



# VII CONGRESSO INTERNACIONAL DE ENSINO DA MATEMÁTICA

ULBRA – Canoas – Rio Grande do Sul – Brasil.

04, 05, 06 e 07 de outubro de 2017

Comunicação Científica

## CONTRIBUIÇÕES DA FERRAMENTA GRÁFICA *BLOCKLY* NO PROCESSO ENSINO-APRENDIZAGEM NA DISCIPLINA DE ALGORITMOS

Carmen Vera Scorsatto Brezolin<sup>1</sup>

Neiva Ignês Grandó<sup>2</sup>

### Educação Matemática, Tecnologias Informáticas e Educação à Distância

**Resumo:** Algoritmos é uma disciplina na qual são abordados os conceitos fundamentais de programação de computadores, considerados essenciais para os alunos que desejam seguir um curso de Tecnologia da Informação. Nessa disciplina, exige-se do aluno elevada capacidade de raciocínio lógico e de abstração, sabendo que para escrever os algoritmos são necessários os conceitos da matemática, uma vez que se fazem necessários o uso dos seus símbolos, operações e equações para que esses funcionem corretamente. Sendo assim a Matemática é essencial na construção da maioria dos softwares. Nesse contexto, este artigo apresenta algumas reflexões referente a pesquisa que se propôs a investigar as potencialidades do uso de uma ferramenta gráfica (*Blockly*) como estratégia de ensino e de aprendizagem na disciplina de Algoritmos. Participaram do estudo alunos e a professora da disciplina de Algoritmos, do Curso Superior em Sistemas para a Internet do Instituto Federal de Educação, Ciência e Tecnologia Sul Rio-Grandense/Ifsul, do campus de Passo Fundo/RS. Os dados da pesquisa foram constituídos de registros das atividades realizadas pelos alunos com o uso da ferramenta gráfica, baseadas na observação do “comportamento”, das “atitudes” e dos procedimentos adotados pelos alunos e pela professora no decorrer do processo investigativo. Constatou-se que o uso de um software com o intuito de auxiliar no processo de ensino-aprendizagem de algoritmos constitui os “saberes” necessários para ensinar.

**Palavras-chave:** Matemática. Algoritmos. Ferramenta gráfica. *Blockly*. Ensino Superior.

### INTRODUÇÃO

A disciplina de Algoritmo é um dos pilares do Curso Superior de Tecnologia em Sistemas para a Internet (TSPI). Nessa disciplina, o aluno deverá estudar os conceitos necessários para desenvolver as bases do raciocínio lógico, voltado ao contexto das linguagens de programação, o que é fundamental para seu êxito no curso como um todo. É nessa disciplina que os alunos são introduzidos aos princípios da lógica e da programação de computadores, áreas que estabelecem o embasamento dos principais conceitos sob os quais a Ciência da Computação está fundada.

<sup>1</sup> Mestre em Educação/Universidade de Passo Fundo. Instituto Federal Sul-rio-grandense, Campus Passo Fundo. carmen.scorsatto@passofundo.ifsul.edu.br.

<sup>2</sup> Mestre em Psicologia Cognitiva/UFPE; Doutora em Educação/UFSC. Universidade de Passo Fundo/RS. Líder do Grupo de Pesquisa Teoria Histórico-cultural e Educação Matemática/CNPq. neiva@upf.br.

Um algoritmo é a definição de uma atividade que desejamos que o computador execute ou uma série de passos que devem ser executadas com uma determinada ordem pré-definida pelo programador. O papel do programador é intermediar o diálogo entre o computador e o usuário, portanto para fazer com que esse diálogo ocorra, ou seja, para que o usuário e o computador se entendam é preciso um *software* (um programa). Um *software*, por sua vez, é construído com algoritmos que, escritos em uma linguagem de programação, tornam possível a execução (da tarefa) em um computador ou qualquer dispositivo computacional. Assim, por meio desses *softwares* desenvolvidos com algoritmos, temos a impressão que estamos dando vida às máquinas.

