

Capítulo

3

Desenvolvimento do Pensamento Computacional através do Uso de Tutoriais Interativos

Gilson Pedroso dos Santos, Ronilson dos Santos Bezerra

Resumo

Este trabalho discute sobre o Pensamento Computacional (PC), sua introdução no processo de ensino-aprendizagem e sua relevância para o indivíduo do século XXI. Apresenta também duas ferramentas que podem ser muito importantes, porque podem auxiliar no processo educativo visando a disseminação do Pensamento Computacional. Uma dessas ferramentas denomina-se Tutorial Interativo (TI), recurso computacional que objetiva facilitar a compreensão do conteúdo pelo receptor. A outra ferramenta é um software de programação em blocos denominado Scratch. Esta já é popularizada em mais de 150 países e possui mais de 28 milhões de usuários. Por ser embasada nos princípios do construcionismo, esta pode ser utilizada nas mais diversas atividades pedagógicas, desenvolvendo no aluno, a capacidade do pensar computacionalmente. Para a elaboração deste trabalho, foi feita uma minuciosa pesquisa bibliográfica sobre o assunto em tela, sob a ótica de vários pesquisadores e percebeu-se que todos convergem para uma necessidade urgente: a necessidade do cidadão do século XXI aprender a utilizar a tecnologia como sua aliada, de forma consciente e pensando computacionalmente.

Palavras-chave: Tutoriais Interativos. Scratch. Pensamento Computacional. Ensino.

Abstract

The Computational Thinking (CP), its introduction to the teaching-learning process, as well as its relevance to the 21st century individual is what is being discussed in this paper. It also presents two learning tools which can be very important as they facilitate the educational process in order to disseminate Computational Thinking skills. One of these tools is called Interactive Tutorial (IT), a computational resource whose aim is help the receiver to understand the learning material sent to him/her. The other tool is a blocks (drag and drop) programming software called Scratch. Based on the constructivism, the latter has already been popularized in more than 150 countries and has more than 28 million users as is it can be used in several educational activities

which lead students to develop computational thinking skills. To accomplish this work, a detailed literature survey was conducted on the reported subject according to the point of view of various researchers and, as a result, it was noticed these researchers are all convergent to an urgent need: the 21st individual's need to learn how to use technology to self-improvement in a conscious way and thinking computationally.

Keywords: Interactive Tutorials. Scratch. Computational Thinking. Teaching.

1. Introdução

O presente trabalho trata sobre a utilização da ferramenta Scratch e de Tutoriais Interativos (TI) no processo de ensino-aprendizagem.

O mesmo busca discutir ainda o desenvolvimento do Pensamento Computacional (PC) aliado aos TI's e como elaborar estas ferramentas através de programas como *Microsoft Office Power Point* e *Microsoft Paint*. A importância deste se reflete na necessidade de se investigar as possibilidades de aplicações do PC na educação. Para auxiliar nessa discussão é apresentado um *template* contendo as principais partes que um tutorial interativo precisa ter.

Como procedimentos metodológicos utilizados, destacam-se as revisões bibliográficas sobre Scratch, Tutoriais Interativos, bem como sobre o Pensamento Computacional e TIC. A partir disso, foi elaborado e documentado o *template* para basilar a produção de Tutoriais Interativos. É apresentada ainda a proposta de desenvolvimento de um TI através de desafios.

A seguir é informado como está estruturado o presente trabalho.

Na seção 2 é apresentado o referencial teórico, em três subseções, onde: Na subseção 2.1 consta uma explanação sobre o PC, sob a ótica de vários pesquisadores, e sua aplicação no processo de ensino-aprendizagem; Em seguida, na subseção 2.2, é exposto o conceito de TI e sua importância no processo da comunicação e na subseção 2.3 são discutidos sobre o Scratch e sua aplicabilidade no processo educativo.

Na seção 3 foi abordado sobre as TIC no processo de ensino-aprendizagem, importância do professor para a apropriação dos recursos computacionais, dentre outros.

Já na seção 4 é evidenciada a construção de um tutorial interativo e foi abordado ainda sobre o processo de atribuição de interatividade à ferramenta.

Apresenta-se na seção 5 uma proposta de desenvolvimento de um tutorial interativo através da execução de desafios.

Por fim, na seção 6, os autores teceram suas considerações finais sobre este trabalho.

2. Referencial Teórico

2.1. Pensamento Computacional

Esta seção aborda sobre o Pensamento Computacional, bem como sobre a sua aplicação no processo de ensino-aprendizagem.

Nas últimas décadas, os dispositivos eletrônicos passaram a fazer parte do cotidiano das pessoas. Se antes os computadores eram restritos às forças militares,

universidades e grandes empresas, atualmente eles estão presentes em todos os lugares. Com isso, a computação, enquanto ciência, foi ganhando espaço na vida das pessoas e por mais que estas não tenham conhecimento sobre esta área, diariamente, elas fazem uso de algum dispositivo computadorizado, seja um caixa eletrônico, celular, *tablet*, computador de bordo, *desktops*, *notebooks* e até mesmo nos eletrodomésticos.

Este acesso diário aos computadores, somado à popularização da internet, mudou completamente a forma como as pessoas obtêm informações. Se antes as informações passavam primeiro pelas redações dos jornais impressos e apenas horas ou dias depois eram publicadas, hoje as pessoas têm acesso às informações quase que de forma instantânea. Ademais, com o advento da *Web 2.0*¹, além de consumidoras de informações, estas pessoas também passaram a atuar na produção e compartilhamento de conteúdo.

Além disso, a geração atual se depara, cotidianamente, com a necessidade de saber lidar com soluções computacionais, uma vez que elas estão presentes nas mais diversas áreas. Desse modo, estar familiarizado com essas soluções pode facilitar ou não a inserção de uma pessoa no mercado globalizado atual.

Diante da ascensão do uso dessas tecnologias, tornou-se imprescindível a discussão sobre a inserção das mesmas no processo de ensino-aprendizagem. Desde então o uso das Tecnologias da Informação e da Comunicação (TIC) ganhou mais destaque e tornou-se objeto de estudo de diversos pesquisadores. E neste sentido é necessário fazer uma distinção entre o ensino da informática - ensino restrito ao “adestramento” para uso de alguns aplicativos, tais como editores de texto, acesso à internet e outros - do ensino da computação enquanto ciência. Na concepção de França et al. (2014 apud Nunes, 2008) esta diferenciação é imprescindível.

Ao abordar sobre a computação no processo educativo, Geraldes (2017 apud CSTA, 2005) destaca que a computação enquanto ciência não se refere apenas às habilidades de apontar e clicar, mas sim de utilizar um conjunto de fundamentos da computação na resolução sistemática de problemas do mundo real. O que segundo o autor é tão essencial ao estudante do presente século, quanto as ciências tradicionais.

Blikstein (2008) complementa destacando que atualmente não basta apenas saber somar, subtrair, ler e escrever. Segundo o autor, para exercer plenamente a cidadania no século XXI, faz-se necessário também a obtenção de diversos outros conhecimentos/habilidades. E dentre estes o autor destaca o “Pensamento Computacional”, que segundo Wing (2006) é uma habilidade indispensável para todos e não apenas para profissionais da área da computação. A referida autora define o Pensamento Computacional como algo muito além da capacidade de saber utilizar um aplicativo de computador e sim como uma habilidade de saber utilizar o computador como um instrumento de aumento do poder cognitivo e operacional humano, o que é capaz de ser convertido em resultados como aumento da produtividade, inventividade e criatividade humana.

¹ Considerada “a segunda geração de serviços *on-line* que se caracteriza por potencializar as formas de publicação, compartilhamento e organização de informações, além de ampliar os espaços para a interação entre os participantes do processo” (PRIMO, 2007, p. 1).

Apesar do pioneirismo de Wing, Seymour Papert também teve uma influência significativa no desenvolvimento do Pensamento Computacional. O trabalho de Papert sobre o Construcionismo e a linguagem de programação LOGO foi fundamental para o desenvolvimento do Pensamento Computacional, uma vez que nele foi apresentado o computador como instrumento capaz de apoiar a construção do conhecimento. Trabalhou-se ainda na hipótese de que a programação de computadores poderia facilitar o processo de resolução de problemas e o aprendizado em diversas áreas [Rodrigues 2015].

