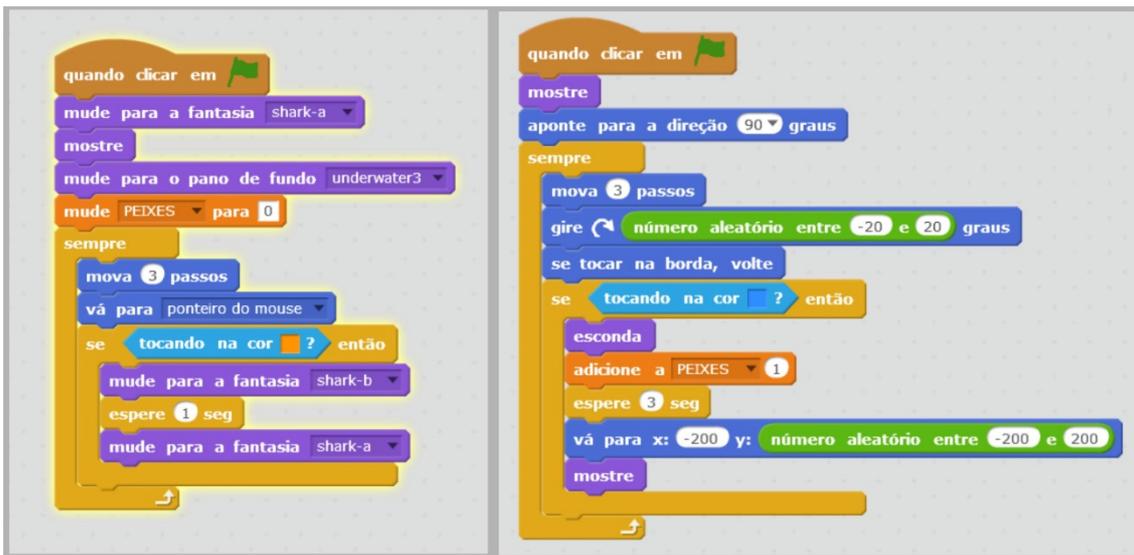


Figura 21. A esquerda vê-se o script utilizado para o ator “tubarão” e a direita o script utilizado nos três atores “peixes”. Fonte: <https://scratch.mit.edu/projects/12936663>.

Durante o jogo, caso algum peixe toque na cor azul (indicando que ele tocou no tubarão, pois o mesmo possui esta cor) este peixe irá desaparecer e a variável “PEIXES” será incrementada com 1, indicando que mais um peixe foi consumido pelo tubarão. Em seguida espera-se 3 segundos e um novo peixe irá surgir em uma posição gerada aleatoriamente, fazendo com que sempre haja peixes sobre o pano de fundo para que o jogo possa continuar.

Após o aluno analisar e explorar essa animação, ele terá visto como criar scripts para vários atores e lidar com várias fantasias. Ademais, o usuário terá noção de criação de números aleatórios, especialmente de posicionamento no plano cartesiano e de como lidar com incremento de variáveis. Diante disto é apresentado seguinte desafio: *com base na exploração do Jogo-Peixe, desenvolva um projeto, com no mínimo dois atores, tendo um critério de pontuação, uma variável para incrementar os pontos feitos e um critério de parada que indique que o usuário venceu o jogo.*

3.3.1.3. Desafio 3

1. Usando o comando “diga... por 2 segundos” imprima na tela a mensagem “Vamos conhecer um pouco dos operadores do Scratch” e;

2. *Após os dois segundos imprima a mensagem “Vamos começar pela Aritmética”;*
3. *Usando o comando “pense... por 2 segundos” imprima na tela a mensagem “Adição: $2+2$ é igual a” e o resultado correspondente;*
4. *Usando o comando “pense... por 2 segundos” imprima na tela a mensagem “Subtração: $2-2$ é igual a” e o resultado correspondente;*
5. *Usando o comando “pense... por 2 segundos” imprima na tela a mensagem “Multiplicação: $2*2$ é igual a” e o resultado correspondente;*
6. *Usando o comando “pense... por 2 segundos” imprima na tela a mensagem “Divisão: $2/2$ é igual a” e o resultado correspondente;*

Na execução deste desafio, espera-se que o usuário utilize os quatro operadores aritméticos (adição, subtração, divisão e multiplicação) para montar seu código, além de usar os comandos “diga... por 2 segundos” e “pense... por 2 segundos”. O código esperado é apresentado na Figura 1822.

