



O USO DO SCRATCH COMO FERRAMENTA DE AUXÍLIO NO ENSINO SUPERIOR

Luciana Rocha Cardoso¹, Danielly dos Santos Emerick Faria²

¹ Mestre em Ciência da Computação, UNIFACIG, luroca@sempre.unifacig.edu.br

² Graduanda em Análise e Desenvolvimento de Sistemas, UNIFACIG, daniellysef@gmail.com

Resumo: Este trabalho aborda a importância do uso do *SCRATCH* no ensino do pensamento computacional para alunos de graduação superior. O *SCRATCH* é uma ferramenta que proporciona a criação de um ambiente para aprender sobre lógica matemática utilizando blocos programáveis como operadores lógicos, aritméticos e relacionais, dentre outros para o ensino e aprendizagem em diferentes áreas, não somente aos cursos de computação, por proporcionar habilidades que são essenciais para qualquer área que utiliza o raciocínio lógico. Para o desenvolvimento deste trabalho, foram realizadas pesquisas bibliográficas e em sites sobre Pensamento Computacional, *SCRATCH*, lógica matemática, entre outros assuntos que abordam sobre o tema proposto. O desfecho da referente pesquisa converge para o uso da ferramenta *SCRATCH* em ambientes de ensino, desde o ensino básico ao ensino superior como uma das principais motivações para desenvolver habilidades e competências.

Palavras-chave: Lógica Matemática; Pensamento Computacional; *SCRATCH*.

Área do Conhecimento: Ciências Humanas.

1 INTRODUÇÃO

Com a evolução das Tecnologias da Informação e Comunicação (TICS), ocorreram transformações em quase todas as áreas na sociedade, e principalmente na área da educação, estabelecendo novos ritmos e dimensões na incumbência de ensinar e aprender.

As tecnologias em sala de aula, principalmente o uso do computador na área educacional, auxilia o aluno na construção do conhecimento trazendo ganhos significativos para o aprendizado. Segundo Valente (1999),

O computador pode enriquecer ambientes de aprendizagem onde o aluno, interagindo com os objetos desse ambiente, tem chance de construir o seu conhecimento. Nesse caso, o conhecimento não é passado para o aluno. O aluno não é mais instruído, ensinado, mas é o construtor do seu próprio conhecimento. Esse é o paradigma construcionista onde a ênfase está na aprendizagem ao invés de estar no ensino; na construção do conhecimento e não na instrução (VALENTE, 1999, p. 24-25).

E assim, na construção do pensamento computacional, o aluno pode incluir um conjunto de itens e ferramentas, tais como: sistemas, objetos, algoritmos, problemas, soluções, abstrações e conjuntos de dados e/ou informações, para compreender e raciocinar sobre aspectos do mundo real para o computacional, resolvendo problemas, dividindo-os em partes e desenvolver algoritmos para solucioná-los (ANDRÉ, 2018).

Segundo Wing (2011, p. 20, apud André, 2018): “o termo pensamento computacional é definido por Wing como: processos de pensamento envolvidos na formulação de problemas e suas soluções, sendo que as soluções devem ser representadas de forma que possa ser realizada por agentes de processamento de informações”. Assim, para resolução de problemas, a solução é executada por um humano e uma máquina, em que onde o ser humano desenvolve métodos através do raciocínio lógico na construção de algoritmos computacionais, ou seja, o pensamento computacional é todo o procedimento de desenvolvimento de competências para auxiliar tanto o raciocínio quanto ao aprendizado.

Com o uso de tecnologias em sala de aula, especialmente com o uso do computador, auxilia o aluno na construção do pensamento computacional, trazendo ganhos significativos para o aprendizado, envolvendo uma visão interdisciplinar e compartilhando o conhecimento entre áreas distintas. Destacando o papel ativo do aluno, ensinando-o a colocar a mão na massa e se tornar protagonista do seu próprio aprendizado, promovendo habilidades e atitudes que permitem o uso de ferramentas tecnológicas de maneira proativa, lúdica, reflexiva e ética.

É fundamental que o aluno seja protagonista do seu ato educativo, mas sem desvalorizar o educador e a escola, pois o professor o orienta em seu trabalho e possibilita um ambiente adequado para desenvolvê-lo, e guiando o aluno a aquisição do conhecimento.

