

Funções e programação no Scratch

André Eduardo Ventorini – PPGEMEF/UFSM - andreduardoventorini@gmail.com

Leandra Anversa Fioreze – PPGEMat/ UFRGS – leandra.fioreze@gmail.com

Resumo: Este artigo se propõe a verificar o potencial do *software* de programação Scratch no processo de construção de conceitos relacionados com funções. A pesquisa de cunho qualitativo tem como aportes teóricos os Campos Conceituais, de Vergnaud, devido a importância de sua teoria na análise do funcionamento cognitivo do "sujeito-em-situação" e o Construcionismo, de Papert, destacadamente, na relevância da programação na construção do conhecimento. A partir dos dados analisados em uma pesquisa desenvolvida com alunos do primeiro ano do ensino médio de uma escola pública, pode-se concluir que o uso do *software* de programação Scratch contribuiu na construção de conceitos relacionados com funções. Uma das contribuições está relacionada à compreensão dos esquemas utilizados no processo de criação e substituição de variáveis através de comandos específicos para esta função.

Palavras-chave: funções, teoria dos Campos Conceituais, construcionismo, Scratch.

Functions and programming in Scratch

Abstract: This article verifies the potential of the programming software Scratch in the process of building concepts related to functions. This is a qualitative research and it relies on the theory of conceptual fields, by Vergnaud, due the significance of his theory in the analysis of the cognitive working of the "subject in situation". The research relies also in the constructionism of Papert, due the outstanding importance of programming in building knowledge. Analyzing the collected data in a research developed with students of the first grade of secondary school in a public institution, we concluded that the use of the programming software Scratch contributed in the construction of the concepts related to functions. One of the contributions is related to the understanding of the schemes used in the process of creation and substitution of variables through specific instructions for this function.

Keywords: functions, theory of conceptual fields, constructionism, Scratch.

1. Introdução

O Scratch é um *software* de programação criado pelo Lifelong Kindergarten Group do Massachusetts Institute of Technology/MIT Media Lab, em colaboração com o grupo de Alan Kay da Universidade da Califórnia (UCLA), tendo sido disponibilizado para o público em geral a partir de 2007. Este *software* possui uma sintaxe mais intuitiva se comparada com outras linguagens de programação como Pascal, Python, Fortran, Cobol e C. Ao programar no Scratch, não é necessário digitar o comando, pois o mesmo vem em forma de blocos em que basta arrastá-los e encaixá-los, de acordo com uma determinada ordem lógica a ser estabelecida. Esses blocos são apresentados de forma visível, os quais facilitam a produção de histórias multimídias interativas ou qualquer outro tipo de programação (Ventorini; Fioreze, 2014). Por outro lado, para um estudante aprender a programar com essas outras linguagens, "terá que conhecer, antes de tudo, as palavras exatas para cada comando, a forma como estas palavras devem ser organizadas (sintaxe e semântica) e escrevê-las" (Lessa, 2018, p. 84).

A programação tem um importante papel no desenvolvimento da aprendizagem, pois “programar significa comunicar-se com o computador numa linguagem que tanto ele quanto o homem podem entender” (Papert, 1985, p.18). O uso do Scratch, ao possibilitar a representação de situações em forma de simulações e experimentações, pode contribuir para que os alunos criem e se apropriem de uma linguagem abstrata, distante do cotidiano.

As situações que foram criadas (pelo professor pesquisador ou pelo aluno) estão relacionadas com funções, cujas aplicações podem ter relações com o cotidiano (por exemplo, o valor a pagar em função do número de litros consumidos por um carro em um posto de gasolina), as quais envolvem conceitos abstratos (como conceitos de variável, de dependência, de regularidade, de correspondência e de generalização). Por exemplo, partindo dos dados apresentados em uma determinada situação, um modelo matemático que envolve dependência de variáveis representada por uma linguagem algébrica pode ser expresso via programação no Scratch.

Ao objetivar analisar o funcionamento cognitivo do “sujeito em situação” ao programar no Scratch, encontramos na Teoria dos Campos Conceituais um aporte teórico que fundamenta esta análise. Sua teoria envolve a aprendizagem de conceitos matemáticos e também sugere caminhos para a didática da Matemática. Pela importância dada à programação na construção do conhecimento, utilizamos o Construcionismo de Papert,

A proposta deste artigo é verificar o potencial do *software* de programação Scratch na construção de conceitos relacionados com funções, a partir de uma experiência desenvolvida com alunos do primeiro ano do ensino médio de uma escola pública localizada em uma cidade do interior do Rio Grande do Sul - RS.

