

PROGRAMAÇÃO: UMA EXPERIÊNCIA RECOGNITIVA OU INVENTIVA?

Rafael Schilling Fuck¹

Leonardo Rocha de Almeida²

Resumo

Atualmente, observamos um crescente interesse pelo ensino e aprendizagem de programação de computadores. Nesse contexto, uma das questões que emerge é o problema da cognição mediada por ambientes de programação. Desse modo, o objetivo do artigo é problematizar a atividade de programar, sob a perspectiva da Cognição Inventiva. Para atendê-lo, expomos, inicialmente, os conceitos-ferramenta relativos à Cognição Inventiva. Em seguida, apresentamos uma discussão acerca da atividade de programação promovida por algumas iniciativas disponíveis no ciberespaço e, por fim, ampliamos a discussão acerca da relação entre programação e cognição. Diante da atual necessidade de inovações nas práticas pedagógicas, problematizar a cognição mediada por tecnologias digitais é uma atitude essencial para o desenvolvimento de práticas inventivas e originais, que superem o modo hegemônico do conhecer representativo.

Palavras-chave: programação; tecnologia; invenção; cognição.

Abstract

Currently, we see a growing interest in the teaching and learning computer programming. In this context, one of the questions that arises is the problem of cognition mediated programming environments. The objective of this article is to discuss the activity program, from the perspective of Cognition Inventive. To answer it, we set out initially the tool concepts related to Cognition Inventive. Next is a discussion about the programming activity promoted by some initiatives available in cyberspace and eventually expanded the discussion about the relationship between programming and cognition. Given the current need for innovation in teaching practices, questioning the cognition mediated by digital technologies is an essential attitude for the development of inventive and original practices, which go beyond the hegemonic way of knowing representative.

Keyword: programming; technology; invention; cognition

INTRODUÇÃO

Atualmente, observamos um crescente interesse pelo ensino e aprendizagem de programação de computadores. Diversas investigações evidenciam tal interesse, dentre as quais se destacam as desenvolvidas por Martins (2012), França e Amaral (2013), Medeiros e Santos (2014), Barcelos (2014) e Fuck (2015). Ainda, pode-se encontrar diversos relatos de

¹ Doutor em Educação (Unisinos), Mestre em Educação em Ciências e Matemática (PUCRS), Graduado em Licenciatura em Matemática (Unisinos). Professor de Matemática da Rede Municipal de Educação de Canoas. E-mail: rafaelschillingf@gmail.com

² Mestre em Gestão Educacional (Unisinos), Especialista em Mídias na Educação (UFRGS); Graduado em Pedagogia (UniRitter), Graduando em Fonoaudiologia (UFCSPA). Professor da Rede Municipal de Ensino de Porto Alegre. Tutor do Centro Universitário Internacional UNINTER. E-mail: leonard.rocha@hotmail.com

experiências apresentados em forma de resumos em eventos; na *Internet*, é possível perceber ações que visam o incentivo ao ensino de programação para crianças, *blogs* de professores que relatam suas práticas envolvendo o ambiente de programação com seus alunos. Tal interesse não decorre somente em função da demanda de desenvolver competências e habilidades inerentes à Ciência da Computação por sujeitos para atender às expectativas do mercado de trabalho das Tecnologias de Informação (TI), mas, também, pelas possibilidades de desenvolvimento de experiências inovadoras de aprendizagem.

Nesse contexto, emerge o problema da cognição mediada por ambientes de programação, o qual pode ser traduzido pelas seguintes questões: O que o aprendiz de programação aprende sobre si e sobre o mundo ao programar? A programação possibilita aprendizagens inventivas ou representacionistas (recognitivas)? Quais os limites e possibilidades da atividade de programação? Desse modo, o objetivo do artigo é problematizar a atividade de programar, sob a perspectiva da Cognição Inventiva.

No intuito de conduzir o leitor a uma melhor compreensão da discussão desenvolvida neste artigo, primeiramente, expomos os conceitos-ferramenta relativos à Cognição Inventiva, necessários para a discussão dos próximos tópicos. Após essa etapa, apresentamos uma discussão acerca da atividade de programação promovida por algumas iniciativas disponíveis no ciberespaço, a partir da qual elaboramos algumas problematizações no que se refere à cognição. Por fim, ampliamos a discussão acerca da relação entre programação e cognição, seguida das considerações finais e referências.