Na concepção de Blikstein (2008) o Pensamento Computacional pode ser definido como “saber usar o computador como um instrumento de aumento do poder cognitivo e operacional humano”. Já Furber (2012) preconiza o Pensamento Computacional como o processo de reconhecimento dos aspectos computacionais no mundo que nos cerca, além da aplicação de ferramentas e técnicas da Ciência da Computação para compreender e raciocinar sobre os sistemas e processos naturais e artificiais. Assim, o Pensamento Computacional não deve ter necessariamente como resultado final a produção de *software* e *hardware*. Reconhece ainda que os conceitos fundamentais da Computação estarão presentes para resolver problemas em vários contextos do cotidiano [Barcelos e Silveira 2012].

Para Blikstein (2008) o “pensar computacionalmente” pode ser dividido em duas etapas: a) Identificação das tarefas cognitivas que podem ser executadas com mais rapidez e eficiência fazendo o uso da computação; b) Programação do computador para que este realize as tarefas cognitivas identificadas na etapa anterior. Desse modo, os computadores passam a auxiliar na execução dos processos produtivos, pesquisas científicas, dentre outros, de forma a evitar perda de tempo, por parte das pessoas, com a execução de tarefas repetitivas.

2.1.1 Habilidades do Pensamento Computacional

Segundo Barr e Stephenson (2011) compõe a parte central do Pensamento Computacional nove conceitos que são: Coleta de dados; Análise de dados; Representação de dados; Decomposição de problemas; Abstração; Algoritmos e Procedimentos; Automação; Paralelização e Simulação.

Os mesmos autores também pontuam o que significa cada um desses nove conceitos:

- **Coleta de dados:** pressupõe um processo de coleta de dados ou informações sobre determinado problema de forma adequada;
- **Análise de dados:** é um processo onde se atribui significação aos dados, obtêm-se padrões, além de se tirar conclusões;
- **Representação de dados:** processo onde se representa e se organiza dados através de gráficos, tabelas, imagens, textos ou figuras;
- **Decomposição de problemas:** processo no qual se decompõe certo problema complexo em tarefas menores e mais facilmente gerenciáveis;
- **Abstração:** processo necessário para reduzir a complexidade do problema e para definir ideias principais;
- **Algoritmos e Procedimentos:** sequência de passos necessários para resolução de um problema ou para se atingir determinado objetivo;

- **Automação:** processo que indica a utilização de computadores ou máquinas para realizar tarefas repetitivas;
- **Paralelização:** refere-se à organização de recursos para realizar tarefas, simultaneamente, para alcançar um objetivo;
- **Simulação:** representação ou modelo de um processo. A simulação também envolve experimentos sendo executados usando modelos. A aquisição do PC acontece por meio do estudo de fundamentos de Ciência da Computação, podendo ou não ser integrados a disciplinas de currículos escolares para formação básica.

A CSTA (2011) também apresenta os nove conceitos essenciais para o Pensamento Computacional classificados em três grupos:

- Coleta de dados, análise de dados, representação de dados;
- Decomposição de problemas, abstração, algoritmos;
- Automação, simulação e paralelismo;

Já Karen Brennan e Mitchel Resnick, pesquisadores do MIT - Massachusetts Institute of Technology -, têm realizado nos últimos anos muitos estudos relacionados com o PC no que diz respeito à criação de produtos multimídia. Os resultados dessas pesquisas deram origem ao quadro de referências para estudar e avaliar o desenvolvimento do Pensamento Computacional, e que envolve três dimensões: (i) conceitos computacionais; (ii) práticas computacionais; e (iii) perspectivas computacionais [Sousa e Lencastre 2014].

Brennan e Resnick (2012) também utilizam uma abordagem tripartida para a análise do Pensamento Computacional, contudo, com o foco voltado para práticas de programação. Dessa forma, o framework desenvolvido pelos pesquisadores do Media Lab do MIT é composto pelas seguintes dimensões: conceitos, práticas e perspectivas computacionais.

2.1.1 O que não é o Pensamento Computacional

Ciência da computação é área de conhecimento que estuda a computação, isto é, investiga o que pode ser computado e como pode ser computado. O PC possui diversas características como conceptualização. Entretanto ciência da computação não é sinônimo de programação. Pensar como um cientista da computação envolve mais do que ser capaz de programar um computador [Wing 2006]. Assim, saber programar é uma habilidade dos cientistas da computação e está relacionada como o Pensamento Computacional, todavia o Pensamento Computacional vai muito além disso.

O PC não é saber navegar na internet, enviar e-mail, publicar um blog, operar um processador de texto ou planilha eletrônica. Pensamento Computacional pressupõe a utilização do computador como um instrumento capaz de aumentar o poder cognitivo e operacional humano, ou seja, é usar computadores, e redes de computadores, para expandir a produtividade, inventividade e criatividade [BLIKSTEIN 2008]. Inventar, produzir e criar são ações inerentes ao PC, pois permitem ao indivíduo um protagonismo na sua vida em sociedade. Um indivíduo capaz de inventar coisas novas ou se reinventar como pessoa é um indivíduo mais apto a sobreviver numa época onde mudanças profundas ocorrem a todo o momento.

Saber manusear uma ferramenta de escritório como processador de texto ou planilha, ou até mesmo outros aplicativos não significa que a pessoa está “pensando computacionalmente”, porque o Pensamento Computacional propõe ir além do simples executar algo no computador. Contudo, se a pessoa utiliza o programa computacional, como o processador de texto, por exemplo, pra criar, inventar ou aumentar seu conhecimento, ela passa então a “pensar computacionalmente”.

Pensamento Computacional é uma forma para as pessoas resolverem problemas; não é tentar fazer com que seres humanos pensem como computadores. Computadores são tediosos e enfadonhos; humanos são espertos e imaginativos. Os seres humanos transformam a computação em algo empolgante [Wing 2006].

2.1.3 Pensamento Computacional no ensino

Nos últimos anos os Estados Unidos e alguns países da Europa introduziram o ensino da computação na grade curricular. Todavia, França e Tedesco (2015) observam que o mesmo não acontece ainda no Brasil, onde este ensino é ofertado, mais especificamente, à estudantes das áreas de tecnologias (dos níveis técnico e superior). Com exceção para algumas iniciativas envolvendo ensino médio e fundamental.

Dentre estas iniciativas França e Tedesco (2015) destacam algumas olimpíadas de programação, cursos de lógica de programação fazendo uso do *App Inventor* (www.appinventor.mit.edu), projetos visando o desenvolvimento do raciocínio lógico, peças teatrais e construção de jogos digitais.

No Brasil, nas abordagens utilizadas que visam o desenvolvimento do Pensamento Computacional, utilizando-se a lógica de programação como um meio, destacam-se as ferramentas visuais de programação. Dentre as quais sobressaem-se o Scratch, *App Inventor*, *AgentSheets* e *RoboMind*, conforme destacam França e Tedesco (2015).

Neste trabalho optou-se pelo uso de Tutoriais Interativos e da ferramenta Scratch. Esta, foi escolhida pois, além de ser gratuita, é bastante conhecida e possui uma excelente documentação, o que possibilita aos usuários iniciantes melhor compressão da ferramenta. Além disso, ela funciona tanto *online* quanto *off-line*, facilitando a utilização por usuários que ainda não dispõem de acesso à internet. Por sua vez por meio de Tutoriais Interativos é possível apresentar e ensinar a utilizar este ambiente de forma lúdica, aumentando o interesse pelo conteúdo e facilitando a aprendizagem.

2.2. Tutorial Interativo

Esta seção apresenta a definição do que são Tutoriais Interativos, bem como suas principais características.

Tutorial pode ser definido como um programa que busca instruir, podendo envolver leitura, conexão com outros sítios, perguntas e testes [Francisco e Oliveira 2006]. A instrução através do tutorial visa potencializar o processo de construção do conhecimento. Para tanto, o aprendiz terá que ler e exercitar o conhecimento adquirido. A conexão com outros sítios disponíveis na internet, pode apresentar informações auxiliares que nortearão no processo de ensino-aprendizagem.

Tutorias também são textos que visam a instrumentalização de todo indivíduo que está envolvido em uma atividade que irá exigir determinada prática para atuar em uma área específica [Santiago 2010]. Vale lembrar que eles podem ser utilizados para fins de ensino ou não, e também, nas mais diversas áreas.

São também ensinamentos transmitidos na internet que possuem a finalidade de ensinar como se realiza alguma tarefa através da explicitação do seu desenvolvimento em etapas [Reis, Romão e Leite 2011]. Dessa forma, nota-se o “passo-a-passo”, isto é, a sequência de etapas que é inerente ao tutorial. É importante que a descrição dos passos sejam claros e objetivos, para que o aprendiz, ao tentar reproduzir a sequência, consiga facilmente executá-las.