Entretanto para escrever estes algoritmos precisamos dos conceitos da matemática, pois necessitamos fazer uso dos seus símbolos, operações e equações para que esses funcionem corretamente. Sendo assim a Matemática é essencial para o uso correto dos computadores na construção da maioria de seus *softwares*, ou ainda, utilizamos esta “linguagem” matemática para determinar o que deve ser feito, como e quando nestes algoritmos.

Embora os alunos já tenham noções matemáticas oriundas da educação básica, encontram muitas dificuldades para a sua representação no ambiente computacional. Nesse sentido, observamos que, historicamente, nessa disciplina acontece um alto índice de evasão e reprovação. A inquietação acerca das dificuldades apresentadas pelos alunos e os seus altos índices de reprovação e evasão, obrigaram-nos a repensar a prática adotada até então. Entretanto, em busca de respostas para as inquietações advindas da prática docente, iniciamos a busca por ferramentas gráficas que contribuíssem para os processos de ensino e de aprendizagem de algoritmos.

Ferramenta gráfica para o ensino de algoritmos é um programa que, ao invés de utilizar linhas de códigos, na forma de texto, quase sempre no idioma inglês, utiliza-se de objetos gráficos para montar o algoritmo. Geralmente essas ferramentas podem tornar o processo de ensinar conceitos básicos de programação mais acessível por meio da sua utilização, por apresentarem elementos de arrastar-e-soltar e, também, por terem abordagem visual.

Com base nas razões explicitadas, ou seja, nas dificuldades do processo de ensino-aprendizagem na disciplina de Algoritmos e na falta de pesquisas que pudessem orientar esse processo, definimos o tema de pesquisa: A relação entre o

estudo de Algoritmos e o uso de ferramentas gráficas. Assim, definimos como pergunta principal: Em que medida o uso de uma ferramenta gráfica de programação pode qualificar os processos de ensino e de aprendizagem na disciplina de Algoritmos?

Por meio da aplicação da ferramenta surgiram reflexões que ultrapassam a aplicação de um *software*, pela observação do “comportamento”, das atitudes e procedimentos adotados pelos alunos e pela professora durante esse processo. Nossas percepções indicam que o uso de um *software* com o intuito de auxiliar no processo de ensino e aprendizagem, mesmo que na área técnica de algoritmos, para construção de *software* apontam para um indispensável entendimento sobre responsabilidades e “saberes” necessários para o exercício da docência.

Optamos por apresentar esse texto a um Congresso de educação Matemática por entendermos que tanto a disciplina de Algoritmos como a ferramenta utilizada tem estreita relação com conteúdos de matemática.

### **EM BUSCA DE UM REFERENCIAL PARA O ENSINO DE ALGORITMOS**

Compreendendo a importância da “linguagem” matemática para determinar o que deve ser feito, como e quando nos algoritmos nos direcionou para a busca de um referencial na didática da matemática, para assim tentar compreender melhor as dificuldades apresentadas pelos alunos na construção de algoritmos.

A Didática da Matemática, como teorização, iniciou na década de 1970, na França, com intuito de sistematizar os estudos acerca do ensino da matemática. Alguns dos principais autores foram de Guy Brousseau em 1976 e Regine Douady em 1984. É conceituada por Passos e Teixeira como “a arte de conceber e conduzir condições que podem determinar a aprendizagem de um saber matemático por parte de um sujeito” (2013, p. 157). Apresentam a definição de Didática exposta por Brousseau (1986) como uma relação específica entre os conteúdos de ensino, a maneira como os alunos adquirem conhecimentos e os métodos. Ainda asseveram que Brousseau desenvolveu uma teoria para compreender as relações que acontecem entre os alunos, o professor e o saber em sala de aula. Tal teoria é conhecida como Teoria das Situações Didáticas, em que alunos e professores são atores indispensáveis da relação de ensino e aprendizagem, bem como o meio em que a situação didática se faz presente.