COGNIÇÃO INVENTIVA

O cognitivismo computacional defende que conhecer significa processar informações oriundas do meio (*inputs*) produzindo respostas adequadas (*outputs*). Nessa perspectiva, a cognição é definida como computação simbólica mediada por regras lógicas, cujo objetivo é a representação adequada do mundo. Seus pressupostos dão sustentação a um movimento que pressupõe sujeito e objeto como pólos dados *a priori* do processo de conhecer.

No paradigma do cognitivismo, o conhecimento é concebido como prévio à existência humana. Quando organizado em um saber escolarizado, determina seus objetos como dados a serem descobertos em sala de aula, pressupondo, também, a existência de um sujeito humano, entendido como interioridade, o que significa que representa em si aquilo que lhe é exterior. Nesse sentido, a aprendizagem consiste em adequar-se do modo mais perfeito quanto possível

àqueles objetos. Assume-se como um processo de adaptação ao mundo através de sua representação (CAMMAROTA; CLARETO, 2012).

Configura-se, assim, uma atenção seletiva, orientada no sentido de uma ação eficaz, isto é, a aprendizagem do objeto. É exigida do sujeito aprendente a capacidade de memorização que dê conta de reconstituir o objeto exatamente como ele é. Ainda, exige-se uma linguagem desprovida de inconsistências e incoerências que expresse rigorosamente em texto aquilo que o objeto formal já assumia (ibidem).

Em oposição ao cognitivismo computacional, que concebe o conhecimento como representação do mundo através do processamento de informações por meio de leis e regras lógicas, a cognição inventiva entende o conhecimento como uma produção coengendrada do sujeito e do mundo. O conceito de cognição inventiva, proposto por Kastrup (2005), se sustenta nos estudos de Maturana e Varela, desde o conceito de autopoiese ao de enação. Segundo a pesquisadora,

a invenção não é um processo que possa ser atribuído a um sujeito. A invenção não deve ser entendida a partir do inventor. O sujeito, bem como o objeto, são efeitos, resultados do processo de invenção. (...) é a ação, o fazer, a prática cognitiva que configura o sujeito e o objeto, o si e o mundo. A transformação temporal da cognição não segue um caminho necessário, não leva a uma sequência de estruturas cognitivas e estágios que seguiriam uma ordem invariante, como nas teorias do desenvolvimento cognitivo, mas é antes uma deriva, criada a partir dos acoplamentos com as forças do mundo. (KASTRUP, 2005, p. 1275)

Ao problematizar a cognição, Kastrup opera uma ampliação desse conceito, propondo-o como uma cognição aberta à inventividade e à variação, cujos processos levam a resultados que não podem ser antecipados. Assim, as noções de sujeito e objeto como polos se desfazem, pois, de acordo com Varela (2003), sujeito e objeto são efeitos da ação cognitiva, isto é, são coengendrados e produzem-se mutuamente. Logo, a ação cognitiva não é pautada no sujeito ou numa determinação do ambiente sobre o homem.

Nesse momento, consideramos pertinente explicitar a diferença entre criatividade e invenção, termos cujos significados se confundem. O primeiro se reduz à busca de soluções originais para problemas, enquanto que o segundo propõe um movimento de problematização das formas cognitivas. Para Kastrup (2005), a invenção não é uma habilidade de resolver problemas, mas de invenção de problemas. Ainda, “a invenção é sempre invenção do novo, sendo dotada de uma imprevisibilidade que impede sua investigação e o tratamento no interior de um quadro de leis e princípios invariantes da cognição” (ibidem, p. 1274).

A palavra invenção possui sua etimologia no latim, *invenire*, que significa encontrar restos arqueológicos ou relíquias. Entretanto, seu significado não se restringe somente a um

insight, mas a um incansável processo de exploração, de busca, de experimentação. Portanto, para isso, requer-se tempo e o resultado é imprevisível. Sob uma perspectiva arqueológica, a invenção se caracteriza por sua imprevisibilidade e problematização, o que lhe atribui aspectos de dinamicidade, processualidade e interatividade. Nesse sentido, a cognição se assume como um processo de diferenciação de si mesma, a qual se encontra em incessante movimento e permanente processo de autoprodução. O meio, nesse caso, possui a função de incitar perturbações – *breakdowns*³ – na estrutura do organismo e a emergência de problemas (KASTRUP, 2007).