2. O *software* Scratch e o Construcionismo de Seymour Papert

O matemático sul-africano Seymour Papert é considerado um dos maiores especialistas do mundo sobre como o computador pode proporcionar novas maneiras de aprender. Engajou-se em pesquisas na área da matemática e do uso da tecnologia trazendo relevantes contribuições para o desenvolvimento da aprendizagem.

Papert, na década de 60, criou a linguagem de computação LOGO. A intenção, ao criar esta linguagem era dar à criança controle sobre a mais poderosa tecnologia disponível em nossos tempos: o computador. Porém, suas ideias sobre o uso da tecnologia na educação passaram a ser incorporadas por algumas escolas somente na década de 80, quase duas décadas depois. Papert mostrou-se realmente um educador inovador e precursor de novas ideias, na medida em que suas concepções sobre a tríade computador-aluno-aprendizagem, mesmo sendo concebidas há um bom tempo, encontram-se ainda atuais.

“O Construcionismo é gerado sobre a suposição de que as crianças farão melhor descobrindo (“pescando”) por si mesmas o conhecimento específico de que precisam” (Papert, 1994, p.125). De uma forma mais direta, o Construcionismo é uma abordagem na qual a criança constrói, por intermédio do computador, o seu próprio conhecimento.

A base da programação na linguagem LOGO é a tartaruga. Definida como “objeto-de-pensar-com”, ela desempenha um suporte importante para o pensamento. É através da tartaruga que as crianças elaboram programações e exploram os diversos recursos deste ambiente, aventurando-se em seus micromundos. Na linguagem de programação do *software* Scratch, os “objetos-de-pensar-com” são os personagens e os cenários.

Na tentativa de “ensinar” a tartaruga (LOGO) ou o gato (Scratch), por exemplo,

a desenhar um quadrado, a criança acaba descrevendo, passo a passo, as propriedades do polígono, como seus ângulos internos e os movimentos necessários para realizar o desenho (figura 1).

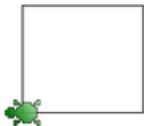

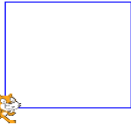
LOGO	SCRATCH
<p>Programação</p> <p>Aprenda quadrado</p> <p>Abaixe a caneta</p> <p>Repita 4 [pf 100 pd 90]</p> <p>Fim</p> <p>Visualização</p> 	<p>Programação</p>  <p>Visualização</p> 

Figura 1 - Comparativo entre as linguagens de programação LOGO

Ao programar, temos a descrição (através das sequências lógicas dos comandos de programação) do processo utilizado pela criança para resolver uma tarefa, possibilitando a reflexão sobre o mesmo. Na tela do computador, a criança pode perceber o que ela programou, e, se necessário, reformular suas ideias a fim de alcançar seu objetivo. Isto é, “[...] ao ensinar um computador a ‘pensar’, a criança embarca numa exploração sobre a maneira como ela própria pensa [...]” (Papert, 1985, p. 35).

3. A Teoria dos Campos Conceituais

A teoria dos Campos Conceituais foi desenvolvida pelo psicólogo e pesquisador Gerard Vergnaud, discípulo e aluno de Jean Piaget. Vergnaud é doutor Honoris Causa da Universidade de Genebra sendo fundador da Escola Francesa de Didática da Matemática. Vergnaud é um dos fundadores do Instituto de Pesquisa sobre o Ensino de Matemática (IREM) nas Universidades da França, na década de 60 (Fioreze, 2010).

Vergnaud (1993) define campo conceitual como um conjunto de problemas e situações cujo tratamento requer conceitos, procedimentos e representações de tipos diferentes, mas intimamente relacionados entre si. A teoria dos Campos Conceituais busca compreender, além dos processos de conceitualização, as construções das estruturas cognitivas do pensamento.

Vergnaud (1993) define conceito como uma trinca (S, I, R) sendo:

S: Conjunto de situações que dão sentido ao conceito (referência).