Almouloud (2007) menciona que o objeto central de estudo da Teoria das Situações Didáticas não é o sujeito cognitivo, mas a situação didática na qual são identificadas as interações estabelecidas. Interações estas que visam estabelecer um ambiente favorável para a aprendizagem, de modo que competiria ao educador promover situações que permitam ao aluno construir sua aprendizagem, sem perder de vista a cientificidade dos conhecimentos propostos. Sobre esses conhecimentos, existe certa preocupação referente às diversas transformações ou adaptações que “um conhecimento científico” sofre desde a sua elaboração até a socialização e a apropriação desse pelos alunos. Tais transformações constituem o processo denominado de transposição didática, assim contextualizada por Chevallard (2000):

Um conteúdo escolar que tenha sido designado como saber a ensinar, sofre desde então um conjunto de transformações adaptativas que irão torná-lo apto para ocupar lugar entre os objetos de ensino. O trabalho que transforma um objeto de saber a ensinar em um objeto de ensino é denominado de transposição didática (p. 65, tradução nossa).

Para Grandó (2000), as transformações desse conhecimento podem ocorrer em momentos diferentes e ocorrem na produção e na publicação, na seleção do que será levado para a escola, naquilo que é, efetivamente, ensinado e no que é, efetivamente, internalizado pelo aluno. Com base na obra de Chevallard, a autora apresenta uma reflexão sobre até que ponto os professores podem adaptar esses conhecimentos, sendo que não podem ficar tão afastados do saber científico a ponto de torná-lo banalizado. Essa reflexão traz à tona a permanente vigilância que o professor precisa desenvolver para que, na angústia de ver as dificuldades de seus alunos solucionadas, não acabe mudando o conhecimento.

Ainda sobre transposição didática, Pais (2008) aponta elementos importantes que falam do processo da preparação prévia pelo qual passa o conteúdo a ser ensinado, ao que chama de “textualização do saber” (p. 32). O autor ainda destaca duas variáveis que considera como fundamentais quando se trata de programação de ensino, quais sejam: o tempo didático e o tempo de aprendizagem. Tempo didático “aquele marcado nos programas escolares e nos livros didáticos em cumprimento a uma exigência legal” (p. 33); e tempo de aprendizagem que “está mais vinculado com as rupturas e os conflitos do conhecimento, exigindo uma permanente reorganização de informações e que caracteriza toda a complexidade do ato de aprender” (p. 34).

O tempo didático admite que seja possível enquadrar o conteúdo num determinado espaço de tempo, preocupando-se mais em cumprir o programa proposto do que com a aprendizagem. Por sua vez, o tempo de aprendizagem respeita o tempo de cada sujeito, aquele necessário ao sujeito para superar seus bloqueios.

Na concepção de Pais (2008) “a superação da distância entre o tempo de aprendizagem e o didático passa por uma retomada constante das noções já estudadas, nas mais variadas situações, sempre buscando novos níveis de formalização das atividades” (p. 35). O autor menciona que para compreender melhor o entrelaçamento entre esses dois tempos “é necessário voltar à outra especificidade do ensino da matemática, que é a resolução de problemas” (p. 35).

## **REFLEXÕES SOBRE A UTILIZAÇÃO DA FERRAMENTA**

Com base nos estudos realizados sobre ferramentas gráficas que contribuíssem para os processos de ensino e de aprendizagem de algoritmos, foram escolhidas para avaliação na pesquisa o *Alice*, *Scratch* e *Blockly*.

*Alice* tem ambiente de programação 3D, iniciado em 1999 na Carnegie Mellon University, o que torna mais fácil o desenvolvimento de animações e jogos interativos. É uma ferramenta de ensino projetada para auxiliar a introdução dos estudantes na programação orientada a objetos e possibilita que os alunos aprendam conceitos fundamentais de programação no contexto de criação de filmes animados e videogames simples (ALICE, 2008).