Ao utilizar os operadores aritméticos do Scratch, o usuário conhecerá outra forma – diferente, dinâmica e mais atrativa – de como se pode trabalhar com as operações básicas. Poderá ser percebido ainda que o símbolo da adição (+) e da subtração (-) são os mesmos que aprendemos na escola, mas os da multiplicação e divisão ganham outra simbologia, respectivamente: (*) e (/).

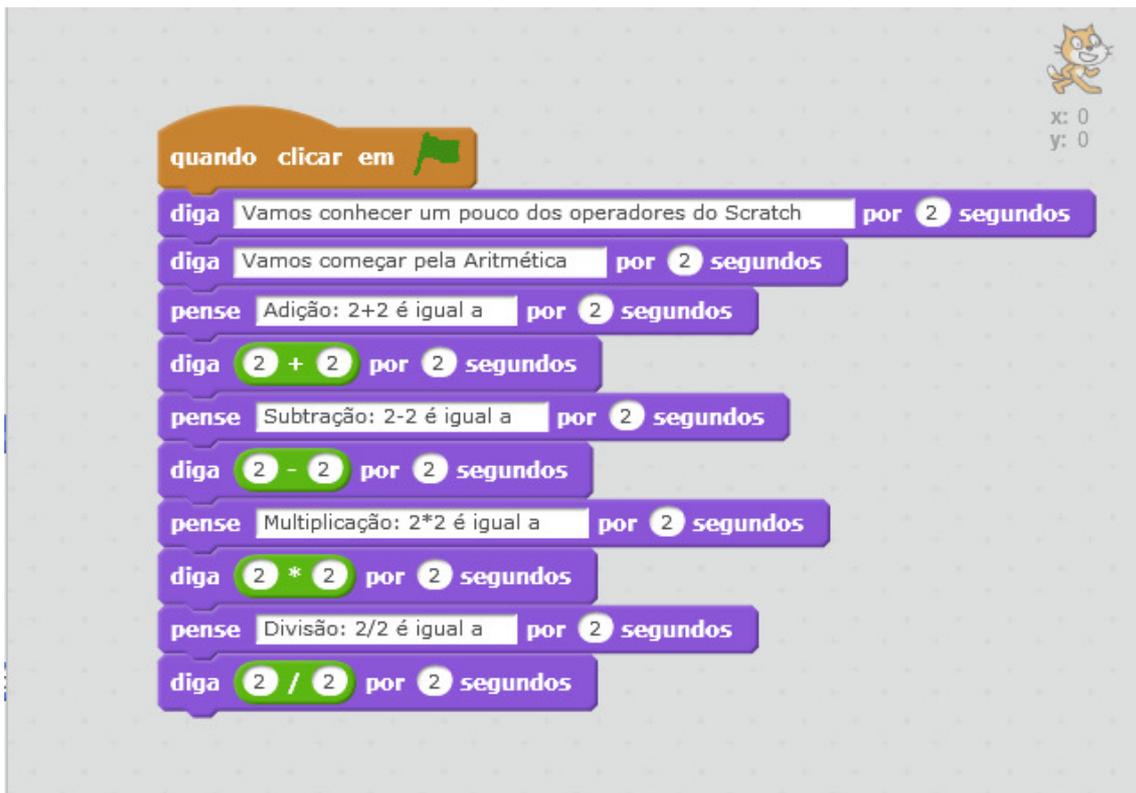
Após o término do desafio por parte de todos os participantes, deverá ser mostrado o código esperado, dando assim, a oportunidade de se verificar se o código do usuário está de acordo com o resultado esperado ou se necessita de adequações. Nesse processo de comparação entre o código que foi feito e o que era esperado, o participante, além de sanar possíveis dúvidas, poderá assimilar com mais facilidade e entender como se pode trabalhar com os operadores no Scratch.

3.3.1.4. Desafio 4

Usando os operadores do Scratch, faça a Tabela Verdade da Conjunção.