I: Conjunto das invariantes em que se baseia a operacionalidade dos esquemas (significado).

R: Conjunto das formas de linguagem que permitem representar simbolicamente o conceito, suas propriedades, as situações e os procedimentos de tratamento (significante).

A construção de um conceito envolve uma relação entre os três elementos da trinca (S, I, R), ou seja, as situações em que estes conceitos aparecem, os esquemas que são

necessários para resolver a situação (significado) e as diferentes formas de representação do conceito (significante). Por exemplo, dentre as formas de linguagem verbal, gráfica e algébrica, pode-se acrescentar a programação, que através da organização dos blocos de comandos, expressam suas ideias, pensamentos e esquemas.

Vergnaud (1993) define esquema como uma organização invariante do comportamento do sujeito em função das características particulares de cada situação, sendo que o esquema abrange:

- invariantes operatórios (conceitos em ação e teoremas em ação): são os conceitos contidos nos esquemas. Teorema em ação é uma proposição considerada como verdadeira sobre o real; conceito em ação é uma categoria de pensamento considerada como pertinente (Vergnaud, 1993).

Exemplo de conceito em ação (conforme figura 2): partindo do pressuposto de que o aluno já construiu uma programação que levasse um personagem “aparecer” no primeiro quadrante, utilizando-se para isso o comando “ $x > 0$ e $y > 0$ ”, ele não precisa, necessariamente, realizar outra programação para perceber que qualquer objeto, para “aparecer” no quarto quadrante de um plano cartesiano, precisaria ter a condição “ $x > 0$ e $y < 0$ ” como parte de sua programação.



Figura 2- Blocos de comando da linguagem Scratch (Ventorini, 2015)

- regras de ação do tipo “se... então...”: permitem gerar sequências de ações pelo aluno que objetivam controlar os resultados. Na figura 3 há um controle dos resultados da ação (uso de “se... então...”) para fazer com que um personagem ou objeto “deslize em 1 segundo para as coordenadas x e y ”, se $y = 2x + 10$:

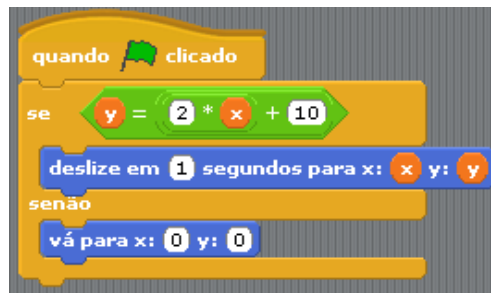


Figura 3 - Blocos de comandos com o uso do “se... então...” (Ventorini, 2015)

- inferências: raciocínios que permitem calcular as regras e as antecipações a partir das informações e dos invariantes operatórios de que dispõe o sujeito, os quais geram uma série de ações para se atingir o objetivo.

- metas a atingir e antecipações: são os efeitos esperados e as eventuais etapas intermediárias. As antecipações são muito exigidas quando se realiza uma programação; o uso de certos comandos como: “vire tantos graus”, “vá para”, “espere até”, estão intimamente ligados a uma antecipação do movimento que eles podem gerar e da importância desse movimento para atingir a meta desejada.

4. Delineamentos Metodológicos

A presente pesquisa possui cunho qualitativo à medida que interessa investigar o potencial do *software* Scratch através de situações que envolvem a programação, na construção de conceitos de função. Sobre a pesquisa qualitativa, Bicudo (2012) cita que este é um tipo de pesquisa onde “trabalha-se com a qualidade”, ou seja, “a qualidade é do objeto e passível de ser observada” (p.17). Visando o desenvolvimento da pesquisa, objetivamos compreender os processos envolvidos nas programações através de descrições sistemáticas e detalhadas.

A pesquisa foi realizada com três turmas de primeiros anos do Colégio Estadual Manoel Ribas do município de Santa Maria – RS. Os sujeitos, quase na totalidade, não tiveram acesso em suas escolas anteriores a *softwares* e/ou aplicativos matemáticos, tampouco a *softwares* de programação.

