

O LETRAMENTO COMPUTACIONAL E A AUTO-HETEROECOFORMAÇÃO TECNOLÓGICA COMPLEXA DOCENTE: UMA VIA PARA A COMPLEXIDADE?

Coding Literacy and complex self-hetero-eco technological teacher's formation: a pathway to complexity?

Luciana Espindola Correa LOURO (LAEL/PUC-SP)¹

RESUMO: Este artigo investiga a possibilidade de religar, por meio da aprendizagem da linguagem de programação, os conceitos de letramento (VEE, 2013, 2017; SOARES, 2004; DISESSA, 2000), auto-heteroecoformação complexa (FREIRE; LEFFA, 2016[2013]) e pensamento complexo (MORIN, 1986, 1990, 2002). Logo, lança-se mão de uma metodologia de pesquisa bibliográfica acerca do tema (CUNHA, 2001). Pergunta-se, portanto, se a escrita de códigos de computadores, apreendida sob essa tessitura conceitual (VEE, 2013, 2017; SOARES, 2004; DISESSA, 2000; FREIRE; LEFFA, 2016 [2013]; MORIN, 1990, 1986, 2002), poderia promover a tomada de consciência da complexidade (GALVANI e PINEAU, 2012). O caráter rizomático dos códigos computacionais, inseridos no contexto da computação (MORIN, 1986, 1990), revelam o aspecto complexo desses. Esse aspecto tende a evidenciar-se quando permeado pela abordagem didática proposta por Freire e Leffa (2016[2013]) bem como apresentado a partir da perspectiva de letramento (VEE, 2013; 2017; DISESSA, 2000), à luz da interpretação brasileira do termo (SOARES, 2004).

PALAVRAS CHAVE: letramento, letramento computacional, complexidade e auto-heteroecoformação tecnológica complexa.

ABSTRACT: This article discusses the possibility of link the programming language learning to the concepts of literacy (VEE, 2013, 2017; SOARES, 2004; DISESSA, 2000) and self-hetero-eco technological teacher's formation (FREIRE; LEFFA, 2016[2013]), under the lens of complexity (MORIN, 1990, 1986 e 2002). Then, a methodology of bibliographic research on the subject is launched (CUNHA, 2001). Therefore, it is asked if the writing of computer codes, learned under this conceptual framework (VEE, 2013, 2017; SOARES, 2004; DISESA, 2000; FREIRE; LEFFA, 2016[2013]; MORIN, 1990), could promote the awareness of complexity (GALVANI and PINEAU, 2012). The rhizomatic character of the computational codes, in the context of the computation (MORIN, 1986), reveal the complex behavior of these. This behavior of codes can be emphasized when it is linked to the didactic approach of Freire e Leffa (2016[2013]) and show through the lens of computational literacy (VEE, 2013, 2017; DISSESSA, 2000), under the Brazilian concept of literacy (SOARES, 2004).

KEYWORDS: literacy, computational literacy, complexity, complex self-hetero-eco technological teacher's formation.

¹ Doutoranda no PEPG – LAEL, da Pontifícia Universidade Católica de São Paulo. Email: leclouro@gmail.com

INTRODUÇÃO

A escola brasileira atual encontra-se distante da realidade contemporânea que está imersa em uma tecnologia sistêmica de redes de troca de conhecimento (FREIRE, 2009). Este paradoxo é explicado por Freire e Leffa (2016[2013]) como consequência do enraizamento da Educação do Brasil em princípios cartesianos de fragmentação do saber e hierarquia dos atores educacionais. Os pesquisadores propõem, como alternativa a tal cenário, uma mudança paradigmática dentro dos muros escolares brasileiros por meio da auto-heteroecoformação tecnológica docente (FREIRE; LEFFA, 2016[2013]).

A problemática da formação tecnológica docente já preocupava o governo brasileiro desde a década de 80. Proposto em 1983, o projeto Educação e Computador (EDUCOM) objetivava formar professores do segundo grau (OLIVEIRA, 2007). Oliveira (2007) explica que, dentro da mesma proposta de formar usuários em tecnologia surgiram várias outras iniciativas, como o Programa Nacional de Informática na educação de 1998, o Projeto Ensino on-line (EOL) e o Projeto um Computador por Aluno (PROUCA), lançado em 2010 (SAMPAIO; ELIA, 2012). Todavia, o PROUCA deu origem a algumas tentativas de implementar o ensino de linguagem de programação para professores de disciplinas da área de exatas, como o “projeto Uca na Cuca” (SAMPAIO; ELIA, 2012 p. 205). Por outro lado, uma breve análise das propostas do PROUCA voltadas às humanidades reforça a ideia predominante neste campo de conhecimento de focar na formação de usuários, procurando adaptar o currículo às novas tecnologias da informação e comunicação (SAMPAIO; ELIA, 2012).