2 A ORIGEM DO SCRATCH E USO ATUAL

Nesse sentido, o raciocínio do texto acima descrito, será apresentada nesta seção a origem da *SCRATCH* que é uma das ferramentas que pode ser utilizada para o desenvolvimento do pensamento computacional, possibilitando o protagonismo do aluno e utilizada em sala de aula através do computador.

Pensando assim, Seymour Papert em 1967 desenvolveu junto com a sua equipe na Universidade Americana MIT (*Massachusetts Institute of Technology*) uma linguagem de programação interpretada e interativa chamada LOGO com objetivo didático voltado para crianças, jovens e adultos, podendo ser utilizada em diversas áreas do conhecimento com eficácia, sendo um ambiente de aprendizagem que possibilita a construção do conhecimento através do uso do computador (VALENTE; ALMEIDA, 1997).

Papert (1996) acreditava que o uso do computador poderia ser um aliado na educação, transferindo seu raciocínio lógico (algoritmo) através de instruções para o LOGO, caso este raciocínio estivesse incorreto LOGO retornaria um erro, possibilitando que aluno a reformule seu pensamento e reconstrua o algoritmo na tentativa de encontrar soluções viáveis para resolver o problema (AQUINO; GUEDES; AQUINO; 2014).

Quando acontece um erro, este torna-se um objeto de análise para que seja identificado e reformulado, desencadeando aprendizagem e desenvolvimento. Esse processo estabelece um ciclo de descrição-depuração - reflexão-depuração, que foi implantado na programação de computadores. (Papert (1986, apud JUNIOR, 2002))

Na década de 70, o uso de computadores era limitado, impossibilitando o experimento de Papert a ser utilizado pelo público geral, especialmente crianças. Pensando nesta limitação, Papert criou um robô que pudesse utilizar a lógica de programação e que seria mais acessível. Esse robô era conectado ao computador através de cabos e possuía motores com movimentos e uma caneta que poderia ser abaixada ou levantada de acordo com as instruções feitas pelo usuário. Logo mais tarde, esse robô ficou conhecido como LOGO *Turtle* (Tartaruga LOGO). O robô recebia as instruções vindos do computador. Foi então criada a linguagem de programação chamada LOGO que controlava os movimentos do robô que desenhava figuras geométricas em papéis, pois nesta época havia também limitação de recursos gráficos dos computadores. A partir da geração dos computadores pessoais e com mais acessibilidade ao público, a LOGO *Turtle* passou a ser virtual, ou seja, podendo movimentar-se na tela e foi conhecida mundialmente como SuperLogo.

Na Figura 1, temos um retrato a imagem de Seymour Papert com sua criação, a LOGO *Turtle*.