O termo *breakdown*, em Kastrup, assume o significado de uma rachadura na continuidade cognitiva, que potencializa a emergência do novo e desencadeia afecções no sujeito. Essa rachadura ocorre devido à continuidade da ação que, frente à problematização, “rastrea” pistas para o problema “em seu repertório constituído em seu processo sócio-histórico”. As teorias da aprendizagem, em geral, preocupam-se com o modo de como o sujeito se adapta ao meio ou como o ambiente determina mudanças no sujeito. Assim, essas teorias sustentam a ideia de previsibilidade e controle. A teoria behaviorista, por exemplo, postula a aprendizagem como um comportamento a ser aprendido pelo hábito e repetição, o qual é previsível e controlado. Ainda, a epistemologia genética de Piaget sustenta a gênese das estruturas cognitivas – que se desenvolvem de modo sequencial e invariante – como produto da adaptação do sujeito ao meio. As teorias que assumem essa perspectiva ambientalista respondem aos postulados do pensamento representativo. No entanto, de acordo com Kastrup (2007), essa perspectiva não possibilita conceber o mundo em contínua e acelerada mutação, como vem ocorrendo na contemporaneidade.

A partir do exposto, o significado de invenção assumido neste artigo é o de diferenciação de si. A invenção se configura quando a cognição sofre transformações em sua essência para outra, diferenciando-se de si mesma, em uma diferença que, na perspectiva bergsoniana, é de natureza e não de grau. Em uma abordagem de diferença de grau, concebe-se que a aprendizagem se dá pela aquisição de mais conhecimento, o que determina uma visão cognitivista, pois somente se acumula graus sem a produção de uma cognição original. Desse modo, percebemos a relevância dos *breakdowns*, rupturas no processo de cognição

³ O termo *breakdown* se refere a uma perturbação que ocorre no sujeito cognitivo. Essa perturbação é possibilitada devido à autopoiese e, como colocou Varela, promove o “nascimento do concreto” que se manifesta por meio da diferenciação de acoplamentos e modos de funcionamento prévios. O *breakdown* “não possui finalidade, não visa nada diferente dele mesmo, mas garante a potência viva do vivo ao conservar uma dimensão de problematização em toda ação” (SANCOVISH, 2010, p.31).

fundamentais para transformar a cognição de recognitiva em inventiva. São nesses *breakdowns* que emerge a invenção de problemas, pois a ruptura consiste na própria problematização da cognição. Portanto, não há sentido em pensar em uma adaptação ao meio, pois, como enfatizamos anteriormente, a cognição se diferencia de si mesma e é nesse processo de diferenciação que há invenção e não criatividade, pois esta se encontra condicionada aos limites decorrentes de uma adaptação ao meio.

INICIATIVAS DE PROGRAMAÇÃO NO CIBERESPAÇO

A partir de 2010, sob o lema “Todos os estudantes, em todas as escolas, devem ter a oportunidade de aprender a programar”, vem ocorrendo um movimento crescente de iniciativas nacionais e internacionais preocupadas em ensinar e disseminar conhecimentos sobre programação para escolas e pessoas leigas. Motivadas pela carência de profissionais da área de tecnologias e apoiadas por empresas desse ramo, essas são iniciativas que buscam convencer a sociedade e políticos acerca da importância do ensino e aprendizagem de programação. É o caso do *Code Academy*⁴, *Code.org*⁵, *Program.ar*⁶, *Code Club*⁷, *Code Club Brasil*⁸, *Programaê*⁹.

O *Code Club* é um projeto britânico fundado em 2012, cujo objetivo é ensinar programação para toda criança por meio de cursos básicos de programação, utilizando elementos atrativos como jogos de computador, por exemplo. Nos EUA, há o projeto *Code.org*, o qual tem ganhado grande destaque na sociedade norte-americana, levando cidades como Chicago e Los Angeles a anunciar, recentemente, a proposta de integrar o ensino de programação no currículo de suas escolas. No Brasil, esse tipo de projeto vem crescendo timidamente. Recentemente, em outubro de 2014, a Fundação Lemann, organização filantrópica, lançou o *site* Programaê, no qual agrega diversas iniciativas voltadas para o ensino de programação, entre elas o *Code Club Brasil*, inspirado no *Code Club* britânico.