Os anos oitenta também foram o berço para discussões acerca de diversos tipos de letramentos relacionados à revolução tecnológica iniciada pelo advento dos computadores e da internet (VEE, 2013, 2017). Para Vee (2013, 2017) esses eram analisados pela linguística aplicada à educação sob a ótica do usuário. Para propor uma consonância em meio a essa miríade de aproximações ancoradas no uso compartimentado de interfaces numéricas, Vee (2013, 2017) relaciona o conceito de letramento, desenvolvido ao longo da história da escrita, ao de linguagem de programação. Alinhado às discussões da época acerca do tema associado à aquisição da escrita, o termo letramento foi diferenciado da alfabetização e relacionado ao pertencimento social de um indivíduo a uma cultura escrita (SOARES, 2004). Tal especificidade (SOARES, 2004) será aprofundada mais adiante com o intuito de contrastá-la ao conceito de letramento computacional (VEE, 2013, 2017). Levando em

conta as considerações feitas até este momento (VEE, 2013, 2017; SOARES, 2004, DISSESSA, 2000; FREIRE, 2009; FREIRE; LEFFA, 2016[2013]; MORIN, 1990) chegou-se à seguinte pergunta:

É possível tecer conceitualmente a aprendizagem da escrita de códigos computacionais, sob a ótica do letramento (VEE, 2013, 2017; DISSESSA, 2000; SOARES, 2004) e da auto-heteroecoformação tecnológica docente (FREIRE; LEFFA, 2016[2013]), enquanto uma via de reconhecimento da complexidade (GALVANI; PINEAU, 2012)?

De acordo com Galvani e Pineau (2012, p. 217) “não se pode ensinar a complexidade, podemos favorecer sua conscientização. Podemos ajudar uns aos outros a reconhecê-la em uma coaprendizagem em diálogo com a vida vivida”. Objetiva-se, portanto, discutir a possibilidade de tecer a aprendizagem da linguagem de programação, sob a ótica do letramento (VEE, 2013, 2017; SOARES, 2004) e da auto-heteroecoformação tecnológica docente complexa (FREIRE; LEFFA, 2016[2013]), enquanto uma via de reconhecimento da complexidade (GALVANI; PINEAU, 2012).

Em consequência da natureza teórica deste estudo, adotar-se-á abordagem metodológica de pesquisa bibliográfica. Segundo Cunha (2001) a pesquisa bibliográfica consiste no uso de informações já elaboradas. Ainda dentro dos temas e escopo anteriormente propostos, optou-se por selecionar como “fonte de informação científica e tecnológica (ICT)” (CUNHA 2001, p. 8) periódicos como documentos primários e livros como secundários. Cunha (2001) explica que, de acordo com Grogan (1970), os dados ou documentos primários trazem ideias novas ou descobertas que são organizadas pelos documentos secundários.

Como detalhado anteriormente, o objetivo desse trabalho é a discussão de uma possível tessitura da aprendizagem da escrita de códigos de computadores, sob o prisma do letramento (VEE, 2013, 2017; SOARES, 2004, FREIRE, 1980) e da auto-heteroecoformação tecnológica docente complexa (FREIRE; LEFFA, 2016[2013]), enquanto um caminho de reconhecimento da complexidade. Logo, a escolha da pesquisa bibliográfica justifica-se pelo potencial da ICT de “servir de manancial de ideias ou para o desenvolvimento de uma ideia” (CUNHA, 2001 p. 8).

Uma vez estabelecidos o objetivo, a abordagem metodológica e a pergunta que nortearão o presente estudo, proceder-se-á a uma síntese dos conceitos teóricos anteriormente mencionados (SOARES, 2004; VEE, 2013, 2017; DISSESSA, 2000;

MORIN, 1986, 1990, 2002). Por fim, confrontar-se-á tais teorias a fim de considerar a viabilidade conceitual de conjecturar uma abordagem pedagógica, pautada na auto-heteroecoformação complexa (FREIRE, 2009; FREIRE;LEFFA, 2016[2013]), que proponha um olhar complexo sobre a aprendizagem da escrita de códigos computacionais.

O QUE A ESCRITA E OS CÓDIGOS COMPUTACIONAIS TÊM EM COMUM?

Em 2013 a campanha Code-org (GUYNN, 2013) popularizou o debate acerca do ensino universal da linguagem de programação. Apoiada pelo fundador do Face Book, Mark Zuckerberg, e o fundador da Microsoft, Bill Gates, a campanha foi criada para incentivar a aprendizagem da linguagem de programação por crianças, jovens e adultos. De acordo com a Code-org, professores, estudantes e profissionais deveriam unir-se em prol da aprendizagem dos códigos computacionais, considerada nova alfabetização do século XXI (GUYNN, 2013). Além da versão brasileira desse movimento, iniciada em 2014 sob o nome de Hora do Código (CAPELO, 2014), o Brasil também contou com a iniciativa do Programaê, realizada pela Fundação Telefônica Vivo em parceria com a Fundação Lemann (FUNDAÇÃO TELEFÔNICA VIVO; FUNDAÇÃO LEMANN, 2018). Como parte de sua campanha para promover o ensino de linguagem de programação na Educação Básica, o Programaê lançou, em 2018, um guia para auxiliar professores a implementar os códigos computacionais em sala de aula (FUNDAÇÃO TELEFÔNICA VIVO; FUNDAÇÃO LEMANN, 2018).