Figura 1: LOGO Turtle



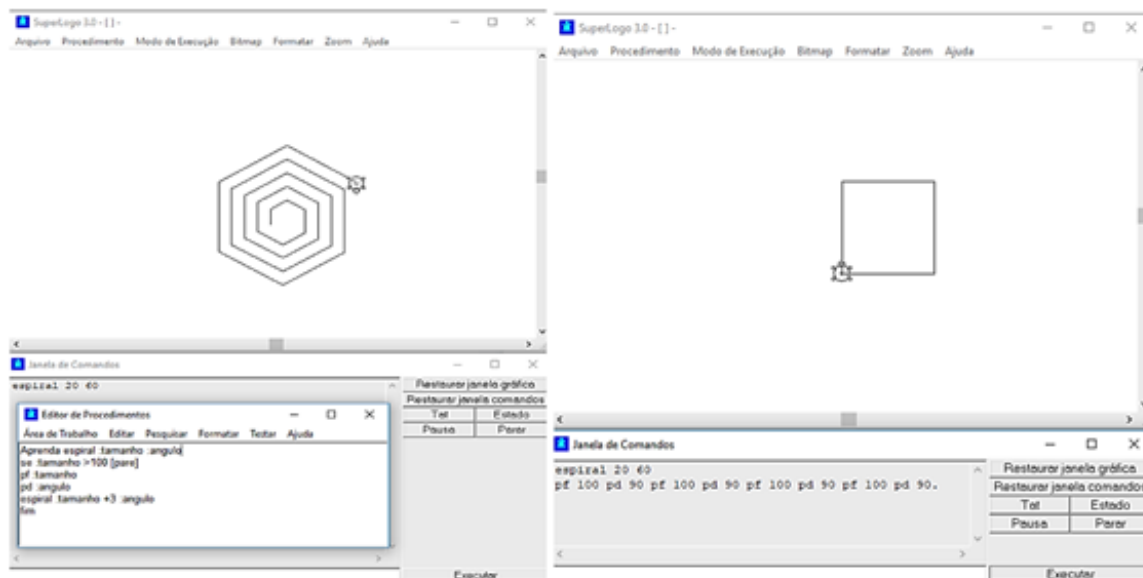
Fonte: The New Your Times¹

O SuperLogo, ou simplesmente SLogo, é um ambiente de programação que simula a movimentação de um robô, representado por uma tartaruga, que permite tanto crianças quanto adultos usarem o computador como ferramenta de aprendizagem e que através de alguns comandos se desloca na tela. Por ser uma linguagem de fácil compreensão, ela foi criada para o ambiente escolar, com objetos de ensinar lógica de programação e lógica matemática.

Os comandos de movimentos são bem simples, como: pf (para frente), pt (para trás), pd (para direita) e pe (para esquerda), sempre especificando o número de passos (pf e pt) ou o grau do giro (pe e pd), entre outros comandos na construção do raciocínio lógico. Veja um exemplo de reprodução desse ambiente em versão SuperLogo (Figura 2).

¹ Seymour Papert, 88, Dies; Saw Education's Future in Computers. Disponível em: <https://www.nytimes.com/2016/08/02/technology/seymour-papert-88-dies-saw-educations-future-in-computers.html>

Figura 2: SuperLogo



Fonte: Acervo Pessoal

Existem várias versões de *softwares* para a linguagem LOGO, com ambientes diferentes e mais abrangentes que o SLogo, trabalhando com textos, fórmulas matemáticas mais complexas e até mesmo trabalhando com IA (Inteligência Artificial): AF Logo, XLogo, Projeto Logo, MSWLogo, BetaLogo etc., que podem ser utilizados como fins de aprendizado.

Em 2007, Mitchel Resnick, também do MIT, lançou o *SCRATCH*, que é uma linguagem gráfica de programação inspirada no LOGO, em que o principal objetivo era o desenvolvimento de um ambiente de programação que permitisse que qualquer pessoa programasse. O *SCRATCH* é um ambiente que possibilita facilmente que pessoas leigas em programação possam criar histórias interativas, jogos, animações e simulações, sem que tenham conhecimentos prévios sobre tal e sem restrições de idade. O *SCRATCH* possui um ambiente *on-line* no qual pode-se compartilhar todas as suas criações com outros usuários (SALAZAR; ODAKURA; BARVINSKI; 2015).

O *SCRATCH* está sendo muito utilizado para fins educacionais no mundo, principalmente o uso da construção do pensamento computacional, desenvolvendo um raciocínio lógico com maior facilidade. Portanto, muito utilizado em instituições de ensino, de diversos níveis de ensino, e no Ensino Superior no aprendizado de programação, em que acadêmicos possuem dificuldades nas disciplinas iniciais de programação e em várias outras disciplinas, como Matemática, Química, Biologia, Física, entre outras.

Sua interface é composta de blocos lógicos e itens de som e imagem. É usado em mais de 150 países e disponível em mais de 40 idiomas (*SCRATCH* Brasil, 2014). Mesmo sendo projetado para uso de crianças, o *SCRATCH* é muito utilizado do Ensino Fundamental ao Ensino Superior.

A capacidade de escrever programas de computador é uma parte importante da alfabetização na sociedade de hoje. Quando as pessoas aprendem a programar usando o *SCRATCH*, elas aprendem estratégias importantes para a resolução de problemas, projetos de design e comunicação de ideias. (*SCRATCH* Brasil², 2014).

Seguindo este pressuposto, o *SCRATCH* foi desenvolvido para encurtar o problema do distanciamento da evolução tecnológica no mundo e habilidade tecnológica que os usuários possuem para promover um contexto construcionista. Desde criança desenvolvem competências transversais para a educação do século XXI. Utilizando tecnologias na aprendizagem de forma mais simples e intuitiva os encorajando e estimulando a não ser meros utilizador da tecnologia e sim, os tornando proativos em criar soluções para problemas de forma colaborativa e aprender habilidades do pensamento lógico e computacional.