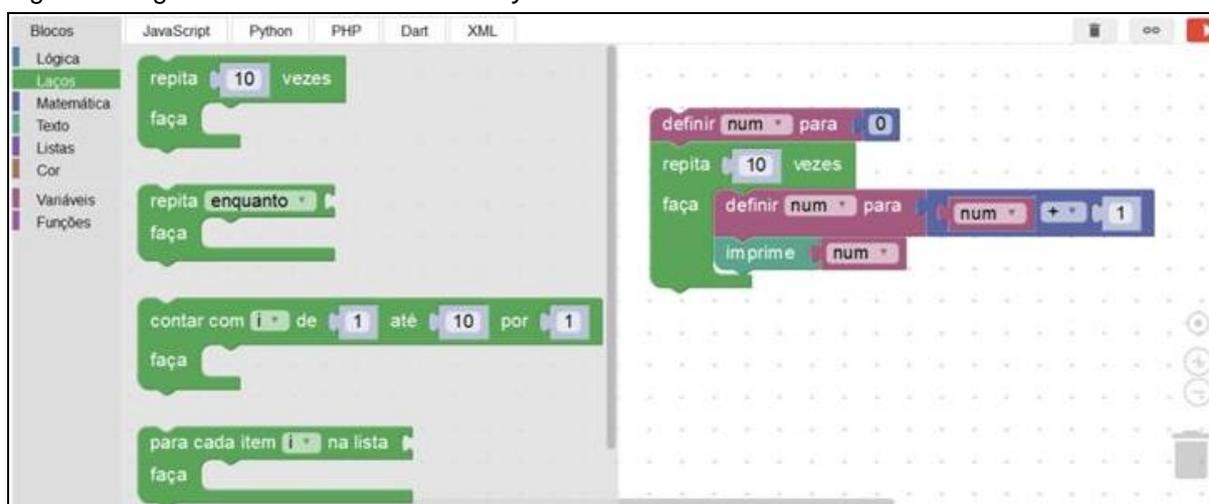
O *Scratch* foi criado pelo MIT no Lab's Lifelong Kindergarten, como uma ferramenta para crianças. Ao invés de solicitar que os usuários escrevam códigos, o *Scratch* faz uso de peças de um quebra-cabeça, que representam diferentes conceitos de programação de computadores, como *loops* e variáveis. Os usuários arrastam as peças para criar jogos e projetos interativos. As peças do quebra-cabeça se encaixam apenas em combinações selecionadas para criar um programa, dessa forma, são ensinados, aos usuários, os conceitos básicos, centrais para a programação em qualquer linguagem, apesar de ter sido, inicialmente, desenvolvido para crianças.

O Google *Blockly*, descrito por Spina (2013), como um editor de programação gráfica, o qual faz uso de blocos semelhantes às peças encontradas no *Scratch* para

criar programas. O que o diferencia do *Scratch* é o fato de que os programas que são criados com a utilização dessa ferramenta podem ser exportados para outras linguagens como JavaScript, Python, ou código XML, o que pode ajudar os usuários a fazer a conexão entre a interface gráfica e as outras linguagens de programação. Todo o código fonte é aberto, tornando possível, aos educadores, a utilização em seus próprios projetos.

Segundo a Google Developers (2015), outra característica importante é que o *Blockly* em si não é uma aplicação educativa, é um editor que pode ser usado como parte de uma ferramenta. Os autores afirmam que, atualmente, está sendo usado como um editor visual por centenas de projetos na sua maioria de natureza educacional. O *Blockly* organiza os comandos por grupos e esses têm uma cor característica para facilitar sua localização, assim como no *Scratch*. Apresentamos, a seguir, (Figura 1) um algoritmo criado para exemplo, construído utilizando o *Blockly*.