A sociedade da qual fazemos parte possui um caráter extremamente imediatista e o tutorial visa atender estas demandas desse tipo de sociedade. Ele pode permitir a aquisição de conhecimento de uma maneira mais rápida e lúdica do que por um meio “mais institucionalizado”. O tutorial também só é possível numa configuração sociotécnica como a da cibercultura. Porque é nesta configuração onde tecnologias e demandas sociais resultam em conexão colaborativa e em todas as características do presenteísmo pós-moderno [Reis, Romão e Leite 2011].

Como afirma os autores, a sociedade atual possui diversas singularidades que décadas atrás eram inimagináveis. Dentre estas, pode-se citar o advento e popularização das Tecnologias da Informação e Comunicação (TIC). Dessa forma, hoje encontra-se disponíveis, nos mais variados sítios, recursos computacionais que podem ser utilizados para enriquecer e auxiliar no processo de ensino-aprendizagem.

Portanto, tutorial pode ser considerado como um mecanismo informal de auxílio na aquisição de conhecimento e informações. Destaca-se seu caráter didático e pormenorizado, uma vez que descreve uma sequência de passos.

Os programas tutoriais são uma versão computacional da instrução programada. Eles dispõem de características que não são permitidas no papel como: animação, som e a manutenção do controle da performance do aprendiz. São muito utilizados devido permitirem a introdução do computador na escola sem provocar muita mudança, pois representa uma versão computadorizada do que já ocorre na sala de aula [Valente 1993]. Dentre as características, ressalta-se a animação, que permite mostrar de forma diferente da tradicional, como a informação sobre determinado conteúdo pode ser apresentada.

Vale lembrar que na internet há predominância de informalidade na produção de tutoriais, uma vez que se pode encontrar em sítios como o Youtube (<http://youtube.com>) diversos vídeos tutoriais sobre os mais variados assuntos, feitos por quaisquer pessoas independentemente de ter um conhecimento científico. Em outros domínios que envolvem tecnologia e o mundo digital, também pode-se encontrar uma grande gama de tutoriais, desde um tutorial sobre remover e/ou instalar um programa a um tutorial sobre como fazer um bolo. Assim, qualquer indivíduo, considerando seus conhecimentos e experiências, é capaz de produzir um tutorial. Todavia, a qualidade deste vai depender da sua capacidade de expressão e conhecimento das características dessa ferramenta.

Quanto ao termo interatividade, este é um fenômeno da “sociedade da informação”. Tem origem nos anos de 1970 e ganha notoriedade a partir dos anos de 1980 [Silva 2010]. Atualmente, é muito comum ouvir que determinado produto é interativo, ou seja, dotado de interatividade.

Alguns elementos necessários às tecnologias interativas são a participação, a intervenção, a bidirecionalidade e a multiplicidade de conexões [Silva 2010]. Dessa forma, pode-se afirmar que um tutorial interativo, como produto dotado de interatividade, deve apresentar características como participação, intervenção, bidirecionalidade e multiplicidade de conexões.

Na interatividade é essencial a relação do emissor com o receptor. Deve haver a participação-intervenção do receptor, onde participar vai muito além do responder sim ou não, ou de apenas escolher uma opção dada. Essa participação tem que ser capaz de modificar e interferir na mensagem. A comunicação existente é produção conjunta da emissão e da recepção, na qual o emissor é receptor em potencial e o receptor é emissor em potencial o que significa que ambos codificam e decodificam [Silva 2010].

Ressalta-se o papel do emissor e do receptor na utilização do tutorial interativo. O receptor deve se preocupar com quem será o seu receptor e como este poderá se comportar.

Em vista do que foi mencionado, conclui-se que no tutorial interativo deve haver a possibilidade do receptor participar através da navegação ao longo deste recurso e poder intervir. Essa intervenção, por exemplo, pode se dar quando o receptor escolhe a forma como vai usar o tutorial.

2.3. Scratch

Esta seção aborda sobre o *software* Scratch e sua utilização na disseminação do Pensamento Computacional.

O ensino da programação e da computação é importante pois permite que o indivíduo compreenda como são criados os sistemas operacionais com os quais ele interage e assim ele passa a entender as intenções deste sistema [Omine 2014]. Na concepção de Omine (*apud* Lemos, 2013) o conhecimento da programação é essencial, inclusive para a manutenção da democracia.

Steve Jobs, durante uma entrevista em 1995, afirmou que todos deveriam aprender a programar uma linguagem computacional, pois ensina a pensar. “É como ir para a faculdade de direito. Não acho que todos devem ser advogados, mas acho que fazer direito pode ser útil, pois ensina a pensar de um certo modo”, defendeu ele [Steve Jobs: The Lost Interview, 2012].

Na concepção de Rodrigues (2018) o mundo está ingressando na revolução industrial 4.0 e isso causará mudanças na economia e em especial no mercado de trabalho, com a automação de muitos trabalhos que atualmente exigem esforços repetitivos. Diante deste cenário, segundo o autor, duas competências serão fundamentais: resolver problemas complexos e gerar resultados a partir do uso de tecnologia. Caso contrário a sociedade estará diante de um abismo social entre os que possuírem estas competências e aqueles que não aprenderam a usar as tecnologias, muito menos a pensar de forma autônoma. Rodrigues (2018) alerta ainda para a urgência da existência de um sistema que familiarize as pessoas às tecnologias, pois não saber programar será o analfabetismo do futuro. Para ele é preciso que as crianças aprendam a programar nas escolas e que os adultos também sejam inseridos neste processo de desenvolvimento, uma vez que a expectativa de vida destes é cada vez maior.

Todavia, grande parte dos usuários têm dificuldade durante o processo de aprendizagem de programação, especialmente os que são iniciantes, pois ainda não dominam a lógica de programação e não possuem conhecimento sobre algoritmos. Diante disso, desde o início da década de 70 pesquisadores buscam desenvolver linguagens e ambientes de programação mais compreensíveis para um maior número de pessoas, deixando a linguagem de programação um pouco mais próximo da linguagem natural [Scaico et al. apud Paush e Kelleher 2005].

Neste sentido o Scratch destaca-se por utilizar o conceito de desenvolvimento orientado ao *design*. Nesta ferramenta, o iniciante não precisa dominar a sintaxe para ter suas primeiras experiências com programação. Isso porque o Scratch utiliza o que chama-se de programação em blocos. Ou seja, para criar seus projetos, o usuário não precisa digitar linhas de código. Em vez disso, ele vai arrastando blocos de código e criando suas histórias, jogos, animações e outros. Este projeto pode ser iniciado “do zero”, recomendado para usuários que já possuem uma certa experiência, ou pode ser criado a partir de outros projetos criados e disponibilizados por outros usuários.

O Scratch é uma ferramenta de programação, disponibilizada no *link* <www.scratch.mit.edu> (ver Figura 1), que permite que seus usuários programem suas histórias, jogos e animações interativas por meio de uma interface gráfica utilizando blocos. Tal ferramenta faz parte de um projeto surgido em 2007 e mantido pelo *Lifelong Kindergarten* do grupo *Massachusetts Institute of Technology (MIT) Media Lab* [LIFELONG KINDERGARTEN GROUP 2007].

Segundo o Lifelong Kindergarten Group (2007), o Scratch foi concebido especialmente para a faixa etária de 8 a 16 anos de idade, todavia atualmente é usado por pessoas de todas as idades (ver Gráfico 1). Milhões de usuários criam projetos Scratch numa grande variedade de contextos, incluindo lares, escolas, museus, bibliotecas e centros comunitários” [Lifelong Kindergarten Group 2007].

Atualmente o Scratch está disponível em mais de 150 países (Figura 2) e em 40 línguas e possui mais de 32 milhões de projetos² que foram criados por seus 28 milhões de usuários e compartilhados para que outras pessoas possam reutilizá-los, dando origem a novas histórias. Desse total de usuários, 1,39% são brasileiros, o que equivale a mais de 413 mil usuários. Este número ainda é baixo, quando comparado com os Estados Unidos que possui mais de 13 milhões de usuários. Todavia, é importante ressaltar que o Brasil é um dos países da América Latina com a maior quantidade de usuários de Scratch.

Ademais, é importante ressaltar que o número de usuários do Scratch cresce a cada ano no mundo inteiro, conforme mostra o Gráfico 2. Portanto, é uma ferramenta que a cada dia torna-se mais popular e, conseqüentemente, cresce o número de projetos que são disponibilizados, facilitando assim, para que novos usuários façam uso destes durante o processo de iniciação ao uso da ferramenta.