De acordo com Blikstein (2008),

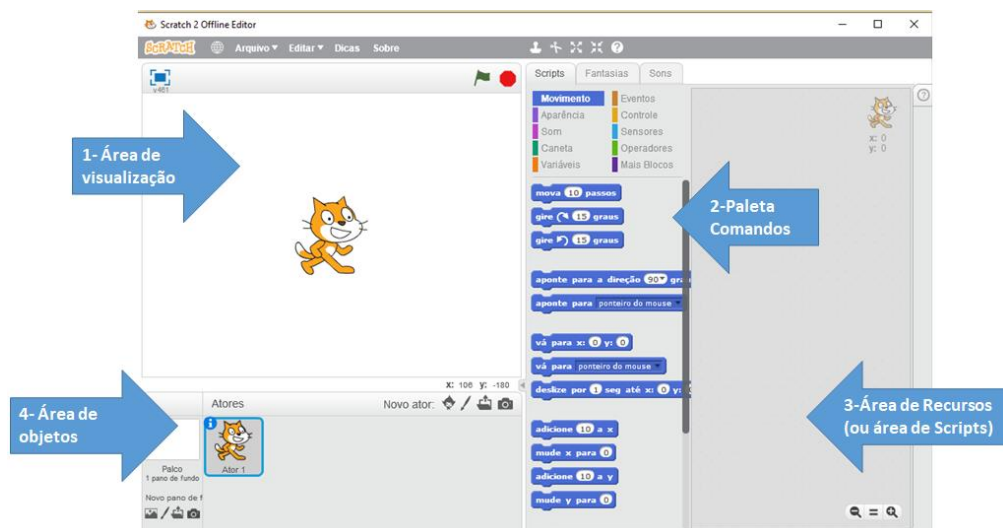
² Site SCRATCH Brasil. Disponível em: <<http://www.scratchbrasil.net.br/>>

Existe uma grande preocupação, no Brasil e em outros países, com a educação básica – em outras palavras, ensinar nossas crianças a ler, escrever, somar, subtrair. [...] O problema, entretanto, é que o mundo atual exige muito mais do que ler, escrever, adição e subtração. A lista de habilidades e conhecimentos necessários para o pleno exercício da cidadania no século XXI é tão extensa quanto controversa (Blikstein³, 2008).

Entretanto, uma indivíduo do século XXI precisa dominar muito mais que as habilidades básicas, tem que usar o computador como meio de potencializar sua capacidade cognitiva. Dessa forma, o *SCRATCH* pode auxiliar estudantes deste século a desenvolverem habilidades de pensar computacionalmente, uma vez que através de blocos programáveis, essa ferramenta auxilia ao ensino aliado a multidisciplinaridade, ensino em diversas áreas, capacitando o estudante ser mais proativo na busca de soluções de problemas simples e complexos através de simulações, testando hipóteses, aumentando a capacidade de criatividade e engenhosidade.

O *SCRATCH*, como dito, é uma ferramenta que auxilia no processo de ensino-aprendizado tanto na Matemática quanto em outras áreas. O *SCRATCH* possui em seu ambiente diversos blocos programáveis e operadores que auxiliam no aprendizado da lógica matemática, como operadores: lógico, aritmético e relacional, que permitem ao usuário criar diversas formas de códigos. Vide Figura 3 a seguir.

Figura 3: Ambiente *SCRATCH*



Fonte: Acervo Pessoal

A Figura 3 mostra o ambiente do *SCRATCH* versão *off-line* que pode ser baixado pelo site do MIT (<https://scratch.mit.edu/>) e é dividido em 4 (quatro) áreas a saber:

1 - Área de visualização: é o local em que se observa o resultado da programação criada, dito palco, onde as coisas acontecem. Aqui ocorrem todas as ações visuais e sonoras programadas pelo usuário, desde movimentações, sons, interações, exibição de valores e assim por diante.

2 - Paleta de comandos: esta área é subdividida em 3 subáreas, a saber:

A - Acima existem 3 abas: *Scripts* (área onde escolhemos os blocos para construção dos roteiros); *Fantasia* (área onde podemos editar a forma como os personagens são exibidos); e, *Sons* (área onde podemos incluir e editar os sons disponíveis para o nosso projeto). O *SCRATCH* disponibiliza uma grande biblioteca,

³ Site Paulo Blikstein. Disponível em:

<http://www.blikstein.com/paulo/documents/online/ol_pensamento_computacional.html>

contendo diversas fantasias e sons, ele também possibilita que personagens e sons possam ser importados do computador, câmera e outros.