Em entrevista concedida à Folha de São Paulo¹⁰, o fundador do *Code Club Brasil* declara que “a programação é importante não para formar profissionais da área, mas porque ensina a criança a pensar” e completa ao destacar que esse tipo de atividade modifica a relação da criança

⁴ <http://www.codecademy.com/pt-BR/learn>

⁵ <http://br.code.org/>

⁶ <http://program.ar/>

⁷ <https://www.codeclub.org.uk/>

⁸ <http://codeclubbrasil.org/>

⁹ <http://programae.org.br/>

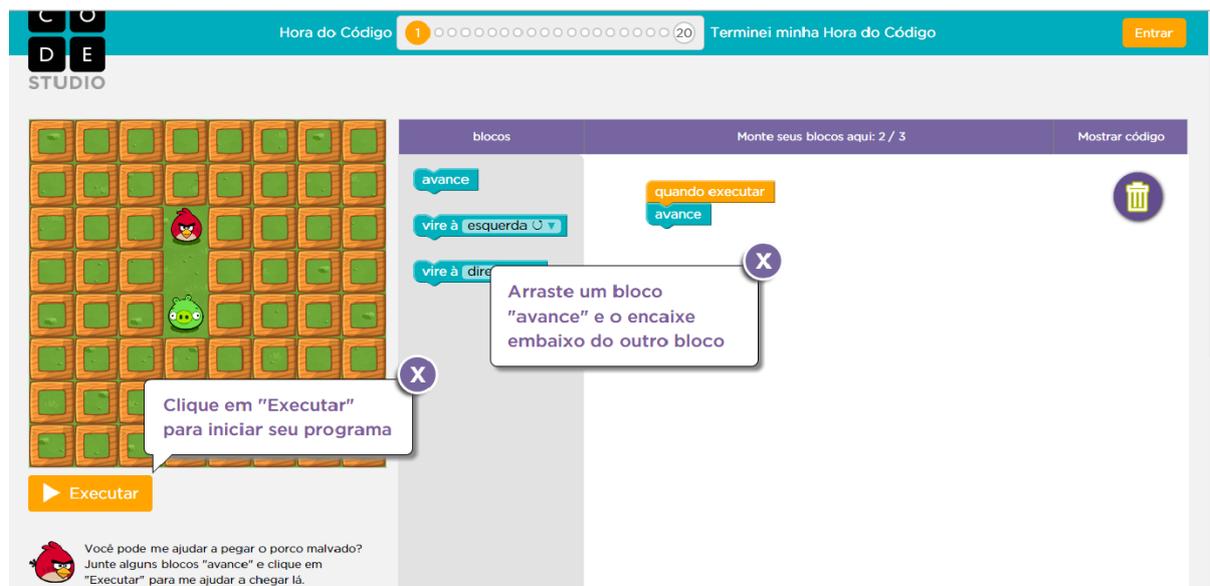
¹⁰ <http://www1.folha.uol.com.br/fsp/tec/192786-projetos-ensinam-programacao-e-em-escolas.shtml>

com a tecnologia, pois “ao invés de simplesmente usar o computador de forma passiva, como um depósito de informações, você passa a usá-lo para criar”. Aqui cabe problematizarmos o que seria esse “modo de pensar”. Que tipo de “modelo cognitivo” está lhe sendo ensinado? Na atividade de programar, um dos tipos de pensamento que o sujeito desenvolve é o algorítmico, pois, para executar uma ação, deve elaborar um algoritmo, um “passo-a-passo”. Nesse processo, percebemos que há pelo menos dois vieses: por um lado, ao desenvolver um algoritmo, o sujeito tem a oportunidade de analisar seu processo de desenvolvimento, de forma atenta e reflexiva, o que pode favorecer a emergência de *breakdowns* e, desse modo, a variação das linhas de fuga no pensamento, as quais revelam possibilidades de aprendizagens inventivas, resistindo e criando um pensamento sem imagem predefinida (DELEUZE, 2006); por outro, parece-nos que o pensamento algorítmico pode limitar as condições para uma aprendizagem inventiva, dando destaque à reconhecimento, pois subjacente a esse pensamento há uma lógica que orienta o sujeito programador a desenvolver um algoritmo de determinado modo, restringindo os espaços de aprendizagem inventiva.

Em uma breve exploração nos *sites* das iniciativas de programação citadas anteriormente, pode-se observar a presença de um grande repertório de tutoriais para introduzir o usuário à programação. Em sua maioria, são tutoriais autoexplicativos desenvolvidos em forma de jogo e destinados a crianças. Dentre eles, destaca-se o tutorial “*Angry Birds*”¹¹ (Figura 01), o qual consiste em orientar o usuário a arrastar um bloco de um determinado comando e encaixá-lo a outro para executar uma ação.