Utilizando o Scratch, o usuário além de aprender a lógica de programação, também pode desenvolver outras habilidades como o pensamento criativo, análise crítica e aprendizagem contínua. Isso porque esta ferramenta é embasada no

² Segundo dados disponíveis no *hiperlink* <https://scratch.mit.edu/statistics>. Acesso em 14 de Junho de 2018.

construcionismo, uma vez que a mesma foi desenvolvida a partir da linguagem LOGO (criada por Seymour Papert). Isso faz com que o Scratch se apresente como um importante aliado para o ensino-aprendizagem, com vistas ao desenvolvimento do Pensamento Computacional.

Ao manuseá-lo, o docente pode fazer uso de várias possibilidades de utilizações pedagógicas, podendo realizar atividades elaboradas de acordo com o conteúdo programático de certa disciplina, com a faixa etária dos estudantes, com o contexto histórico-cultural e etc. No presente trabalho, optou-se por adotar a referida ferramenta, juntamente com Tutoriais Interativos, no processo de ensino-aprendizagem, com o objetivo de desenvolver o Pensamento Computacional.



Figura 1: Página inicial do website do Scratch. Fonte: <https://scratch.mit.edu>. Acesso em 01 de Julho de 2018

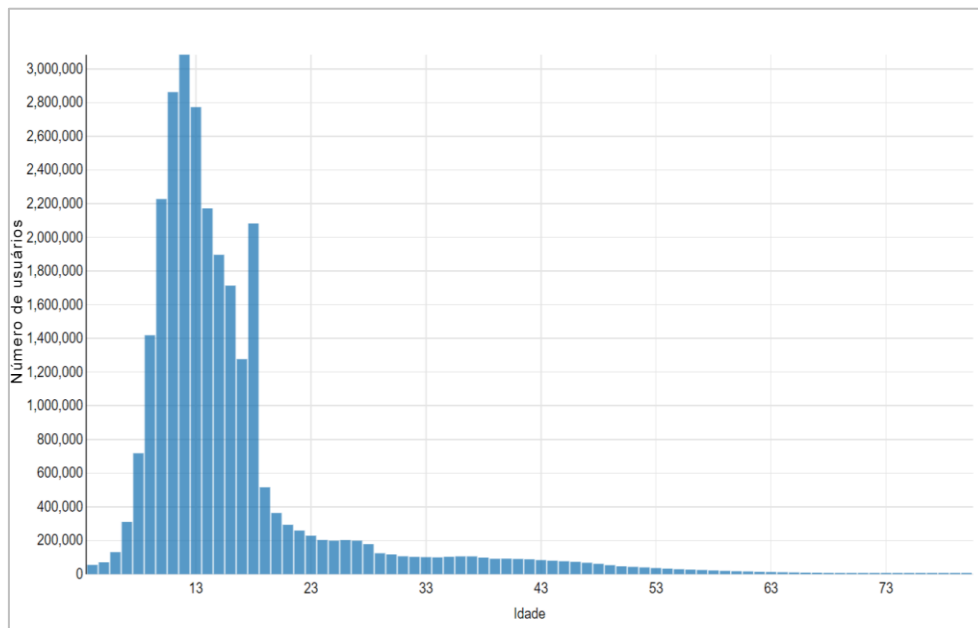


Gráfico 1: Distribuição por idade dos novos usuários de Scratch. Fonte: www.scratch.mit.edu. Acesso em 01 de Setembro de 2018.

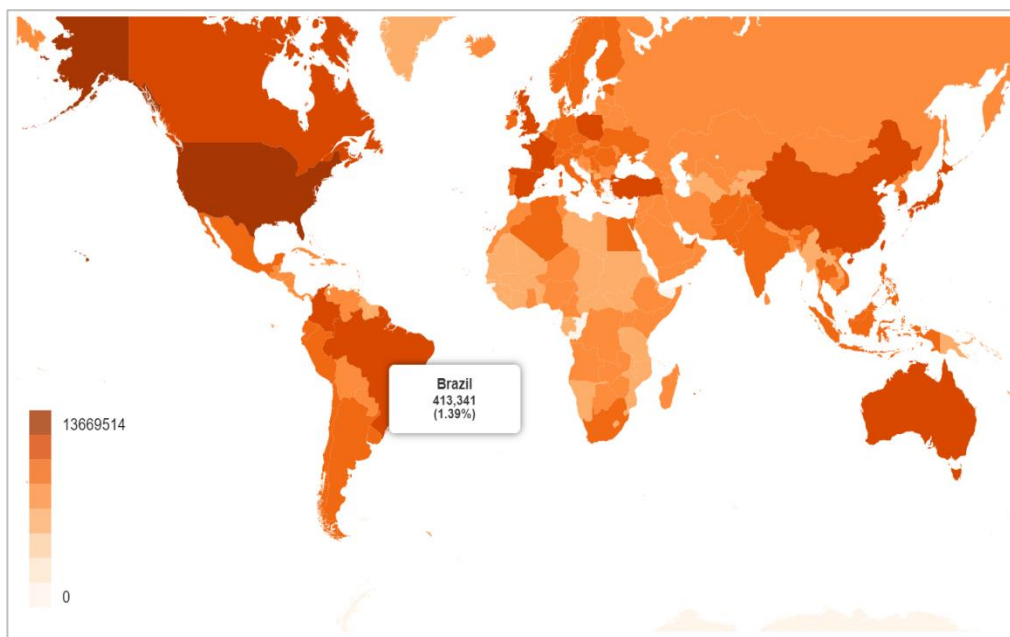


Figura 2: Distribuição dos usuários de Scratch no mundo, com destaque para a quantidade de usuários no Brasil. Fonte: www.scratch.mit.edu. Acesso em 01 de Setembro de 2018

Além do mais, a ferramenta apresenta inúmeras vantagens aos discentes, pois a mesma possibilita que seus usuários absorvam vários conhecimentos enquanto criam seus projetos. Alguns destes conhecimentos estão elencados no artigo “*Learning with Scratch*”³, dentre os quais destacam-se:

³ Learning with Scratch – the Scratch Documentation Website - MIT. Disponível em: <https://llk.media.mit.edu/scratch/Learning-with-Scratch.pdf>. Acesso em 01 de Agosto de 2018.

- **Conhecimento matemático e computacional** – enquanto criam seus projetos os usuários absorvem os conhecimentos básicos da computação, tais como iteração e condicionais. Além destes os usuários também aprendem a lidar com conceitos matemáticos como coordenadas, variáveis e números aleatórios;
- **Processo do “design”** – durante a criação de um projeto no Scratch o usuário cria uma nova ideia. Posteriormente ele faz um protótipo desta ideia, realiza testes, executa a depuração quando surgem erros, além da possibilidade de receber comentários de outros usuários. Isto faz com que o usuário revise o seu projeto por diversas vezes durante o processo criativo. Essa sequência de passos constituem um ciclo contínuo (ver Figura 3) onde um projeto nasce a partir de uma ideia e este dar origem à outros projetos e assim sucessivamente. Com isso ao utilizar o Scratch o usuário está em constante processo de aprendizagem e construção do conhecimento. Por sua vez este processo auxilia no desenvolvimento de cidadãos mais críticos, que tenham habilidade de comunicar-se, de realizar análises sistemáticas, colaborar efetivamente, além de aprender de forma contínua.
- **Maior “fluência” com a tecnologia digital** – ser fluente em tecnologia digital não significa saber realizar tarefas básicas em um computador, com enviar um *e-mail* ou digitar um texto. E sim utilizar o computador para criar algo com ele.

Em vista do que foi mencionado, pode-se afirmar que o Scratch apresenta um grande potencial para fins de ensino. Além do mais, a popularização da ferramenta, somada à sua facilidade de acesso e uma vasta documentação, fez com que a mesma se tornasse uma importante aliada no processo de ensino-aprendizagem. Segundo o Lifelong kindergarten group (2007) diversos educadores têm utilizado a ferramenta no ensino básico, secundário e até mesmo em universidades. Essa utilização tem sido aplicada nas mais diversas áreas do conhecimento. Seja para o ensino-aprendizagem de ciências, matemáticas, línguas estrangeiras, artes, estudos sociais e letras.