A coleta de dados da pesquisa se deu através da utilização de um diário de bordo e das gravações das programações produzidas pelos alunos. O diário de bordo foi uma maneira de registrar os comentários as observações e os questionamentos feitos pelos alunos oralmente. Por fim, se utilizou a fotografia, com o objetivo de registrar a construção das programações.

As atividades desenvolvidas com os alunos se deram através de duas etapas. A primeira etapa teve como objetivo o “conhecimento” e a “exploração” da linguagem de programação Scratch. A segunda etapa aconteceu em dois ambientes, ou seja, na sala de aula e no laboratório de informática, com situações que enfocaram os seguintes conceitos matemáticos específicos: a ideia de função, função inversa, caso particular de domínio da função e plano cartesiano.

Dar-se-á ênfase, considerando o objeto de pesquisa, às situações propostas na segunda fase desenvolvidas no laboratório de informática, que foram em número de quatro. Dentre estas, temos uma situação em aberto, ou seja, o aluno poderia criar algo que fosse de seu interesse e que envolvesse conceitos relacionados com funções. Destaca-se a importância da construção de um produto de significado pessoal para o aprendizado dos alunos (Maltempi, 2004).

Visando analisar detalhadamente as produções dos alunos, o critério adotado foi considerar uma dupla de cada turma que participou da pesquisa, selecionando as duplas que tiveram maior frequência nos encontros e as programações que estão mais completas. As duplas foram identificadas como E24,E28; F2,F9 e G3,G6, respectivamente, e pertencentes às turmas E, F e G.

5. Resultados e Discussão

Devido ao espaço delimitado para este artigo, apresentamos uma das situações propostas a fim de explicitar a análise do processo de construção dos conceitos pelos alunos nas programações no Scratch, bem como as dificuldades envolvidas neste processo. A seguinte situação foi apresentada aos alunos:

“Pedro é taxista e cobra R\$ 1,50 por km rodado, além de R\$ 5,00 fixos pela bandeirada. Quanto Pedro irá cobrar de um cliente que andar 30 km em seu táxi? E para um número qualquer de km?”

Ao programar, para resolver a situação, os alunos deverão relacionar a cada valor atribuído para a variável “x” (km rodados), um único valor correspondente para a variável “y” (valor a pagar em reais), representando essa correspondência, por exemplo, através de um modelo matemático. Os personagens e cenários foram de escolha dos alunos.

Segue a produção de uma das duplas (E24 e E28) (Figura 4):

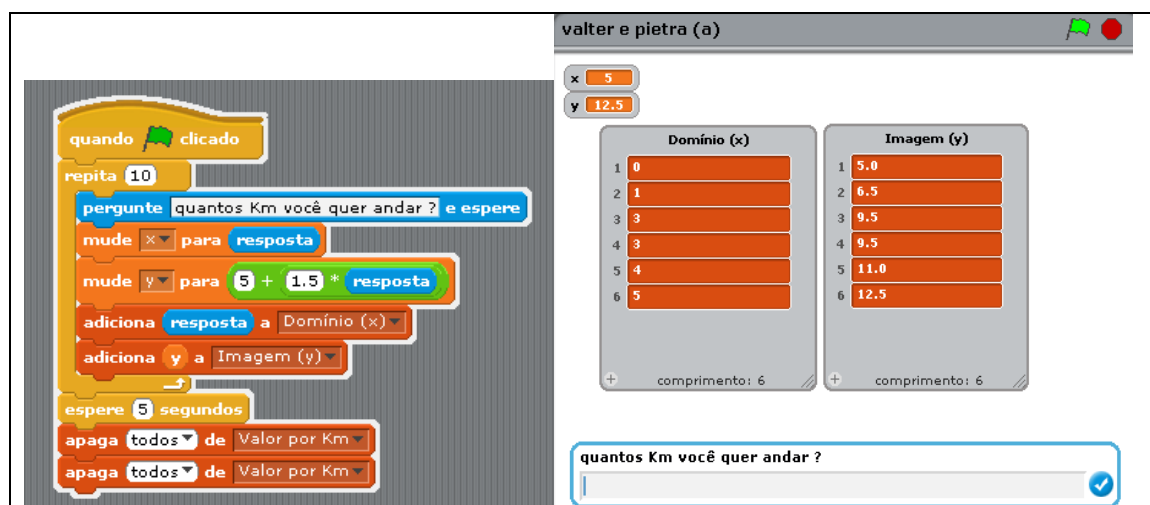


Figura 4 - Produção da dupla de alunos E24,E28 (Ventorini, 2015)