Figura 1 – Interface do jogo *Angry Birds*

¹¹ <http://studio.code.org/hoc/1>



Fonte: *site do jogo*

Ao colocarmos em exame essas iniciativas, observamos um caráter massivo de transmissão de informação para defender a importância de saber programação e de sua inclusão no currículo escolar. Em geral, essa importância é definida pela necessidade de formação de profissionais para atuar na área de Tecnologia da Informação (TI). De acordo com um estudo da consultoria IDC¹², somente no país, cerca de 117 mil vagas de TI ficarão abertas por falta de pessoas com qualificação necessária ao setor.

Essas observações nos levam a pensar que, por um lado, a atividade de programação nas escolas está sendo voltada à formação para o trabalho, a serviço do capital, com atuação no campo limitado de produção de *software*, atendendo a demanda comercial e o movimento do mercado global de consumo. Desse modo, entendemos que a formação concebida nessa perspectiva opera sob o regime da reconhecimento.

Por outro, essa atividade incita a ideia de formação de uma cultura de programação nas escolas, de um movimento da comunidade de *software* livre e da cibercultura a serviço do acesso universal e irrestrito, à cooperação e à solidariedade. Uma formação com esse objetivo opera no campo da criatividade e da invenção, em diálogo com a cultura local, atendendo a demandas sociais e pessoais, para inventar soluções abertas e em cooperação na resolução de problemas comuns. Nesse sentido, entendemos que uma formação voltada a essas características opera sob o regime da invenção.

¹² <http://br.idclatin.com/prodserv/consulting/>

Para disseminar os conceitos de programação e como programar as ações, as iniciativas o fazem por meio de cursos *online* autoassistidos e de diversos e diferentes tutoriais, nos quais o usuário/aluno deve seguir as instruções dadas. Aqui, podemos observar a lógica do modelo fordista presente nos cursos e a dos programas Computer-Aided Instruction (CAI), explícita pelos tutoriais. Assim, percebemos que a cognição do sujeito não é desafiada, no sentido de superação da recongnição. Observamos práticas que colocam em relevo a representação do conhecimento e a efetividade na resolução de problemas, já que o usuário deve apenas reproduzir, passo-a-passo, a programação e apresentar uma solução adequada para executar uma ação predeterminada. Nesse caso, incentiva-se apenas o *saber fazer* e não o *fazer problematizações* que configuraria uma transição para a invenção de problemas pertinentes e rupturas cognitivas.

Tem-se, então, uma política cognitivista que orienta as práticas dessas iniciativas, contribuindo para a manutenção dos regimes cognitivos e do *status quo*. As propostas de tais iniciativas têm apresentado o mesmo limite que o das práticas de Inclusão Digital (ID) concebidas atualmente¹³. Ambas têm se restringido apenas ao domínio da técnica, sem promover condições para o sujeito refletir sobre o uso, em si, dos recursos tecnológicos e da programação, e suas possibilidades para a invenção de si e do mundo.

A partir desse entendimento, um sujeito consciente criticamente – ou, ainda, um sujeito inventivo –, além de apreender os conceitos técnicos de programação, é capaz de fazer problematizações sobre suas ações efetivas e consequências ao programar, fazendo emergir questões pertinentes e significados – que, aliás, é a maior capacidade da cognição – e, desse modo, reinventar a si e o mundo em um contínuo processo de acoplamentos estruturais entre si e o meio. Esse sujeito deixa de ser alienado para tornar-se emancipado, isto é, no contexto da programação, recusa-se em tornar-se um sujeito programado para ser um sujeito programador de ideias originais.

PROGRAMAÇÃO E COGNIÇÃO

¹³ De acordo com diversos estudos, as práticas de inclusão digital, concebidas atualmente, apresentam a preocupação em proporcionar ao sujeito apenas capacitação técnica e acesso à Internet. Embora possibilite a ampliação das habilidades e capacidades dos sujeitos, "os resultados obtidos por programas de Inclusão Digital não se apresentam como suficientes para alterarem suas condições de emprego e renda" (FLEURY; SCHWARTZ, DAHMER, 2006, p. 3). Frente às limitações das práticas de inclusão digital, tem-se pensado em conceitos como Emancipação Digital, assim preconizado por Schwartz (2005), e Emancipação Digital Cidadã, por Schlemmer (2010).