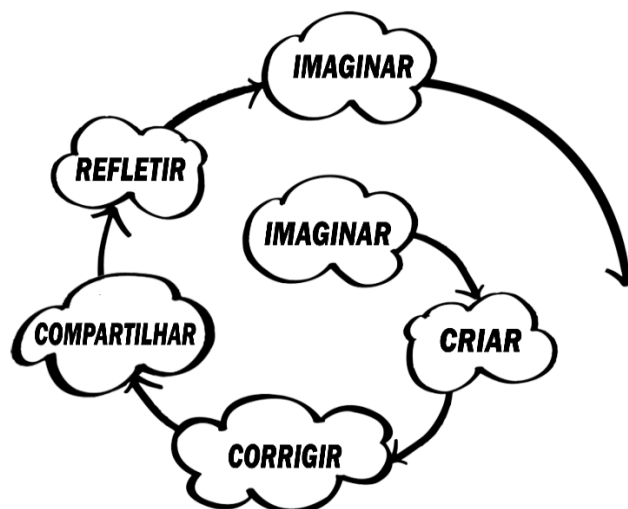


Figura 3: Ciclo de criação no Scratch³

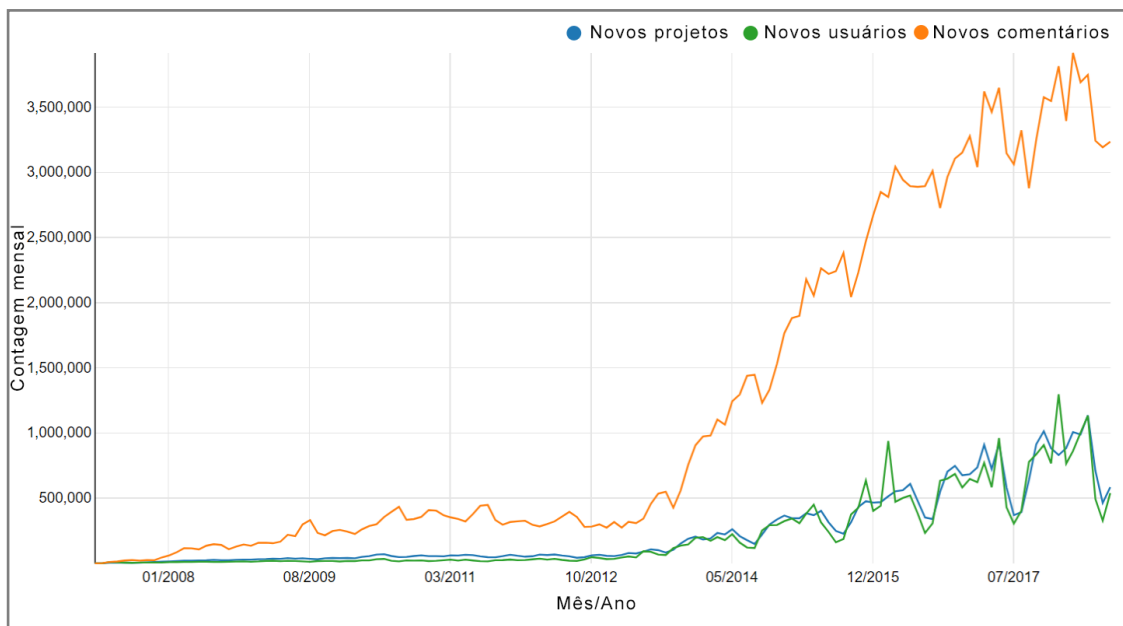


Gráfico 2: Atividades mensais no website no Scratch. Traduzido pelos autores.
Fonte: <https://scratch.mit.edu/statistics/>. Acesso em 01 de Agosto de 2018.

3. O uso de Tutoriais Interativos no desenvolvimento do “pensar computacionalmente”

Nesta seção será abordada de que maneira pode ser elaborado um Tutorial Interativo (TI), apresentando suas telas principais e como atribuir interatividade a esta ferramenta.

Para elaboração dos tutoriais, podem ser usados programas como o *Microsoft Office Power Point* e *Microsoft Paint*, que são *softwares* que estão disponíveis nos computadores de muitas pessoas. A utilização do *Power Point* é recomendada porque possui recursos como botões de ação, *hiperlinks*, dentre outros, que possibilitam atribuir interatividade aos TIs. Já o *Paint* é interessante para tratamento das imagens que serão utilizadas no tutorial. Todavia, é importante destacar, que caso o usuário não disponha destes recursos computacionais, por estes não serem de licença livre, o mesmo pode fazer uso de outras ferramentas que possibilitem a criação de apresentações (similares às criadas no *Power Point*) e edição de imagem (tal qual é possibilitado pelo *Paint*).

A Página de Apresentação, como pode ser observada na Figura 4, será a primeira tela que o usuário visualizará. Para tanto, devem ser colocadas informações como o título do tutorial interativo, botão *play* e nome do autor ou autores. O título deverá ser objetivo e sucinto. O botão *play* servirá para o usuário prosseguir para a próxima tela.

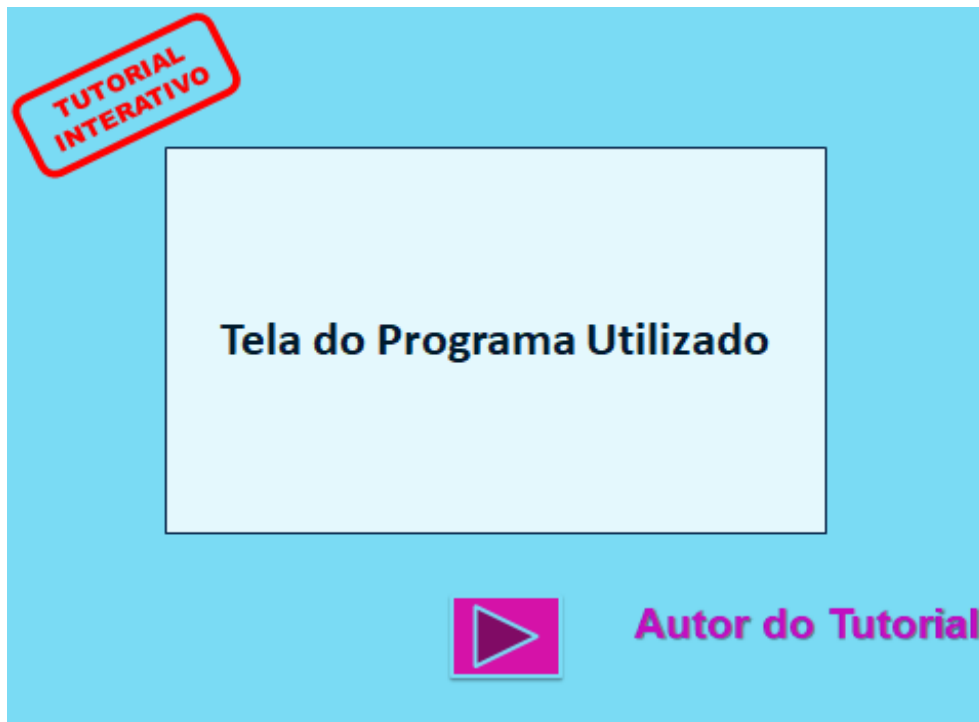


Figura 4: Página de Apresentação. Fonte: Autores (2018)

Na Página Inicial, apresentada na Figura 4, deverá apresentar o título do tutorial, além de três botões localizados na parte inferior da tela: Como Usar, Menu Principal e Ajuda. Estes serão úteis para facilitar a exploração do ambiente, permitindo uma rápida navegação.

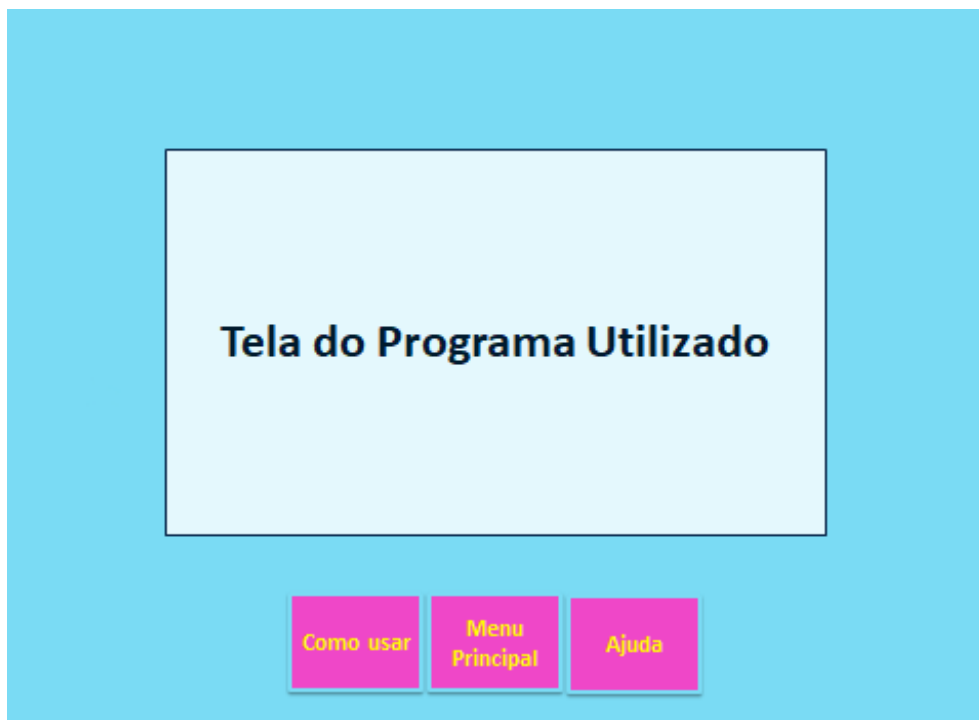


Figura 5: Página Inicial. Fonte: Autores (2018)