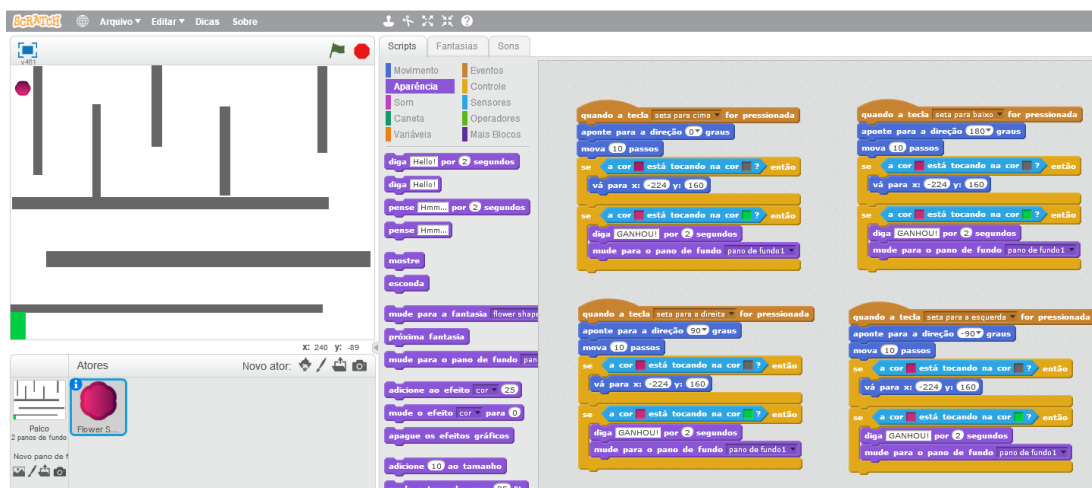
B - São os agrupamentos dos blocos disponíveis no *SCRATCH*. esta área possui dez caixas com diferentes tipos de comandos: (no ambiente *on-line*, possui 9 caixas), para facilitar a memorização e a identificação da localização dos demais blocos cada tipo de comando tem uma cor específica, cada bloco de comando possui sua própria característica, eles são divididos da seguinte forma: movimento, aparência, som, caneta, variáveis, eventos, controle, sensores, operadores e mais blocos.

C - São os elementos básicos de programação no *SCRATCH* - os blocos de comandos derivados de cada bloco de comando é necessário arrastar e agrupá-los na área de *Scripts* (3) (preste atenção nas variações de formato) construímos o algoritmo que dá ação aos nossos projetos.

3 - A área de Recursos ou de *Scripts* é para onde são arrastados os blocos de comandos que serão utilizados e onde se realiza a importação de trajes e sons para os atores. É a área onde, de fato, construímos nossos códigos. Basta arrastar os blocos desejados para essa área e começar a criar o roteiro de cada ator ou do palco.

4 - Nesta área ficam os objetos (personagens, palcos, cenários) usados na animação. O objeto selecionado nessa área poderá reagir aos comandos arrastados para a área de recursos ou *scripts* pelo usuário. São os personagens da nossa grande peça. Cada personagem é programado individualmente, mas é possível - e recomendável - programar interações entre eles. Cada ator adicionado já é incluído no palco, possibilitando que grandes criações possam ser desde animações simples e a jogos. Observe na Figura 4 uma das criações feitas no *SCRATCH*, o jogo do labirinto.