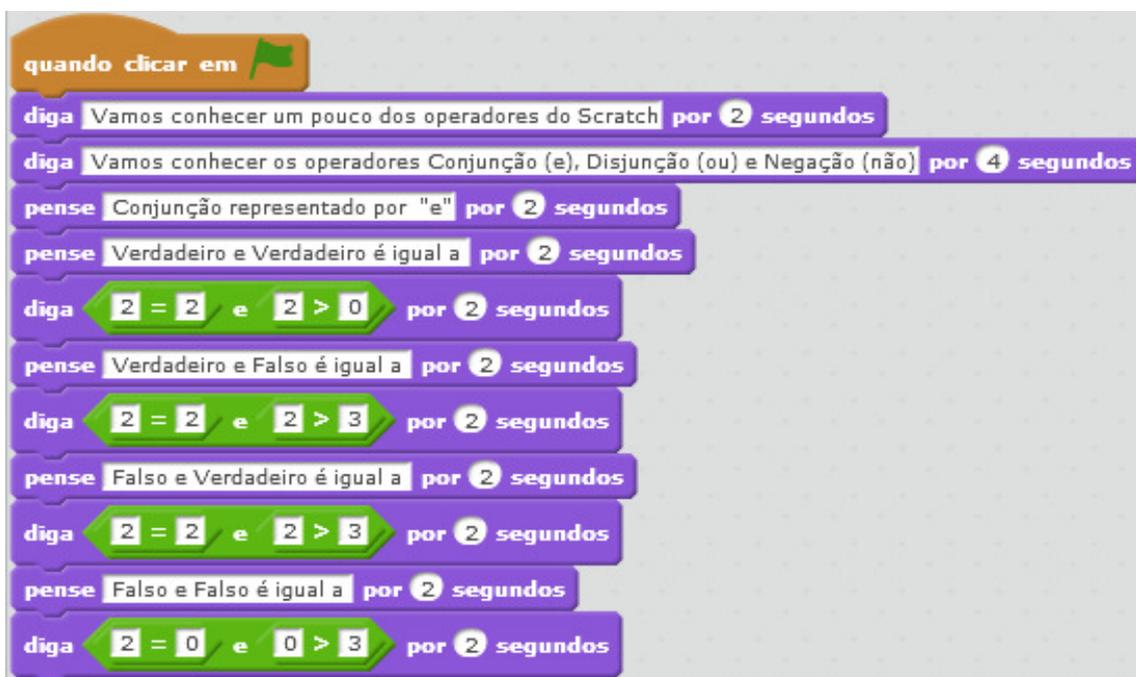
Nesse desafio é solicitado que seja feita a tabela verdade da conjunção, usando para isso o operador “e”, além dos operadores relacionais “menor que”, “maior que” e “igual”. Também serão usados os comandos “Diga... por... segundos” e “Pense... por... segundos”. O código esperado encontra-se na Figura23.

Ao utilizar os operadores relacionais do Scratch, o usuário trabalhará de uma forma mais lúdica quais valores são iguais, ou qual valor é maior ou menor que o outro.



The code starts with a 'quando clicar em' (when clicked) event block. It contains a sequence of 'diga' (say) and 'pense' (think) blocks for 2 seconds each. The blocks are: 'Vamos conhecer um pouco dos operadores do Scratch', 'Vamos começar pela Aritmética', 'Adição: 2+2 é igual a', '2 + 2', 'Subtração: 2-2 é igual a', '2 - 2', 'Multiplicação: 2*2 é igual a', '2 * 2', 'Divisão: 2/2 é igual a', and '2 / 2'. A Scratch logo and coordinates (x: 0, y: 0) are visible in the top right corner.

Figura 18. Código do Desafio 3. Fonte: Autores (2017).



The code starts with a 'quando clicar em' (when clicked) event block. It contains a sequence of 'diga' (say) and 'pense' (think) blocks. The blocks are: 'Vamos conhecer um pouco dos operadores do Scratch' (2s), 'Vamos conhecer os operadores Conjunção (e), Disjunção (ou) e Negação (não)' (4s), 'Conjunção representado por "e"' (2s), 'Verdadeiro e Verdadeiro é igual a' (2s), '2 = 2 e 2 > 0' (2s), 'Verdadeiro e Falso é igual a' (2s), '2 = 2 e 2 > 3' (2s), 'Falso e Verdadeiro é igual a' (2s), '2 = 2 e 2 > 3' (2s), 'Falso e Falso é igual a' (2s), and '2 = 0 e 0 > 3' (2s).