A pertinência de desenvolver este tópico decorre do entendimento de que essa atividade revela um determinado modo de conhecer, de apreender a realidade, de conceber os problemas. Nesse sentido, essa atividade segue a orientação de um paradigma e, portanto, assume uma determinada política cognitiva. O ambiente de programação *Scratch*, por exemplo, foi desenvolvido sob um determinado paradigma, o da programação orientada a objetos (POO), e, desse modo, modela o pensamento do programador/desenvolvedor, conduzindo-o a “ajustar” sua cognição de acordo com suas estruturas e princípios que o fundamentam. Diante do exposto, entendemos que seja importante desenvolver uma discussão a respeito do conceito e da atividade de programar e dos distintos paradigmas que a sustentam.

Um programa de computador consiste em um conjunto de instruções que a máquina deve executar para cumprir uma determinada tarefa. Isto é, a atividade de construir um programa consiste em conceber uma solução dessa tarefa por meio de operações que a máquina possa executar e, em seguida, materializá-la por meio de uma linguagem de programação. O raciocínio inerente à definição da sequência detalhada de instruções para executar um programa é concebido como algoritmo (BINI; KONSCIANSKI, 2009). Assim, programar é construir algoritmos, codificados em determinada linguagem de programação, para a resolução de um problema, tornando possível, desse modo, a execução de um programa pelo computador.

Na atividade de programar, há pelo menos três elementos envolvidos: lógica, algoritmos e linguagem de programação. Jesus e Brito (2009), com base no estudo de diversos autores, definem a lógica como a “arte do bem pensar” (p.133) e colocam o raciocínio como o modo mais complexo do pensamento e a lógica como corretora do raciocínio, organizando a ordem e coerência no pensamento. A partir desses aspectos, descrevem que a lógica de programação, além de objetivar a ordem da razão, preocupa-se com “a simbolização formal do raciocínio para programação dos computadores” (ibidem). Já o conceito de algoritmo, semelhantemente ao que foi colocado anteriormente, é uma sequência lógica de passos a qual visa alcançar um objetivo bem definido. Finalmente, as linguagens de programação constituem ferramentas para implementação de um programa.

De acordo com essas ideias, a programação é uma atividade que remete a aspectos cognitivos, pois para o desenvolvedor/programador criar seu programa, ele o faz a partir de seu entendimento da ação de conhecer, de conceber um problema, de perceber-se a si mesmo e de como se relaciona com o mundo. Esses aspectos colaboram na constituição de sua concepção acerca da lógica, do algoritmo, da linguagem de programação e do produto de sua programação.

São esses aspectos que condicionam o processo de desenvolvimento e produto de sua atividade de programar.

Nesse sentido, a ação de programar implica mobilizar as estruturas cognitivas do sujeito, estruturas essas condicionadas por seu modo de conceber o conhecimento e o mundo. Entretanto, é importante considerar que há pelo menos duas perspectivas envolvidas na atividade de programação. A primeira, que foi amplamente discutida acima, se refere à cognição do sujeito programador, e a segunda, ao paradigma de desenvolvimento da linguagem de programação com a qual o programador interage.

Na ciência da computação, o termo paradigma é utilizado como um modo “de abstrair o pensamento do desenvolvedor em uma determinada estrutura computacional, capaz de definir a execução de um programa” (VALENÇA, 2012, p. 2). Um paradigma é a representação de um modelo a ser seguido, base filosófica, que agrupa métodos e técnicas que seguem um mesmo conjunto de princípios. É o paradigma que determina o ponto de vista da realidade e como se atua sobre ela.

Os paradigmas se distinguem em conceitos e abstrações empregados para representar os elementos de um programa e a maneira com que esses interagem, de modo a determinar o fluxo de execução de tal programa (ibidem). Nesse sentido, as linguagens de programação seguem paradigmas que definem suas características. Além disso, uma linguagem de programação pode combinar dois ou mais paradigmas para potencializar as análises e soluções. Desse modo, cabe ao programador escolher o paradigma mais adequado para analisar e resolver cada problema.

No contexto da programação, há diversos paradigmas, os quais são classificados quanto ao seu conceito de base, dentre os quais se destacam: imperativo, estruturado e orientado a objetos. Cada qual condiciona um modo particular de abordar os problemas e de formular respectivas soluções. O primeiro, paradigma imperativo, segue o conceito de um estado e de ações que manipulam esse estado, recorrendo a procedimentos que servem de mecanismos de estruturação; o segundo preconiza que todos os programas podem ser reduzidos a três estruturas: sequência, decisão e iteração. O paradigma estruturado orienta o programador para a criação de estruturas simples em seu programa, utilizando sub-rotinas e funções. Por fim, o mais influente de todos, o paradigma de Programação Orientada a Objetos (POO) determina a composição e interação de diversas unidades do programa denominadas objetos. O funcionamento de um programa orientado a objetos ocorre por meio do relacionamento e troca de mensagens entre esses objetos (JUNGTHON; GOULART, s/d).