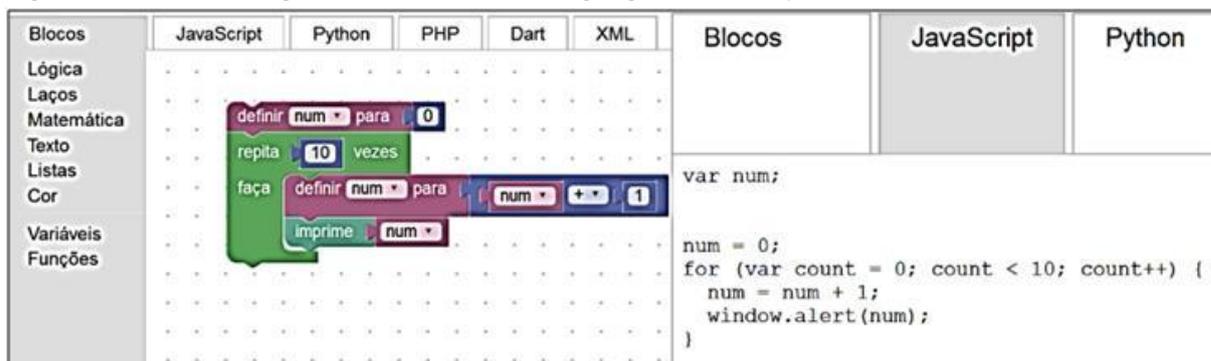
Figura 1 - Algoritmo construído com *Blockly*



Fonte: BREZOLIN, 2016, p. 45.

Além dos blocos para a construção dos Algoritmos, o *Blockly* tem como saída códigos em uma linguagem de formato textual, ou seja, é capaz de gerar em linguagens como Javascript, Python, PHP, Dart e XML, com base nos blocos do algoritmo construído, como ilustrado na Figura 2. O comando inserido na guia Blocos, automaticamente, converterá e atualizará as outras guias das cinco linguagens de programação.

Figura 2 - Trecho de Algoritmo visualizado na linguagem JavaScript



Fonte: BREZOLIN, 2016, p. 46.

Sobre o *Blockly*, localizamos exemplos de projetos que a utilizam com fins educativos e com o objetivo de instigar nos alunos a vontade de programar. Segundo a revista *Olhardigital* (2013), um dos projetos é o *code.org*, objetivando difundir o ensino de programação, apresentando os conceitos de algoritmos na forma de desafios, com o estilo de programação de arrastar e soltar.

Percebemos que os alunos na sua maioria pretendem entrar rapidamente no mercado de trabalho, por essa razão, acreditamos que eles se sentiriam mais motivados em aprender uma ferramenta semelhante com as utilizadas na programação formal. Acreditamos que com a ferramenta *Blockly* os alunos estariam aptos a utilizá-la para construir suas próprias linhas de códigos, com a possibilidade de converter para linguagem textual e inserir em seus futuros programas.

Um aspecto relevante observado durante a análise dos dados da pesquisa foi a dificuldade demonstrada dos alunos em agrupar e classificar comandos. Alguns alunos não conseguiram identificar comandos que estavam sendo escritos repetitivamente e perceber que esses poderiam ser agrupados, reorganizados, buscando um melhor desempenho do código (algoritmo). A capacidade de perceber essa possibilidade e resolver determinada situação de maneira mais inteligente, na área de computação é muito valorizada, pelo fato de que os códigos ficam menores, requerem menos processamento e são mais fáceis de alterações.

Na ferramenta gráfica, os comandos são representados pelos blocos que podem ser selecionados e arrastados, o que tornou visível essa dificuldade dos alunos em agrupar e classificar. Podemos ilustrar no exemplo da Figura 3, em que o aluno (A20) não conseguiu identificar que poderia ter agrupado os comandos “avance” e “vire à direita”, ou seja, não conseguiu estabelecer esse tipo de relação, utilizou dez “avance” para resolver o passo proposto pela atividade.

Figura 3 – Exemplo do passo 17 da atividade 2 (A<sub>20</sub>)



Fonte: BREZOLIN, 2016, p. 102.

Essa maneira de resolver não está totalmente errada, mas demanda muito mais processamento, pois são necessários dezesseis comandos, enquanto uma resolução mais econômica poderia ter quatro comandos, conforme se pode verificar no exemplo da Figura 4, em que o aluno (A<sub>23</sub>) conseguiu perceber que o “boneco” executava o mesmo procedimento de avançar e virar à direita, caso houvesse caminho disponível.

Figura 4 - Exemplo do passo 17 da atividade 2 (A<sub>23</sub>)



Fonte: BREZOLIN, 2016, p. 102.