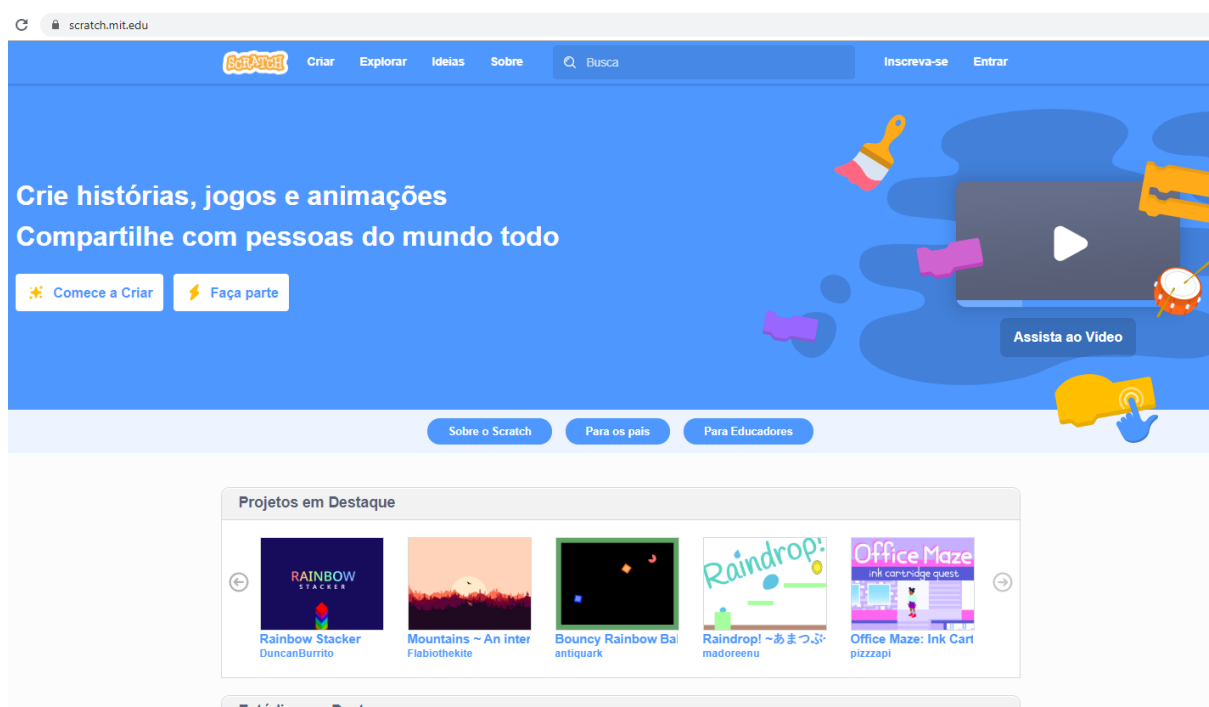
Figura 4: Jogo Labirinto



Fonte: Acervo Pessoal.

No *SCRATCH on-line* possui vários recursos como: Criar, no qual é possível criar jogos, histórias e animações, assim como no *software* do *SCRATCH*; Explorar, em que se pode explorar as criações dos outros usuários; Ideias, no qual há guias e tutoriais para criações de projetos novos e existentes, onde pode ser explorado internamente a codificação; Sobre no qual se pode ler e assistir vários vídeos sobre a história e interação com o *SCRATCH*. Além de tudo pode-se criar uma conta dentro do site e pode compartilhar suas ideias aos outros usuários. Assim como podemos ver a seguir na figura 5 a interface do site oficial.

Figura 5: *SCRATCH* on-line



Fonte: Site <https://scratch.mit.edu/>.

O *SCRATCH* é utilizado para metodologias de aprendizagem baseada em resolução de problemas e baseada em jogos digitais. A primeira, é uma técnica de ensino que leva ao aluno uma situação-problema que seja instigante e significativa, de modo a despertar a curiosidade e a participação em todo o processo para que esse problema seja resolvido, despertando neles a elaboração de processos de investigação no contexto real ou imaginário, desenvolvendo o raciocínio lógico, o pensamento crítico e a criatividade. A segunda metodologia, baseada em jogos digitais no processo de ensino e aprendizagem, desperta maior o interesse em busca de conhecimento, essa metodologia é envolvente, lúdica e desafiadora. Através da criação de jogos, desmascaram-se as dificuldades na medida em que se evolui na inserção da programação dentro do contexto, a partir do protótipo visual, inserindo os blocos de programação e visualizando as ações para solucionar uma determinada tarefa. Essas metodologias procuram abordar o conteúdo de uma forma diferente da tradicional, contribuindo para o raciocínio lógico, na tomada de decisões, análises e possíveis reformulações dos procedimentos criados durante o jogo. Também, faz com que o aluno seja o protagonista de suas próprias criações, despertando diversas habilidade e competências num contexto disciplinar e transdisciplinar.