A partir do exposto, consideramos que a atividade de programar é atravessada por questões acerca da cognição. O que essa discussão remete é que a programação não está condicionada apenas à concepção de cognição do programador, como também ao paradigma sob o qual a linguagem de programação foi desenvolvida. Ao analisar essas questões, observa-se que a atividade de programar, geralmente, assume características da reconição, pois, para traduzir um problema ou um fenômeno, o programador o faz por meio de regras lógicas, do algoritmo e da linguagem de programação, representando-o de forma mais adequada possível, o que resultará em um programa. Assim, o paradigma de cognição do programador e o paradigma de desenvolvimento da linguagem de programação atuam como “filtro” do mundo, o que implica no entendimento de que o processo de conhecer, pressupondo sujeito e objeto como polos *a priori*, consiste apenas na representação do mundo.

Decorrente dessa análise, retomamos a seguinte questão: como a atividade de programar é concebida na perspectiva da Cognição Inventiva? Diferentemente da reconição, que concebe sujeito e mundo como elementos predefinidos ao ato de conhecer, a Cognição Inventiva entende que tais elementos são coengendrados. A atividade inventiva de programar se constitui por meio de acoplamentos entre o programador e a linguagem de programação que emprega para elaborar seus programas. Nessa ótica, entendemos que não seja possível concebermos o sujeito programador e o programa como polos, em que o primeiro interage sobre o segundo, produzindo, desse modo, conhecimento. O conhecimento é produto provisório da moldagem recíproca entre sujeito e tecnologias, ou seja, as tecnologias moldam o modo como o sujeito pensa do mesmo modo que são moldadas por este mesmo sujeito, pois “os seres humanos são constituídos por técnicas que estendem e modificam seu raciocínio e, ao mesmo tempo, esses seres humanos estão constantemente transformando essas técnicas” (BORBA; PENTEADO, 2005, p. 46).

Por fim, levantamos outra questão: como o programador pode superar a reconição, de tal modo a produzir conhecimentos inventivos? Com base no entendimento de que há uma tensão entre o criar e o repetir (KASTRUP, 2007; FUCK, 2016), o que significa que a invenção surge de base cognitivista, o programador necessita problematizar, continuamente, sua atividade de programar, concentrando-se em aspectos que escapam do “filtro” do paradigma cognitivo. São esses aspectos que podem revelar a originalidade de sua produção, que podem se constituir em algo inédito, diferenciando-se do que já está dado e exaustivamente representado.

CONSIDERAÇÕES FINAIS

Neste artigo, buscamos problematizar a cognição mediada por alguns ambientes de programação disponibilizados no ciberespaço e discutir a relação entre programação e cognição. A atividade de programação pode se constituir em uma oportunidade de o sujeito deixar de ser consumidor para ser criador de tecnologias. Programar exige pensar, compreender o que se faz e o que se pensa, interpretar, enfim, problematizar, fazendo emergir questões pertinentes nesse processo. Entretanto, o uso de ambientes de programação, por si só, não significa que o sujeito vá romper com a reconição. Os diferentes modos de conceber a aprendizagem conduzem a diferentes potencialidades na utilização das tecnologias. Desse modo, depende em qual paradigma a atividade de programar está inserida.

A pertinência de abordar essa temática está no entendimento de que a atividade de programar pode condicionar o desenvolvimento cognitivo do sujeito programador em direção à reconição ou à invenção. Diante da atual necessidade de inovações nas práticas pedagógicas, problematizar a cognição mediada por tecnologias digitais é uma atitude essencial para o desenvolvimento de práticas inventivas e originais, que superem o modo hegemônico de conhecer representativo.

REFERÊNCIAS

BARCELOS, T. S. **Relações entre o Pensamento Computacional e a Matemática em atividades didáticas de construção de jogos digitais**. 2014. 276 f. Tese (Doutorado) – Programa de Pós-Graduação em Ensino de Ciências e Matemática, Universidade Cruzeiro do Sul, São Paulo, 2014.