Analisar ambas as maneiras de resolver o mesmo passo da atividade nos faz repensar sobre o tempo que cada aluno necessita para entender os conteúdos. Percebemos que enquanto um dos alunos precisou descrever as dezesseis instruções para fazer o caminho necessário, o outro conseguiu entender a possibilidade de agrupar e economizar instruções. Ocorrência essa que nos faz avaliar os conceitos do tempo didático da transposição didática apresentados por Pais (2008), referentes à programação de ensino. Essa programação está voltada a cumprir o programa proposto, enquanto o de aprendizagem deve respeitar o tempo necessário de cada sujeito para superar os bloqueios.

Assim, podemos refletir se a forma que ferramenta foi utilizada, de maneira individual e extraclasse priorizou o tempo de aprendizagem ou o tempo didático. Ainda, nesse sentido, conduzindo a questionamentos referentes ao planejamento das aulas da disciplina de algoritmos, indagamos se esse apresenta maior ênfase no tempo didático ou no tempo de aprendizagem.

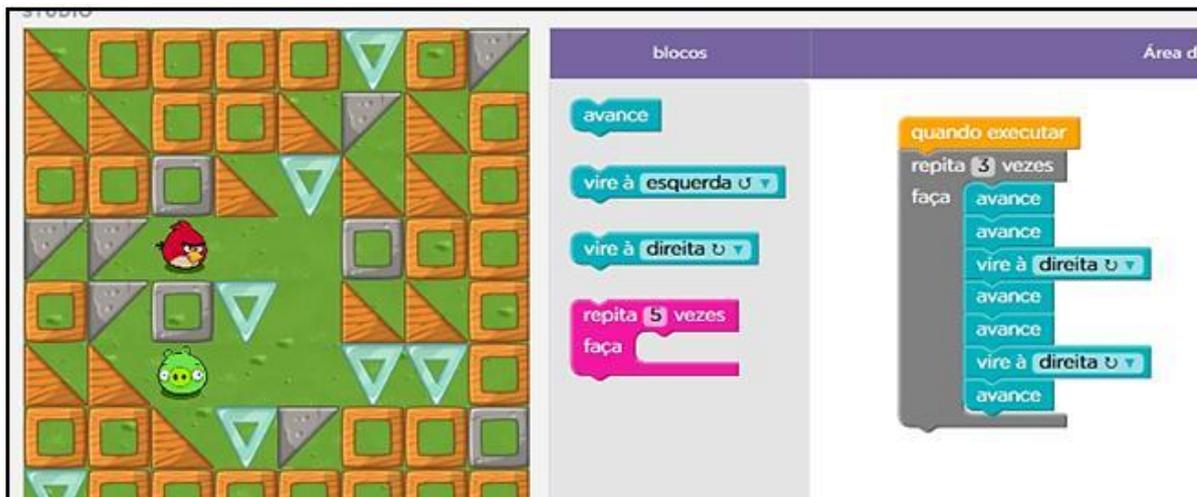
Observamos, com esse passo da atividade, que cada aluno apresentou uma solução refletindo o seu estágio de aprendizagem, ficando explícita a relação de que alguns precisaram realizar mais atividades para abstrair e identificar quais comandos poderiam estar agrupados para não ser necessário repetir. Essa dificuldade em agrupar, classificar e organizar os comandos está relacionada com a compreensão dos comandos de repetição, pois, se o aluno não consegue entender o que pode ser agrupado, também não conseguirá perceber o que pode ser envolvido por um comando de laço.

Entretanto um aspecto que se revelou como novidade com relação ao uso de laço de repetição diz respeito à maneira como os alunos entendem o seu funcionamento. Entendíamos que a dificuldade era em relação a definir os passos necessários para resolver o problema em si, ou seja, de estabelecer o plano de ação e determinar as instruções necessárias que estariam inseridas no comando de laço. Mas observamos que a dificuldade é quanto à compreensão da forma de execução do próprio comando de “laço”.