3 METODOLOGIA

Referente à metodologia, essa investigação foi desenvolvida à luz da pesquisa exploratória e qualitativa pautados: Pensamento Computacional, *SCRATCH*, lógica matemática, entre outros assuntos que abordam sobre o tema proposto.

Com o estudo sobre o tema foi realizado uma pesquisa bibliográfica de materiais publicados em livros, artigos, entre outros materiais já realizados com a utilização do *SCRATCH* como ferramenta de auxílio no ensino superior. Realizar-se-á uma busca na internet por periódicos, teses, dissertações e artigos disponíveis para consulta. Após tal levantamento, a busca se dará por teóricos que voltam seus estudos para identificar o papel das tecnologias digitais na sociedade e no indivíduo e seu impacto na aprendizagem e no ensino.

4 CONCLUSÃO

Percebe-se que o pensamento computacional só tem a agregar benefícios a educação. A tecnologia é um forte aliado para que a educação possa ser de maior qualidade e podemos identificar isso no software do *SCRATCH* que possibilita que os seus usuários usem a criatividade e aprendam com seus erros de forma que podem criar animações, histórias interativas, jogos, coisas que são de seu interesse que prende a sua atenção e consequentemente auxilia no seu aprendizado. Possibilitando aprender de forma lúdica, criativa e interativa.

O estudante ao elaborar mentalmente estratégias e depois repassá-las em forma de comandos para o computador desenvolve seu protagonismo e no seu desenvolvimento. Pode se usar o *SCRATCH* como material didático, montado através dele formas diferentes o aprendizado de determinados assuntos como a matemática.

5 REFERÊNCIAS

ANDRÉ, Cláudio Fernando; O pensamento computacional como estratégia de aprendizagem, autoria digital e construção da cidadania. In: teccogs – **Revista Digital de Tecnologias Cognitivas**, n. 18, jul./dez. 2018, p. 94-109.

AQUINO, Bruno Macabeus Mendes de; GUEDES, Ricardo Bezerra de Menezes; AQUINO, Francisco José Alves de; Construção de fractais com a linguagem de programação Logo. **XX Encontro de Iniciação, à Pesquisa Universidade de Fortaleza**, Outubro de 2014. Disponível em: <<http://hp.unifor.br/encontros2014/PDFs/17883%20-%20Resumo.pdf>>. Acesso em: 09 out. 2019.

BLIKSTEIN, Paulo. **O pensamento computacional e a reinvenção do computador na educação**. Disponível em: <http://www.blikstein.com/paulo/documents/online/ol_pensamento_computacional.html>. Acesso em: 10 out. 2019.

JÚNIOR, Eduardo Teodoro Silva. **wxLogo**. Disponível em: <<http://algoritmo.dcc.ufpa.br/~bruno/wxlogo/docs/wxLogo.html>>. Acesso em: 08 out. 2019.

SALAZAR, Rafael; ODAKURA, Valguima; BARVINSKI, Carla. Scratch no ensino superior: motivação. **Anais do XXVI Simpósio Brasileiro de Informática na Educação (SBIE 2015)**. Disponível em: <<https://pdfs.semanticscholar.org/ca55/719fa23b543c22728fcb7562be3dea16a16a.pdf>>. Acesso em: 10 out. 2019.

Scratch Brasil. **Você Conhece o Scratch?**. Publicado em 2014 Disponível em: <<http://www.scratchbrasil.net.br/index.php/sobre-o-scratch/73-conhece-scratch.html>>. Acesso em: 11 out. 2019.

VALENTE, J. A.; ALMEIDA, F. J. Visão Analítica da Informática na Educação no Brasil: A Questão da Formação do Professor. **Revista Brasileira de Informática na Educação** – Número 1 – 1997. Disponível em: <<https://www.br-ie.org/pub/index.php/rbie/article/view/2324/2083>>. Acesso em: 12 out. 2019.

VALENTE, José Armando (org). O computador na sociedade do conhecimento. Campinas: UNICAMP/NIED, 1999.

WING, J. **Computational thinking**. Microsoft Asia Faculty Summit 26 October 2012 Tianjin, China Disponível em: <https://www.microsoft.com/en-us/research/wp-content/uploads/2012/08/Jeanette_Wing.pdf>. Acesso em: 12 out. 2019.