Figura 23. Código do Desafio 4. Fonte: Autores (2017).

Aliando os operadores relacionais aos lógicos, poderá ser facilmente elaborado o código sobre a tabela verdade da conjunção representada pelo símbolo “e”.

No fim desse desafio, deverá ser mostrado o código esperado, para que os participantes tenham a oportunidade de observar se o código produzido pelo usuário está de acordo com o resultado esperado ou não. Após essa verificação, o usuário terá aprendido como pode utilizar os operadores relacionais e lógicos, além de aprender sobre a tabela verdade da conjunção.

3.3.1.5. Desafio 5

Usando os operadores do Scratch, faça a Tabela Verdade da Disjunção.

Nessa etapa propõe-se que seja feita a tabela-verdade da disjunção. Para isso deverá ser utilizado o operador “ou”, além dos operadores relacionais “menor que”, “maior que” e “igual”. Ainda deverão ser usados os comandos “Diga... por... segundos” e “Pense... por... segundos”. O código esperado encontra-se na Figura .

Ao utilizar os operadores relacionais do Scratch, o usuário trabalhará de uma forma mais dinâmica quais valores são iguais, ou qual valor é maior ou menor que o outro. Ao se trabalhar juntamente os operadores relacionais e lógicos poderão ser facilmente feito o código sobre a tabela verdade da disjunção representada pelo símbolo “ou”.

Na conclusão desse desafio, deverá também ser mostrado o código esperado, para que os participantes tenham a oportunidade de analisar se o código produzido por eles está em conformidade com o resultado esperado ou não. Posteriormente a essa análise, o usuário terá entendido como pode utilizar os operadores relacionais e lógicos, além de aprender sobre a tabela verdade da disjunção.

3.3.1.6. Desafio 6

Usando os operadores do Scratch, faça a Tabela Verdade da Negação.

Já nessa fase o objetivo é a elaboração da tabela verdade da negação, onde deverão ser usados os operadores relacionais (“menor que”, “maior que” e “igual”) e lógico (“não”), além de outros blocos de código. O resultado esperado é apresentado na Figura 19.

Trabalhando com operadores relacionais do Scratch, o usuário poderá entender de uma forma melhor quais valores são iguais, ou qual valor é maior ou menor que o outro. Ao unir os operadores relacionais e lógicos em determinado código, poderá ser feita a representação da tabela verdade da negação, representada pelo símbolo “não”, n ambiente Scratch.

No final desse desafio, deverá também ser apresentado o código esperado, para que os participantes tenham a oportunidade de analisar se o código deles cumpre ou não com o objetivo do desafio. Em seguida a essa análise, o usuário terá observado como pode utilizar os operadores relacionais e lógicos, além de aprender sobre a tabela verdade da negação.

```
quando clicar em [bandeira verde]
diga [Vamos conhecer um pouco dos operadores do Scratch] por 2 segundos
diga [Vamos conhecer o operador Disjunção (ou)] por 4 segundos
pense [Disjunção representado por "ou"] por 2 segundos
pense [Verdadeiro ou Verdadeiro é igual a] por 2 segundos
diga [2 = 2 ou 2 > 0] por 2 segundos
pense [Verdadeiro ou Falso é igual a] por 2 segundos
diga [2 = 2 ou 2 > 3] por 2 segundos
pense [Falso ou Verdadeiro é igual a] por 2 segundos
diga [2 > 3 ou 2 = 2] por 2 segundos
pense [Falso ou Falso é igual a] por 2 segundos
diga [2 = 0 ou 0 > 3] por 2 segundos
```

Figura 24. Código do Desafio 5. Fonte: Autores (2017).

```
quando clicar em [bandeira verde]
diga [Vamos conhecer um pouco dos operadores do Scratch] por 2 segundos
diga [Vamos conhecer o operador Negação (não)] por 4 segundos
pense [Negação representado por "não"] por 2 segundos
pense [Não (Verdadeiro) é igual a] por 2 segundos
diga [não 2 = 2] por 2 segundos
pense [Não (Falso) é igual a] por 2 segundos
diga [não 2 = 3] por 2 segundos
```

Figura 195. Código do Desafio 6. Fonte: Autores (2017).