A dificuldade observada sobre a utilização de laços foi com o uso do comando “repita “n” vezes”, em que a variável “n” significa quantas vezes o laço irá se repetir. Observamos que muitos alunos parecem ignorar o número “3”, como no exemplo da (Figura 5) de um dos alunos ( $A_2$ ), e, assim, definem internamente os comandos necessários para resolver o passo como se não estivessem utilizando laços. Esse

fato mostra que os alunos não entenderam que os comandos envolvidos pelo “repita 3 vezes faça” executará três vezes. Ao invés disso, selecionaram os comandos “avance” e “vire à direita” desnecessários.

Figura 5 – Exemplo do passo 9 da atividade 2 envolvendo laços (A<sub>2</sub>)



Fonte: BREZOLIN, 2016, p. 104.

A proposta do passo 9 da atividade 2 tratava do movimento para que o personagem (vermelho) chegasse até o (verde), utilizando o comando “repita n vezes faça”. O que constatamos é que os alunos não apresentaram dificuldades para pensar no caminho necessário, mas não souberam realizar esse caminho de maneira menos trabalhosa.

Nesse sentido, é necessária a reflexão sobre esses comandos, pois analisando a produção das atividades dos alunos no ambiente gráfico, percebemos que esses não entenderam o conceito de laço, que, de fato, não têm generalizados em suas mentes, o que é um laço. Assim, é necessário repensar a apresentação desses comandos para que os seus significados sejam apropriados pelos alunos, relacionando com o significado e o sentido da palavra desses comandos.

Na concepção de Vygotsky, segundo Palangana “o desenvolvimento da linguagem coloca-se como paradigma para explicar a formação de todas as demais operações que envolvem o uso de signos” (2001, p. 102). O autor também acredita que é da relação entre a fala e a inteligência prática ou ainda, da combinação entre o instrumento e o signo que emergem as funções cognitivas superiores. Oliveira também descreve a análise de Vygotsky das relações entre pensamento e linguagem e afirma que:

A questão do significado ocupa um lugar central. O significado é um componente essencial da palavra, e é ao mesmo tempo, um ato do pensamento, pois o significado de uma palavra já é em si uma generalização. Isto é, no significado da palavra é que o pensamento e a fala se unem em pensamento verbal (1992, p. 48).

Para reforçar a importância do significado das palavras, Oliveira afirma que “do ponto de vista da psicologia, o significado de cada palavra é uma generalização de um conceito. E como as generalizações e os conceitos são inegavelmente atos do pensamento, podemos considerar o significado como um fenômeno do pensamento” (1992, p. 48).

Portanto, esse entendimento sobre o significado dos comandos que a professora esperava que os alunos já tivessem, podem ser comparados com as “noções paramatemáticas” apresentadas por Chevallard (2000, p. 59), segundo o autor são conhecimentos supostamente obtidos pelo aluno durante a sua formação anterior; noções essas que, geralmente, são apresentadas aos alunos por meio de demonstração sem uma reflexão sobre o seu significado.

## **CONSIDERAÇÕES FINAIS**

Um dos principais problemas percebidos durante a prática docente foi identificar quais eram as reais dificuldades dos alunos e essa problemática resultou em reflexões sobre as estratégias de ensino utilizadas em mais de dez anos de sala de aula como ministrante dessa disciplina. Essas inquietações instigaram a pesquisa que permitiu elucidar as dificuldades apresentadas pelos alunos e problemas no processo de ensino-aprendizagem da disciplina de Algoritmos. Nesse sentido, buscou-se na ferramenta gráfica blocky recursos para auxiliar no processo de ensino e aprendizagem dos alunos da disciplina de Algoritmos.

A partir desse contexto, a presente pesquisa, buscou investigar em que medida o uso de uma ferramenta gráfica de programação pode qualificar os processos de ensino e de aprendizagem na disciplina de Algoritmos. Após a análise dos dados, observou-se que a ferramenta além de permitir aos alunos aplicar e ampliar os conhecimentos abordados na disciplina, tornou-se um instrumento de diagnóstico. O material de análise produzido permitiu identificar dificuldades do processo da construção de algoritmos em relação aos conteúdos já abordados

anteriormente na disciplina que, até então, como professora-pesquisadora não tínhamos conseguido observar.