Na Página Como Usar, mostrada na Figura 6, deve-se apresentar botões de atalho na parte superior direita da tela, o título da página, balões explicativos e botões voltar e avançar. Estes permitirão também uma navegação rápida e eficiente pelo ambiente. Os balões explicativos servem para informar o que determinada funcionalidade faz.



Figura 6: Página Como Usar. Fonte: Autores (2018)

No Menu Principal, tela apresentada na Figura 7, há botões de atalho, título da página, botões voltar e avançar, além dos *links* das principais partes da ferramenta.



Figura 7: Página Menu Principal. Fonte: Autores (2018)

Em O que é o Programa, Figura 8, pode-se expor sobre o que é o *software* escolhido, suas funcionalidades e possibilidades de uso.

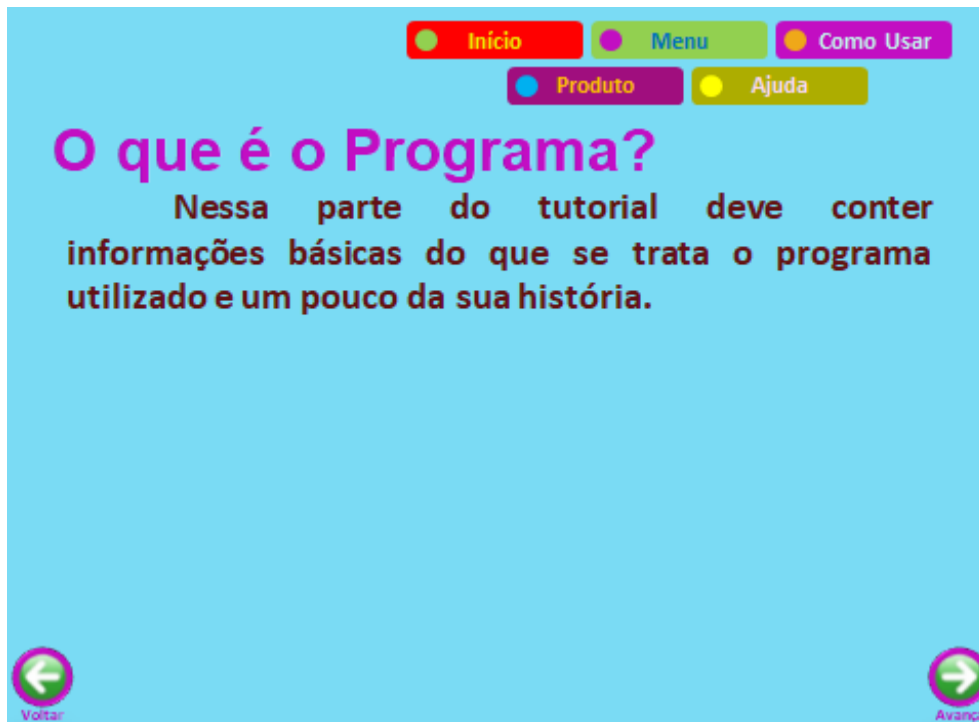


Figura 8: Página O que é o Programa. Fonte: Autores (2018)

Em Baixando o Programa, Figura 9, é interessante fornecer informações de onde se pode baixar o programa.

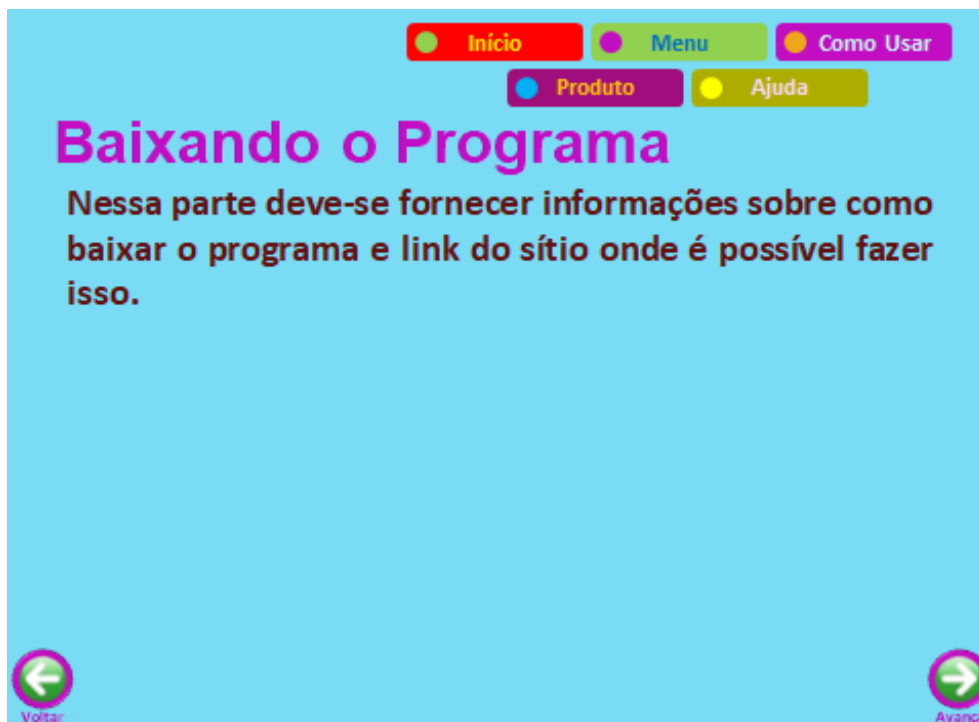


Figura 9: Página Baixando o Programa. Fonte: Autores (2018)

Já em Instalando o Programa, Figura 10, é necessário destacar em quais sistemas operacionais o programa pode ser instalado, quais requisitos são necessários, dentre outros.



Figura 10: Página Instalando o Programa. Fonte: Autores (2018)

Na Página Conhecendo o Ambiente, Figura 11, o usuário poderá conhecer de forma mais detalhada as funcionalidades do programa. Sugere-se a inserção de imagens ilustrativas do ambiente, bem como animação de como utilizar as funcionalidades.



Figura 11: Conhecendo o Ambiente. Fonte: Autores (2018)

Criando o Produto, Figura 12, é uma das partes mais importantes do tutorial interativo, porque deve apresentar como é elaborado o produto do programa. Por exemplo, se for um programa de planilha eletrônica, deverá elencar os passos

necessários para se criar uma planilha, se for um editor de simulação deverá mostrar as etapas para se utilizar a simulação.

Em Ajuda, Figura 13, há informações para sanar dúvidas do usuário quanto à utilização do tutorial.



Figura 12: Página Criando o Produto. Fonte: Autores (2018)



Figura 13 – Página Ajuda. Fonte: Autores (2018)

4. Proposta de Desenvolvimento de um Tutorial Interativo

A seguir são apresentadas propostas de desafios para nortear e estimular o desenvolvimento de um tutorial interativo. Destaca-se que o recurso computacional utilizado para a construção do tutorial será o Scratch. As figuras 14 a 26 ilustram o resultado esperado das atividades realizadas nos desafios a seguir.

4.1 Desafios

Embora os desafios estejam relacionados ao Scratch, eles podem ser aplicados aos mais diversos recursos computacionais.