A dupla de alunos E24, E28 indicou através do comando “criar uma variável” as variáveis “x” e “y” para representar, respectivamente, os quilômetros rodados e os valores a pagar. Houve uma articulação lógica entre os comandos “pergunte... e espere” que é a entrada de valores para a variável “x” no comando “resposta” que corresponde ao valor digitado pelo usuário e o comando “mude x para...” que substitui a variável “x” pelo valor digitado pelo usuário. Raciocínio semelhante se deu quando da utilização do comando “mude y para...” que objetivou substituir a variável “y” pelo valor do resultado vindo da expressão algébrica “ $5 + 1,5 \cdot \text{resposta}$ ” que foi estruturada por eles.

Para construir os conjuntos (tabelas) do “domínio” e da “imagem” da função, a dupla utilizou o comando “cria uma lista” denominando o conjunto do domínio da função por “domínio x” e o conjunto imagem da função por “imagem y”, conforme é observado na figura 5.

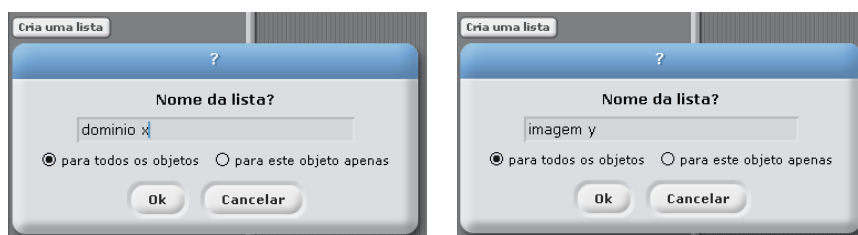


Figura 5 - Comandos do Scratch com a função de criar tabelas (listas) (Ventorini, 2015)

A programação atribui para a lista “domínio x” os valores digitados pelo usuário e para a lista “imagem y” os valores resultantes do comando “mude y para...”. Por fim, o comando de recursividade “repita 10” gerou o número de vezes (neste caso específico, dez vezes) que o usuário pode “entrar” (atribuir) com valores para a variável “x”.

Uma das fases da programação que mais requereram esforço e atenção dos alunos foi a compreensão do processo de substituição de variáveis na linguagem do Scratch. Os questionamentos e diálogos a seguir entre o professor pesquisador e alunos

da turma E dão a noção destas dificuldades:

- E28: Onde é que vai o comando verde? (o aluno se referiu à expressão algébrica construída por eles).
- Professor: O que representa para você o comando verde?
- E28: A fórmula que dá o valor de trinta quilômetros.
- Professor: E se quiséssemos saber o valor que pagaríamos por quarenta quilômetros, o que precisaríamos fazer?
- E28: Apagar o trinta e colocar o quarenta no lugar dele.
- Professor: E para cem quilômetros, o que deveríamos fazer então?
- E28: A mesma coisa.
- Professor: Como poderíamos então estruturar esta fórmula de maneira que cada aluno pudesse colocar o valor que desejasse?

(Diálogo entre professor e aluno, Ventorini, 2015)

Após alguns minutos o aluno E24 sugeriu colocar o comando “resposta” na expressão algébrica, pois era o valor que as pessoas iriam atribuir para os quilômetros rodados. De modo análogo, se discutiu outras substituições pertinentes nos outros comandos da programação, como por exemplo, “adiciona resposta a domínio x” para adicionar ao conjunto (lista) dos “x” os valores digitados pelos usuários.

Algumas duplas tiveram dificuldades na construção das expressões, pois os comandos “operadores” precisam ser encaixados respeitando a ordem hierárquica das operações. Um exemplo de erro é o da dupla G3, G6, da turma G, que fez sua construção de forma equivocada ao construir a expressão numérica “5 + 1,5.30”. Da forma como estavam estruturados os comandos, primeiro acontecia a operação adição para só depois acontecer a operação multiplicação com o número trinta (km), o que originava uma resposta de cento e noventa e cinco (reais).