Ao analisarmos de forma individual o processo de desenvolvimento das atividades dos alunos com a ferramenta, procurando compreender o que poderia ter causado as dificuldades apresentadas, identificamos o conceito de “noções paramatemáticas”. Consideradas “noções ferramentas”, que se referem a conhecimentos que se supõe que o aluno já tenha obtido durante a sua formação anterior (CHEVALLARD, 2000).

Refletindo sobre a forma como são apresentados os conceitos da disciplina aos alunos com relação ao significado e o sentido das palavras, percebemos a necessidade de priorizar o significado dos signos, ou seja, das palavras, de cada um dos comandos utilizados, procurando evitar a simples apresentação de sua função ou uso excessivo de analogias nas suas aplicações.

Portanto, podemos afirmar que, mesmo diante do pequeno espaço de tempo do uso, e do período escolhido para aplicação, o processo nos possibilitou compreender a ferramenta como um instrumento mediador. Instrumento utilizado para estimular os alunos a elaborar os planos de ações e resolver os desafios, ao mesmo tempo em que dá pistas para incidir na Zona Desenvolvimento Proximal. Assim como também nos permitiu identificar elementos importantes que vêm ao encontro de uma das principais finalidades desta pesquisa: qualificar a prática da professora com relação à disciplina de Algoritmos.

## REFERÊNCIAS

ALICE. Versão 3.x. Carnegie Mellon University, 2008. Disponível em: <<http://www.alice.org>>. Acesso em: 13 abr. 2017.

ALMOULOUD, Saddo A. *Fundamentos da didática da matemática*. Curitiba: UFPR, 2007.

BREZOLIN, Carmen Vera Scorsatto. *Contribuições da Ferramenta Gráfica Blockly no Processo Ensino-Aprendizagem na Disciplina de Algoritmos*. Dissertação (Mestrado em Educação) Universidade de Passo Fundo, Passo Fundo, 2016.

CHEVALLARD, Yves. *La transposición didáctica: del saber sabio al saber enseñado*. Trad. Claudia Gilman. 3. ed. Buenos Aires: Aique, 2000.

GOOGLE DEVELOPERS (2015) *Blockly*: A visual programming editor. Disponível em: <<https://developers.google.com/blockly>>. Acesso em: 10 mai. 2017.

GRANDO, Neiva Ignês. Transposição didática e educação matemática. In: RAYS, Oswaldo Alonso (Org.). *Educação e ensino: constatações, inquietações e proposições*. Santa Maria: Pallotti, 2000.

OLIVEIRA, Marta Kohl de. *Vygotsky: aprendizado e desenvolvimento um processo sócio-histórico*. São Paulo: Scipione, 1992.

OLHARDIGITAL. Bill Gates e Zuckerberg ensinarão programação em projeto educacional. *Revista Online*. 2013. Disponível em: <<http://olhardigital.uol.com.br/pro/noticia/38236/38236>>. Acesso em: 28 abr. 2017.

PAIS, Luiz Carlos. Transposição Didática. In: MACHADO, Silvia Dias Alcântara (Org.). *Educação Matemática: uma (nova) introdução*. 3. ed. São Paulo: EDUC, 2008. p. 11- 48.

PALANGANA, Isilda Campaner. *Desenvolvimento e aprendizagem em Piaget e Vygotsky: a relevância social*. 3. ed. São Paulo: Sammus, 2001.

PASSOS, Claudio Cesar Manso; TEIXEIRA, Paulo Jorge Magalhães. Um pouco da teoria das situações didáticas (tsd) de Guy Brousseau. In: *ZETETIKÉ*. Campinas, v. 21, n. 39, Jan/Jun. 2013.

SPINA, Carli. Learn computer programming and Web. *Revista American Library Association*. 2013. Disponível em: <<http://crln.acrl.org/index.php/crlnews/article/view/9025/9838>>. Acesso em: 11 mai. 2017.