4.1.1 Desafio 1

1. Analise a ferramenta Scratch;
2. Verifique as partes que compõe o ambiente;
3. Pesquise informações adicionais sobre o Scratch;
4. Visite e explore a aplicação Teste de Matemática 2 (Figura 14), disponível em <https://scratch.mit.edu/projects/3295166/#editor>

É importante a análise do recurso que se pretende utilizar, porque através disso, o desenvolvedor terá uma visão geral de como o programa funciona. A partir de então, poderá ser feita a reflexão de como o TI será estruturado e de como planejá-lo. Esse planejamento influenciará a qualidade do tutorial. Ao visite a aplicação Teste de Matemática, espera-se que o usuário tenha algumas noções de como se organiza um código no ambiente Scratch.



Figura 14: Teste de matemática. Disponível em: <https://scratch.mit.edu/projects/3295166/#editor>. Acesso em 10 de Setembro de 2018

4.1.2 Desafio 2

1. *Desenvolva a Página de Apresentação com as seguintes partes: nome do autor do tutorial, imagem da tela do ambiente e botão play;*

A página de apresentação será a primeira tela que o usuário terá contato. Dessa forma, há a necessidade de despertar a atenção e desejo do sujeito em utilizar o tutorial. Essa tela é essencial, uma vez que através dela, as primeiras informações serão repassadas ao usuário.



Figura 15: Página de Apresentação. Fonte: Autores (2018)

4.1.3 Desafio 3

1. *Crie a Página Inicial que deverá abarcar os botões Como Usar, Menu Principal e Ajuda, além da imagem do recurso computacional utilizado;*

A página inicial, ao apresentar esses três botões, facilita a navegação pelo TI.



Figura 16. Início. Fonte: Autores (2018)

4.1.4 Desafio 4

1. Elabore as telas Como Usar, incluindo os botões de atalho, balões explicativos, imagens das funcionalidades explicadas e botões voltar e avançar;



Figura 17. Como Usar. Fonte: Autores (2018)

Diferentemente de outras partes do tutorial, que apresentam apenas uma tela, o Como Usar traz diversas telas, dependendo apenas da intenção de quem está desenvolvendo o tutorial.

4.1.5 Desafio 5

1. Produza o Menu Principal para facilitar a navegação pelo tutorial. Para isso, insira os botões de atalho, título da página, botões voltar e avançar, além dos links das principais partes do ambiente;

Por apresentar as principais partes que o tutorial está organizado, pode-se afirmar que é uma das telas mais importantes e que precisa ser bem elaborada.



Figura 18: Menu Principal. Fonte: Autores (2018).

4.1.6 Desafio 6

1. Faça a tela O que é o Programa, destacando as principais informações sobre o ambiente;

Essa tela é relevante devido ser através dela que serão mostradas noções do potencial do recurso escolhido.

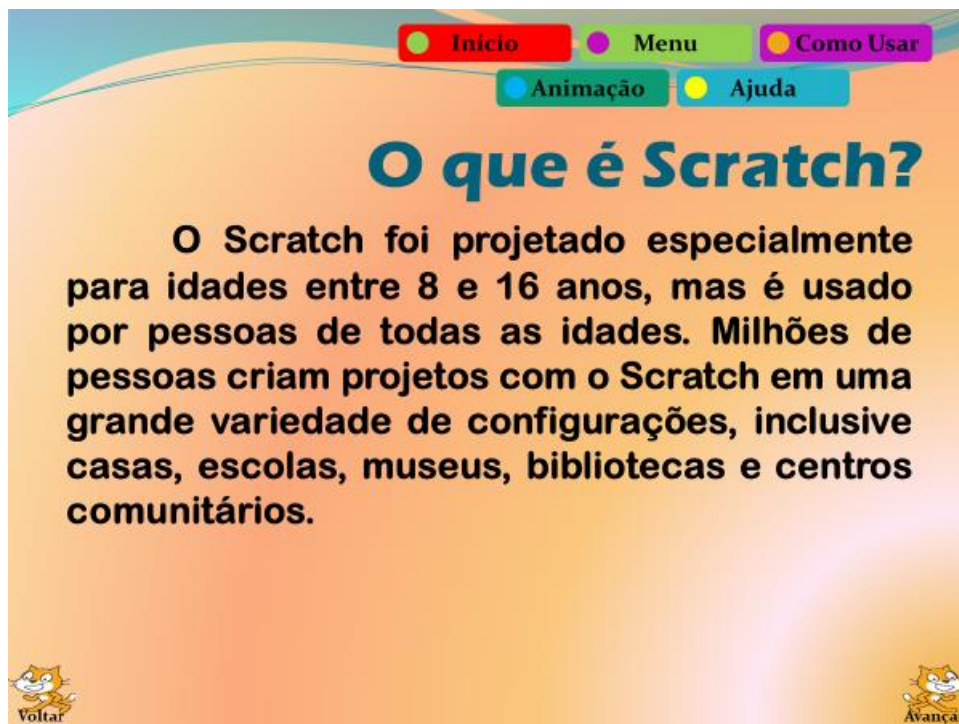


Figura 19: Menu Principal. Fonte: Autores (2018)

4.1.7 Desafio 7

1. Informe em Baixando o Programa o sítio onde o recurso está disponível para ser baixado;



Figura 20: Ensinando como baixar o Scratch. Fonte: Autores (2018)

Nessa parte deve conter indicações de onde pode-se baixar o programa. Isso é relevante pois, muitas das vezes, o usuário não tem o software instalado no seu computador. Vale ressaltar também que alguns ambientes podem ser utilizados *online*.

4.1.8 Desafio 8

1. *Apresente em Instalando o Programa como o usuário poderá proceder com a instalação do software;*

Após baixar o recurso computacional, torna-se primordial saber como instalá-lo. Como algumas instalações não são tão simples e intuitivas, deve-se informar quais recursos extras são necessários e em quais sistemas operacionais o programa pode ser instalado.



Figura 21: Instalando o Scratch. Fonte: Autores (2018)

4.1.9 Desafio 9

1. *Para a tela Conhecendo o Ambiente do Programa, demonstre quais as principais funcionalidades da aplicação;*

As funcionalidades mais imprescindíveis devem ser destacadas, para quando o usuário for criar o produto, ele terá conhecimento mínimo de como proceder.



Figura 22: Conhecendo o Scratch. Fonte: Autores (2018)



Figura 23: Exemplo de uma funcionalidade do programa. Fonte: Autores (2018)

4.1.10 Desafio 10

1. Ao elaborar a tela Criando um Produto Prático do Programa, apresente as etapas necessárias para a criação do produto que pode ser desenvolvido através do ambiente;

Sugere-se a apresentação de uma sequência de passos numerada para que o usuário saiba qual etapa ele deve executar antes de passar para a próxima.

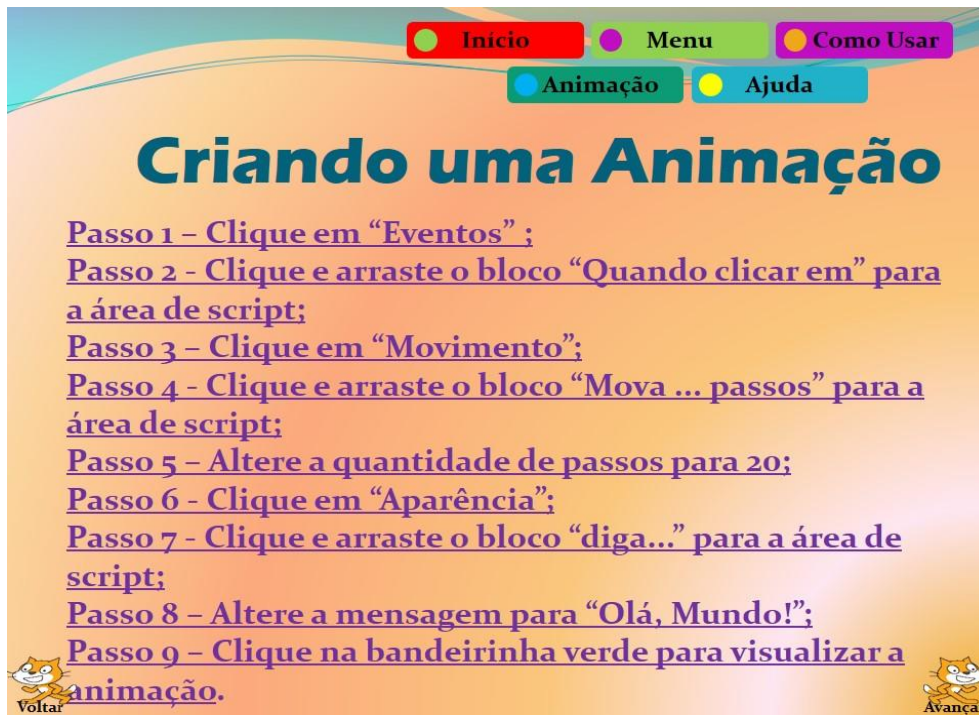


Figura 24: Criando uma Animação. Fonte: Autores (2018)



Figura 25: Exemplo de uma etapa. Fonte: Autores (2018)

4.1.11 Desafio 11

1. *Implemente a parte Ajuda com a finalidade de fornecer ao usuário, informações capazes de sanar possíveis dúvidas.*

Como, ao longo da exploração do ambiente, o utilizador da ferramenta pode ficar em dúvida sobre algo, essa parte deve prever as possíveis incertezas que poderão surgir.