A figura 6 explicita a estrutura construída pela dupla de alunos G3, G6:

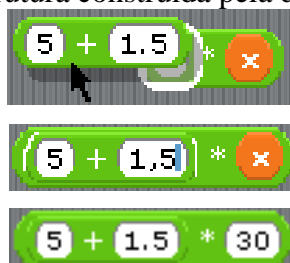


Figura 6 - Expressão aritmética organizada pela dupla G3,G6 na atividade 1 com os comandos do Scratch (Ventorini, 2015)

O esquema desenvolvido inicialmente pelos alunos G3 e G6 descrito na figura 6 não foi eficiente, pois não levou à uma correspondência lógica entre os km rodados e o valor a pagar. Segundo Vergnaud (1993) os esquemas, são em geral, eficazes, mas nem sempre efetivos. “Quando a criança utiliza um esquema ineficaz para determinada situação, a experiência a leva, seja a mudar de esquema, seja a modificar o esquema” (p. 3). Foi o que aconteceu com a dupla ao modificar o esquema.

As conversas, na forma de questionamentos partindo do professor pesquisador, acerca do problema, auxiliaram as duplas de alunos na compreensão, escolha e encaixes de certos comandos. Os questionamentos eram feitos de maneira geral, o que os incentivava discutir entre si esclarecendo alguns fatos para os colegas e permitindo que quem houvesse compreendido explicasse para os demais. É pertinente ressaltar, a importância que Vergnaud dá, didaticamente, ao papel da mediação no processo do

ensino e da aprendizagem:

[...] atos de mediação do professor vão ter que acompanhar, pelo menos de vez em quando, o trabalho do aluno, jamais totalmente autônomo, seja o que for aquilo que desejamos. Esses atos de mediação tomam muitas vezes a forma de palavras, mas somente; um gesto do dedo apontando para o erro pode significar tanto quanto uma explicação. Contudo, as explicações são indispensáveis, e a forma predicada do conhecimento vem então socorrer a forma operatória que é o “esquema de pensamento”, com uma função vizinha, guardadas as proporções, daquela da linguagem da criança que aprende a falar, e cujas primeiras palavras e enunciados vem socorrer as primeiras invariantes operatórias construídas por ela, objetos e relações. (Vergnaud, 2008, p.31).

Em relação à utilização do comando “repita” nas construções dos objetos de aprendizagem desta atividade, surgiu do aluno F2 um questionamento bastante pertinente, que resultou no seguinte diálogo:

- F2: *Eu posso colocar número negativo dentro dele?*

- Professor: *O que representa o número que vai dentro deste comando, nesta nossa atividade?*

- F2: *As “vezes” que vai aparecer a pergunta para colocar o valor de “x”.(o aluno se referiu ao número de oportunidades que o usuário poderá atribuir, neste caso, para valores que correspondem à variável “x”).*

O aluno, em ato contínuo a sua colocação nem esperou outra intervenção do professor para responder seu próprio questionamento:

- F2: *Não, não pode ser negativo, porque é o número de vezes para digitar.*

(Diálogo entre professor e aluno, Ventorini, 2015)

Uma das classes de situações propostas por Vergnaud (1993, p. 6) é aquela em que o sujeito não dispõe de todas as competências necessárias ao tratamento imediato da situação, o que o obriga a um tempo de reflexão e exploração, a hesitações, a tentativas frustradas, levando-o eventualmente ao sucesso ou ao fracasso. É o que parece estar explícito nesta construção das programações tendo em vista os questionamentos dos alunos presentes nos diálogos com o professor pesquisador.

6. Considerações Finais

O objetivo deste artigo foi verificar o potencial do Scratch no processo de construção de conceitos relacionados com funções através da programação, apresentando uma forma alternativa de trabalhar a matemática na escola. A partir dos dados analisados, pode-se concluir que a utilização do *software* de programação Scratch na construção de conceitos relacionados com funções oportunizou aos alunos fazer antecipações, controlar resultados, modificar esquemas, fazer inferências.

Uma potencialidade que pode ser atribuída ao *software* Scratch, foi a compreensão dos processos de criação e substituição de variáveis através de comandos específicos para esta função. Outra potencialidade verificada é que as características da linguagem de programação, aliada ao conteúdo função, possibilitou que os alunos se apropriassem de algoritmos (programações) que contemplassem não só uma única situação, mas várias situações, contribuindo para o processo de generalização.