3.3.1.7. Desafio 7

Use os operadores relacionais “maior que (>)”, “menor que (<)” e “igual (=)” para criar um jogo, animação ou história no Scratch com, no mínimo: dois atores, dois cenários e nove linhas de código.

Esse desafio sugere que sejam utilizados operadores os relacionais maior que (>), menor que (<) e igual (=), entre outros blocos de código. O código esperado é apresentado na figura 26.

Ao montar um código utilizando operadores relacionais do Scratch, o participante poderá assimilar de uma forma inovadora quais valores são iguais, ou qual valor é maior ou menor que o outro.

Após a produção do código proposto no desafio, deverá também ser socializado o código esperado, para que os participantes possam perceber se o código deles cumpre ou não com o objetivo do desafio. Depois disso, o usuário terá compreendido como pode utilizar os operadores relacionais do Scratch.

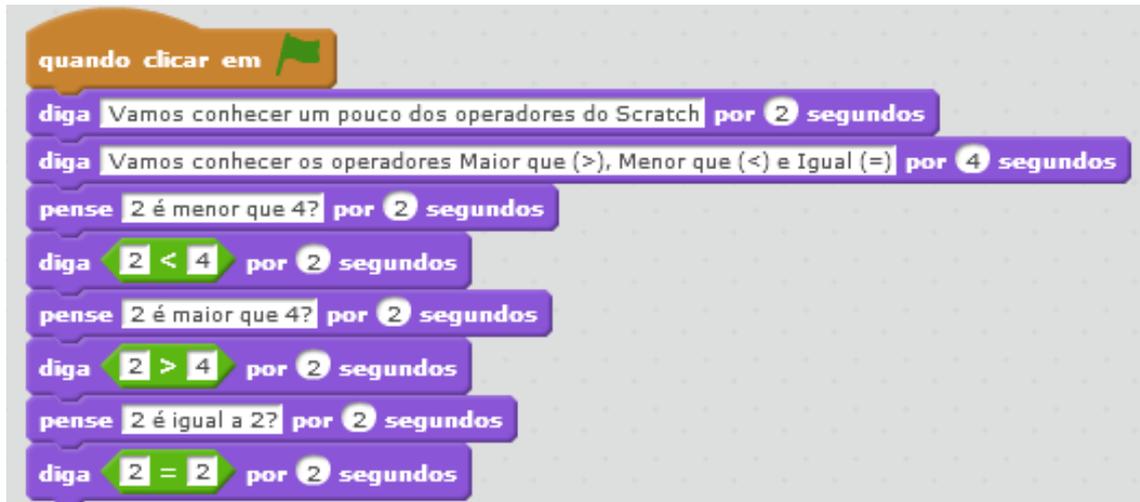


Figura 206. Código do Desafio 7. Fonte: Autores (2017).

3.3.1.8. Desafio 8

Use os operadores relacionais, lógicos e aritméticos para criar um jogo, animação ou história no Scratch com no mínimo dois atores, dois cenários e quinze linhas de código.

Nessa atividade o usuário deverá usar sua criatividade, inventividade e cognição para desenvolver seu código.

Ao montar um código utilizando diversos blocos de comando, o participante poderá explorar de uma forma inovadora e dinâmica as diversas funcionalidades do Scratch.

Após a produção do código proposto no desafio, este deverá ser socializado entre os demais participantes para que se possa coletivamente analisar e discutir os códigos produzidos, bem como observar a criatividade e inventividade empregada na construção dos códigos.

3.4. Considerações finais

Na elaboração deste trabalho verificou-se a importância do pensamento computacional, como sendo uma habilidade fundamental para as pessoas, na nova dinâmica social que se apresenta no século XXI.

Esta habilidade não se refere à capacidade de uma pessoa utilizar um computador para acessar internet ou digitar um texto, por exemplo, mas sim de utilizar o computador como ferramenta de propulsão de sua capacidade cognitiva, a fim de utilizar-se do potencial computacional, existente atualmente, na busca por soluções de problemas ou mesmo em benefício da inventividade e de novas pesquisas científicas.

Apesar desta grande relevância do pensamento computacional, sendo inclusive - na concepção de Blikstein (2008) - indispensável para o exercício da cidadania no século presente, no Brasil ele ainda não integra a base curricular da educação básica, fato que deixa nosso país muito atrás de outras nações que já incluíram o pensamento computacional nas suas grades curriculares.