Figura 26: Ajuda. Fonte: Autores (2018)

4.1.12 Desafio 12

1. *Verifique se os links estão funcionando corretamente, bem como as animações e botões de ação.*

Para finalizar a construção do tutorial, é necessário revisar todo o tutorial com a finalidade de se verificar se há algum erro e, conseqüentemente, corrigi-lo.

5. Considerações Finais

Durante a elaboração deste ficou evidente a importância da disseminação do Pensamento Computacional, uma vez que este é tão importante para o indivíduo do presente século.

Sabe-se que em algumas escolas do Brasil a disciplina de “Informática” já está presente. Todavia, é preciso que algo a mais seja feito para que as aulas possam ir além de ensinar como utilizar programas de computadores, tais como editores de texto, de imagem ou navegar na internet. É necessário que hajam iniciativas no sentido do ensino da computação enquanto ciência, de modo que o aluno consiga programar o computador e utilizá-lo como um “agente” potencializador de suas habilidades.

Entretanto, ensinar sobre o Pensamento Computacional ainda pode ser um desafio no Brasil, seja por falta de laboratórios - em várias escolas, ou até mesmo

devido este ainda não está na grade curricular. Dessa forma, parte dos professores tem se utilizado do espaço de outras disciplinas para disseminar este conhecimento tão importante.

Neste sentido, o Scratch apresenta-se como uma excelente ferramenta, pois a mesma representa um suporte multidisciplinar, permitindo que o docente trabalhe com os mais variados assuntos, sem, contudo, deixar de lado o ensino da lógica de programação e o desenvolvimento do Pensamento Computacional.

E aqui convém destacar a importância dos Tutoriais Interativos para facilitar este processo de ensino. Os TI facilitam no processo comunicativo, fazendo com que o conteúdo fique mais dinâmico e mais atrativo.

Portanto, estas duas ferramentas podem ser importantes aliadas dos educadores na disseminação do Pensamento Computacional. São de fácil acesso e não exigem que o educador tenha um grande conhecimento em informática para manuseá-las. Além disso, podem ser utilizadas para o ensino-aprendizagem de quase todos os conteúdos, das mais diversas disciplinas, nos mais diversos níveis de escolaridade.

Referências

- Barcelos, T. S., & Silveira, I. F. (2012) “Pensamento computacional e educação matemática: Relações para o ensino de computação na educação básica”. In XX Workshop sobre Educação em Computação, Curitiba. Anais do XXXII CSBC (Vol. 2, p. 23).
- Barr, V., & Stephenson, C. (2011) “Bringing computational thinking to K-12: what is involved and what is the role of the computer science education community?”. *Acm Inroads*, 2(1), 48-54.
- Blikstein, P. (2008) “O Pensamento Computacional e a reinvenção do computador na educação”.
- Brennan, K., & Resnick, M. (2012) “New frameworks for studying and assessing the development of computational thinking”. In Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada (Vol. 1, p. 25).
- Charlot, B. (2008) “O professor na sociedade contemporânea: um trabalhador da contradição”. *Revista da FAEEBA–Educação e Contemporaneidade*, Salvador, 17(30), 17-31.
- Francisco, A. C.; Oliveira, A. C. (2006) “Educação à distância no ensino superior: constatações e recomendações”. In XIII SIMPEP, 06 a 08 de novembro de 2006, Bauru, SP, Brasil.
- França, R. S., Ferreira, V. F. S., Almeida, L. D., & 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., & 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).
- Geraldes, W. B. (2017) “O Pensamento Computacional no ensino profissional e tecnológico”. 81 f. Dissertação (Mestrado em Gestão do Conhecimento e Tecnologia da Informação) – Pós-Graduação Stricto Sensu em Gestão do Conhecimento e Tecnologia da Informação – Universidade Católica de Brasília, Brasília, 2017.
- Lifelong Kindergarten Group (2007) “Scratch: Imagine, Program, Share”. MIT MediaLab, 2007. Disponível em: <http://Scratch.mit.edu/>. Acesso em 01 maio 2017.
- Primo, Alex. (2007) “O aspecto relacional das interações na Web 2.0”. E- Compós (Brasília), v. 9, p. 1-21, 2007.
- Reis, L. V. P.; Romão, A. U.; Leite, J. C. (2011) “Tutoriais: Um Modo Tipicamente Contemporâneo de Troca de Conhecimentos”. In XIII Congresso de Ciências da Comunicação na Região Centro-Oeste – Cuiabá – MT, 8 a 10 de junho de 2011, Cuiabá, Mato Grosso, 2011.
- Rodrigues, M. B. “Tecnologia deve ser ensinada desde a infância”: superintendente Centro-Sul American Chamber (Amcham) (Brasil). Disponível em: <<https://gauchazh.clicrbs.com.br/opiniao/noticia/2018/06/tecnologia-deve-ser-ensinada-desde-a-infancia-cji3pt6j50e8a01qoj83jm3ti.html>>. Acesso em: 14 jun. 2018.
- Rodrigues, R. da S. et al. (2015) “Análise dos efeitos do Pensamento Computacional nas habilidades de estudantes no ensino básico: um estudo sob a perspectiva da programação de computadores”. In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE). p. 121.
- Santiago, M. S. (2010) “Aspectos linguísticos de fraseologias em tutoriais da educação a distância”. In III Coloquio do PPG-Letras/UFRGS. Cadernos do IL. Porto Alegre, n.o 40, junho de 2010. p. 95-105. Disponível em <<http://www.seer.ufrgs.br/cadernosdoil/>>. Acesso em: 18 jan. 2016.
- Silva, M. (2010) “Sala de aula interativa: educação, comunicação, mídia clássica...” 5ª edição, São Paulo: Loyola.
- Sousa, R. M.; Lencastre, J. A. (2014) “Scratch: uma opção válida para desenvolver o Pensamento Computacional e a competência de resolução de problemas”. 2º Encontro sobre Jogos e Mobile Learning, p. 256-267.
- Steve Jobs: The Lost Interview. Direção de Paul Sen. Produção de Robert X. Cringely (2012). Legendado. Disponível em: <<https://www.netflix.com/br/title/70243590>>. Acesso em: 14 jun. 2018
- Valente, J. A. (1993) “Diferentes usos do Computador na Educação”. Disponível em:<<http://www.mrherondomingues.seed.pr.gov.br/redeescola/escolas/27/1470/14/arquivos/File/PPP/Diferentesusosdocomputadoreducacao.PDF>>. Acesso em: 13 out. 2015.

Valente, J. A. (2005) “A espiral da espiral de aprendizagem: o processo de compreensão do papel das tecnologias de informação e comunicação na educação”. 238 p. Tese (livre-docência) - Universidade Estadual de Campinas, Instituto de Artes, Campinas, SP. Disponível em: <<http://www.bibliotecadigital.unicamp.br/document/?code=000857072>>. Acesso em: 1 abr. 2017.

The CSTA Standards Task Force, “CSTA K-12 Computer Science Standards,” ACM Computer Science Teachers Association, New York, 2011.

Sobre os autores

Gilson Pedroso dos Santos



Mestre em Educação com ênfase em Práticas Educativas, Linguagens e Tecnologias, pela Universidade Federal do Oeste do Pará. Possui especialização em Gestão Pública pela Faculdade Venda Nova do Imigrante (FAVENI), Bacharel em Sistemas de Informação pela Universidade Federal do Oeste do Pará (UFOPA) e licenciado em Letras pela Universidade Luterana do Brasil (ULBRA). Atualmente pesquisa sobre Pensamento Computacional e TIC no ensino da matemática. Integra o Grupo de Estudos e Pesquisas em Educação Matemática e Interdisciplinaridade na Amazônia (GEPEIMAZ). 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



Cientista da Computação pela Universidade Federal do Oeste do Pará (Ufopa) e pós-graduação em Informática da Educação e Docência do Ensino Superior pelo Instituto de Ciência, Educação e Tecnologia de Votuporanga - Faculdade Futura. 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.