Durante o desenvolvimento da programação, na montagem das expressões algébricas, os erros acontecidos permitiram que os alunos reconstruíssem seu pensamento analisando o seu processo de construção, até porque, “[...] quando se

aprende a programar um computador dificilmente se acerta na primeira tentativa” (PAPERT, 1985, p. 39). Para Papert (1985, p. 141), “os erros são benéficos porque levam os alunos a estudarem o que aconteceu, a interpretar o que deu errado, e, através do entendimento, corrigi-lo” e o Scratch parece ter contribuído, através das características da sua linguagem, neste processo.

O processo sequencial utilizado na programação com o Scratch, permitiu que os alunos testassem suas hipóteses ao realizarem um determinado procedimento, confrontando os resultados imediatamente a partir do *feedback* dado. O retorno imediato na tela do computador possibilita refletir sobre a ação “e se necessário, a reformulação das hipóteses iniciais para testá-las novamente. É um processo em espiral, em que se pode perceber, nas representações nos códigos, as descrições do pensamento e os diversos conhecimentos em ação” (LESSA, 2018, p. 3).

Esta pesquisa apresenta um olhar diferenciado na busca por alternativas para contribuir para a aprendizagem de matemática dos alunos apresentando a programação e como alternativa para o desenvolvimento deste processo. Partindo de situações a serem programadas com o Scratch, os erros são permitidos e considerados como parte importante do processo, e que tiveram ainda como intenção desenvolver no aluno a autonomia e a compreensão de como ele está pensando ao programar.

Para finalizar, a satisfação ao findar essa pesquisa, aponta para uma aproximação ainda maior no tema programação e educação, pois denota um caminho a ser percorrido e explorado, que se abre para inúmeras possibilidades a cada passo que se dá rumo a sua compreensão.

Referências Bibliográficas

BICUDO, Maria Aparecida Viggiani. A pesquisa em educação matemática: a prevalência da abordagem qualitativa. **Revista Brasileira de Ensino de Ciências e Tecnologia**. Curitiba, vol 5, n. 2, mai./ago. 2012.

FIGLIARELLI, L. A.. **Atividades digitais e a construção dos conceitos de proporcionalidade: uma análise a partir da teoria dos campos conceituais**. Porto Alegre: PPGIE/UFRGS, 2010, 245 p.. Tese de Doutorado.

LESSA, V.. **A programação de computadores e a função afim: um estudo sobre a representação e a compreensão de invariantes operatórios**. Passo Fundo: PPGEdU/UPF, 2018, 184 p.. Tese de Doutorado.

MALTEMPI, M. V. Construcionismo: pano de fundo para pesquisas em informática aplicada à educação matemática In: BICUDO, M. A. V.; M. C. BORBA (Orgs.). Educação Matemática: pesquisa em movimento. São Paulo: Cortez, p. 264-282, 2004.

PAPERT, S. **A Máquina das Crianças: Repensando a Escola na Era da Informática**. Porto Alegre: Artes Médicas, 1994.

PAPERT, S. **LOGO: computadores e educação**. São Paulo: Brasiliense, 1985.

VENTORINI, A. **Construção de relações funcionais através do software Scratch**. Santa Maria: PPGEMEF/UFSM, 2015, 168 p.. Dissertação de Mestrado Acadêmico.



VENTORINI, A. E.; FIOREZE, L. A.. O software Scratch: uma contribuição para o ensino e a aprendizagem da matemática. In: **IV ESCOLA DE INVERNO DE EDUCAÇÃO MATEMÁTICA**, 2014, Santa Maria. Anais. Santa Maria: UFSM, 2014. ISSN 2316-7785.

VERGNAUD, G. **Atividade humana e conceituação**. Porto Alegre: GEEMPA, 2008.

VERGNAUD, G. Teoria dos campos conceituais. In: **ANAIS DO 1º SEMINÁRIO INTERNACIONAL DE EDUCAÇÃO MATEMÁTICA DO RIO DE JANEIRO**, 1993, Rio de Janeiro. Anais. Rio de Janeiro: Projeto Fundação - Instituto de Matemática - UFRJ, 1993. p.1 – 26.