Diante disto, ensinar o pensamento computacional nas escolas brasileiras ainda pode ser um desafio, cita-se, por exemplo, casos de docentes/pesquisadores que para ensinarem precisam utilizar do espaço concedido a outra disciplina, como por exemplo, informática.

Todavia este desafio pode ser superado quando verifica-se que o Pensamento Computacional, de certa forma, alia-se à multidisciplinaridade, sendo possível então utilizá-lo para o ensino em diversas áreas. Além disso, o pensamento computacional também pode ser aplicado sem a utilização de computadores numa abordagem chamada de “computação desplugada”. Esta consiste em ensinar os princípios da computação sem fazer uso, necessariamente de computadores. Em vez disto, pode-se utilizar de materiais didáticos diversos, tais como recortes de papel, peças teatrais e outros. Na concepção de Vieira, Passos e Barreto (2013) este método pode ser uma maneira de desmistificar, e tornar menos complexo, o entendimento dos conceitos que envolvem a computação.

Além disso, é possível facilitar a disseminação do pensamento computacional utilizando ferramentas que auxiliem neste processo. Dentre essas ferramentas destacam-se os ambientes visuais de programação, com alto nível de abstração, que facilitam ensino-aprendizagem de usuários que ainda não dominam nenhuma linguagem de programação. Uma pesquisa realizada por Bombasar (2015), no período de 2006 a 2015 constatou-se que as linguagens de programação visuais, ou VLPs (*Visual Programming Languages*) se destacaram entre as principais ferramentas, utilizada no ensino do Pensamento Computacional para a educação básica.

Dentre estes ambientes visuais destaca-se o Scratch, que se utiliza do método de programação em blocos, é código aberto, tem uma vasta documentação e por estes motivos foi a ferramenta utilizada neste trabalho.

O Scratch é uma ferramenta desenvolvida no MIT, bastante difundida no mundo, mas pouco utilizada no Brasil. Vale ressaltar a importância que mais educadores

e pesquisadores em geral possam fazer uso dessa ferramenta que apresenta muitas possibilidades de aplicações no ensino.

A empregabilidade do Scratch no ensino do Pensamento Computacional pode ser justificada devido esta ferramenta direcionar o usuário para o desenvolvimento de seus projetos em um ciclo contínuo (conforme mostrado na Figura 2), onde o usuário cria um projeto, corrige quando o funcionamento não ocorre como o esperado, compartilha (seja com amigos ou mesmo no website do Scratch), recebe sugestões de melhorias e este projeto pode ser corrigido, melhorado ou dar origem à novos projetos.

Este ciclo correlaciona-se com o ciclo de ações *descrição-execução-reflexão-depuração* descrito por Valente (2005), sendo este capaz de oferecer resultados que podem ampliar o conhecimento a cada ciclo. Tais ciclos, tanto o do Scratch quanto o relatado por Valente (2005) coadunam com o entendimento de Bombasar (2015 *apud* CSTA 2011) sobre o Pensamento Computacional como sendo um processo de resolução de problemas e que apresenta nove conceitos que podem ser utilizados durante este processo: *coleta de dados, análise de dados, representação de dados, decomposição de problemas, abstração, algoritmos, automação, simulação e paralelização*. Neste sentido, utilizando o Scratch como aliado do ensino do Pensamento Computacional é possível explorar praticamente todos estes conceitos, dependendo unicamente da proposta de ensino e metodologia utilizada pelo docente/pesquisador.

Dentre as muitas possibilidades, o Scratch pode auxiliar no ensino da lógica e colaborar para o desenvolvimento do pensamento computacional, bastando para isso que o aluno seja estimulado através de desafios e exercícios que utilizam os operadores lógicos, aritméticos e relacionais.

O Scratch é uma ferramenta que pode ser utilizada para auxiliar no processo de ensino-aprendizagem tanto na matemática quanto em diversas em outras áreas. Verificou-se que a lógica matemática pode ser ensinada com o apoio do ambiente Scratch, utilizando para isso, os operadores do Scratch: lógico, aritmético e relacional. Esses operadores permitem ao usuário do ambiente diversas formas de criação de códigos. Os usuários podem utilizar também os operadores aritméticos para aprender ou ensinar sobre as operações básicas. Os operadores relacionais possibilitam o aprendizado sobre os valores “igual”, “maior que” ou “menor que”. Já os operadores lógicos permitem, por exemplo, ensinar e aprender sobre a conjunção “e”, disjunção “ou” e negação “não”, além de ser possível codificar no Scratch as suas tabelas verdades.