BINI, E. M.; KOSCIANSKI, A. O ensino de programação de computadores em um ambiente criativo e motivador. In: ENCONTRO NACIONAL DE PESQUISAS EM EDUCAÇÃO EM CIÊNCIAS – ENPeC, 7, 2009, Florianópolis. Anais... Florianópolis, 2009.

BORBA, M. C.; PENTEADO, M. G. **Informática e Educação Matemática**. Belo Horizonte: Autêntica, 2005.

CAMMAROTA, G.; CLARETO, S. M. A cognição em questão: invenção, aprendizagem e Educação Matemática. **Práxis Educativa**, Ponta Grossa, v. 7, n. 2, p. 585-602, jul./dez.2012.

DELEUZE, G. **Diferença e repetição**. Rio de Janeiro: Graal, 2006.

FRANÇA, R. S.; AMARAL, H. J. C. Proposta Metodológica de Ensino e Avaliação para o Desenvolvimento do Pensamento Computacional com o Uso do Scratch. In: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 2, 2013.

FLEURY, A. L.; SCHWARTZ, G; DAHMER, A. Z. **Da Inclusão à Emancipação Digital: novos modelos de produção e distribuição de conteúdo digital**. XXVI ENEGEP, 2006.

FUCK, R. S. **Da reconhecimento e da cognição inventiva**: uma cartografia das experiências de programação por estudantes de escolas públicas do Ensino Fundamental. 2016. 352 f. Tese (Doutorado em Educação) – Universidade do Vale do Rio dos Sinos (UNISINOS), São Leopoldo, 2016.

_____. Cartografia dos processos cognitivos emergentes da atividade de programação: um estudo com alunos participantes de uma oficina de *Scratch*. **Redin**, Taquara, v.4, n.1, 2015.

JESUS, A.; BRITO, G. S. Concepção de Ensino-Aprendizagem de Algoritmos e Programação de Computadores: a prática docente. In: ENCONTRO NACIONAL DE INFORMÁTICA E EDUCAÇÃO – ENINED, 1, 2009, Curitiba. **Anais...** Curitiba, 2009.

JUNGTHON, G.; GOULART, C. M. Paradimas de Programação. s/d. Disponível em: <https://fit.faccat.br/~guto/artigos/Artigo_Paradigmas_de_Programacao.pdf>. Acesso em: 03 nov. 2015

KASTRUP, V. Políticas cognitivas na formação do professor e o problema do devir-mestre. **Educação & Sociedade**, Campinas, v. 26, n. 93, p. 1273-1288, set./dez. 2005.

_____. **A invenção de si e do mundo**: uma introdução do tempo e do coletivo no estudo da cognição. Belo Horizonte: Autêntica, 2007.

MARTINS, A. R. Q. **Usando o Scratch para potencializar o pensamento criativo em crianças do ensino fundamental**. 2012. 113 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Educação, Universidade de Passo Fundo (UPF), Passo Fundo, 2012.

MEDEIROS, J. S. S.; SANTOS, C. P. F. Scratch no Ensino de Ciências: potencializando o raciocínio lógico e a aprendizagem de estudantes no ensino fundamental. In: CONGRESSO INTERNACIONAL DE EDUCAÇÃO E INCLUSÃO - CINTEDI, 2014, Campina Grande. **Anais eletrônicos...** Campina Grande, 2014. Disponível em: <<http://editorarealize.com.br/revistas/cintedi/resumo.php?idtrabalho=807>>. Acesso em: 14 dez. 2014.

SANCOVISH, B. **Sobre as práticas de estudo de estudantes de psicologia: uma cartografia da cognição contemporânea**. 2010. 265 f. Tese (Doutorado) – Programa de Pós-Graduação em Psicologia, Universidade Federal do Rio de Janeiro (UFRJ), Rio de Janeiro, 2010.

SCHLEMMER, E. Formação de professores na modalidade *on-line*: experiências e reflexões sobre a criação de espaços de convivência digitais virtuais. **Em Aberto**, Brasília, v. 23, n. 84, p. 99-122, nov. 2010.

SCHWARTZ, G. Educação Digital para Emancipação Social. **Jornal da USP**, Universidade de São Paulo, 2005.

VALENÇA, G. Z. **Contribuição para a materialização do paradigma orientado a notificações (PON) via *framework* e *wizard***. 2012. 205 f. Dissertação (Mestrado) – Univeridade Tecnológica Federal do Paraná (UTFPR), Curitiba, 2012.

VARELA, F. O reencantamento do concreto. **Cadernos de Subjetividade**. São Paulo: Hucitec Educ, 2003. p.71-86.