Como trabalhos futuros pode-se citar o uso do Scratch em outros cenários de ensino-aprendizagem como em outras disciplinas como a literatura, física, biologia, química, dentre outras.

Referências

- Blikstein, P. (2008). “O pensamento computacional e a reinvenção do computador na educação”.
- Bombasar, J. et al.. Ferramentas para o ensino-aprendizagem do pensamento computacional: onde está alan turing?. In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE). 2015. p. 81.

- De Alencar Filho, E. (2002). Iniciação à lógica matemática. NBL Editora.
- Ferreira, J. C. (2011). Elementos de Lógica Matemática e Teoria dos Conjuntos.
- Freire, P. (1987). “Pedagogia do oprimido”. 17ª. Ed. Rio de Janeiro: Paz e Terra, 3.
- França, R. S., Ferreira, V. F. S., Almeida, L. D., and Amaral, H. D. (2014). “A disseminação do pensamento computacional na educação básica: lições aprendidas com experiências de licenciandos em computação”. In Anais do XXII Workshop sobre Educação em Computação (WEI-CSBC).
- França, R., and Tedesco, P. (2015). “Desafios e oportunidades ao ensino do pensamento computacional na educação básica no Brasil”. In Anais dos Workshops do Congresso Brasileiro de Informática na Educação (Vol. 4, No. 1, p. 1464).
- Lifelong kindergarten group. Scratch: Imagine, Program, Share. MIT MediaLab, 2007. Disponível em: <http://Scratch.mit.edu/>. Acesso em 01 maio 2017.
- Omine, E. H. (2014) “Design gráfico computacional: computação aplicada no projeto e na produção de imagens dinâmicas e interativas” (Doctoral dissertation, Universidade de São Paulo).
- Scaico, P. D., de Lima, A. A., da Silva, J. B. B., Azevedo, S., Paiva, L. F., Raposo, E. H. S., ... and Mendes, J. P. (2012). “Programação no ensino médio: uma abordagem de ensino orientado ao design com Scratch”. In Anais do Workshop de Informática na Escola (Vol. 1, No. 1).
- Sobreira, E. S. R., Takinami, O. K., and dos Santos, V. G. (2013). “Programando, Criando e Inovando com o Scratch: em busca da formação do cidadão do século XXI”. Jornada de Atualização em Informática na Educação, 1(1).
- Souza, J. (2015). Lógica para ciência da computação. Elsevier Brasil.
- Valente, J. A. (2005). A espiral da espiral de aprendizagem: o processo de compreensão do papel das tecnologias de informação e comunicação na educação. Campinas, SP: [s.n.].
- Vieira, A; Passos, O; Barreto, R. Um relato de experiência do uso da técnica computação desplugada. Anais do XXI WEI, p. 670-679, 2013.
- Wing, J. M. (2006). “Computational thinking”. Communications of the ACM, 49(3), 33-35.

Sobre os autores

Gilson Pedroso dos Santos

Mestrando em Educação, Linha de Pesquisa 2 - Práticas Educativas, Linguagens e Tecnologias, pela Universidade Federal do Oeste do Pará, bacharel em Sistemas de Informação pela Universidade Federal do Oeste do Pará (2016) e licenciado em Letras pela Universidade Luterana do Brasil. Atualmente pesquisa sobre Pensamento Computacional e TIC's no ensino da matemática. Integra o Grupo de Estudos e Pesquisas em Educação Matemática e Interdisciplinaridade na Amazônia. Tem experiência com a construção de objetos de aprendizagem e na utilização de tecnologias para auxiliar no processo de ensino-aprendizagem.

Ronilson dos Santos Bezerra

Possui graduação em Ciência da Computação pela Universidade Federal do Oeste do Pará (Ufopa). Servidor Público Federal, atuando desde 2012 na Coordenação Acadêmica do Instituto de Ciências da Sociedade – Ufopa. Possui interesse principalmente em áreas de Informática na Educação, Pensamento Computacional, Inclusão Digital, Modelagem de dados, Programação Web e Robótica.