Todavia o debate acerca da pertinência do ensino universal dos códigos de computadores causa polêmica. De acordo com o relato de Perlis (1964), essa perspectiva dos códigos enquanto um letramento já era contestada por Elias, professor de Ciência de Computação do *Massachusetts Institute of Technology (MIT)*. Elias (PERLIS, 1964) pregava que os códigos tornar-se-iam tão análogos à língua humana que não seria mais preciso aprender como escrevê-los. Outra fala comum no meio científico, contrária à massificação do ensino de códigos de computadores, relaciona a habilidade de programar a contextos profissionais específicos, conforme sintetiza o post “Por favor, Não Aprenda a Programar!”² X de Atwood (2012, tradução nossa).

Porém, a linguista Vee (2013, 2017) ressalta que os códigos desenvolveram-se além dos cenários profissionais relacionados à Ciência da Computação. Neste sentido, a

² “Please Don’t Learn to Code” (ATWOOD, 2012)

autora (2013, 2017) cita a inserção da escrita de códigos no conceito de letramento, advogada por Prensky (2008), estudioso que lançou o conceito de nativo digital em 2001 (PRENSKY, 2001). Outro estudioso que defende a universalização do ensino dos códigos computacionais é Rushkoff (2010), teórico e entusiasta que acredita nos códigos abertos enquanto um caminho para a solução de questões sociais (RUSHKOFF, 2010).

Vee (2017, 263-264) acredita, assim, que o letramento deve ser compreendido a partir de seus usos sociais e pode ser definido como: “um conjunto de práticas envolvendo tecnologias da comunicação infra estruturais, amplamente difundidas cujo uso é socialmente valorizado”³

A definição da pesquisadora (VEE, 2017) fundamenta-se, principalmente, no que diSessa (2000, p. 6, tradução nossa), cientista computacional e escritor, chama de “material intelligence” definida como “uma Inteligência construída cooperativamente com materiais externos”. Por “material externo” diSessa (2000, p. 7) considera qualquer forma física estável, reproduzível, manipulável e transportável onde é possível instalar alguns aspectos do pensamento humano. Vee (2017, p. 97), complementa esse conceito dizendo que inteligência é “a habilidade de interpretar” o componente material da escrita.

Nesse sentido a inteligência material seria um letramento que acontece localmente em uma ou mais comunidades de prática, mas não organiza a estrutura social, cultural, econômica e pessoal de uma sociedade. Ainda segundo diSessa (2000) todo letramento começa como uma inteligência material.

Vee (2013; 2017) defende um estudo comparativo da história da transformação da escrita em um letramento com o trajeto sociocultural percorrido pela escrita de códigos de computadores. Para a autora (VEE, 2013; 2017) esse estudo permitiria enxergar o fenômeno da linguagem de programação enquanto um letramento emergente. Seria, portanto, possível verificar se a prática social da escrita de códigos estaria transformando as formas como as pessoas experienciam, ensinam, aprendem e desenvolvem suas habilidades individuais relativas às tecnologias de comunicação (VEE, 2013; 2017). Para isso, Vee (2017) traçou um paralelo entre a escrita e a linguagem de programação que pode ser resumido em quatro pontos conforme resume o quadro abaixo.

³ “a widely held, socially useful and valued set of practices with infrastructural communication technologies.” (VEE, 2017, 263-264).

APROXIMAÇÕES ENTRE A ESCRITA E A LINGUAGEM DE PROGRAMAÇÃO	
Escrita	Linguagem de Programação
Substituiu muitas vezes a comunicação direta feita pela oralidade (face a face). Mas a comunicação oral continuou existindo ao lado da escrita, criando uma relação de interdependência com essa.	Substituiu a escrita no papel e em outros dispositivos tradicionais. Os códigos computacionais estão sobre e sob a infraestrutura social escrita e os letramentos oriundos da cultura digital. Por isso eles convivem ecologicamente com a escrita e com os letramentos, sem suplantá-los.
Documentos escritos celebram as transações econômicas, a vida pública e privada.	Todos documentos foram transformados em dados digitais e dependem da computação para serem organizados e disponibilizados.
A circulação da computação depende da escrita.	A circulação da escrita depende da computação.
A escrita facilitou e transformou as relações humanas, diminuindo distâncias e criando novas formas de interação entre os indivíduos bem como entre os cidadãos e seus governos.	A computação também diminuiu distâncias e transformou as relações pessoais, cívicas e econômicas.

Tabela 1- Pontos de toque entre a linguagem de programação e a escrita. Quadro feito a partir dos argumentos de Vee (2017, p.)

Ancorada nessa ideia, Vee (2017) defende que a escrita de códigos deveria ser moldada e ensinada por diversos setores da sociedade, transbordando as fronteiras de ciência da computação. De onde veio a necessidade de analisar a relação histórica entre a prática de programar e a ciência da computação. A computação, assim como escrita, teria produzido novos modos de informação e comunicação (VEE, 2017). Esses não substituíram a escrita, mas fizeram com o que o seu uso fosse estendido, mudando seu público, a forma de estruturá-la e seus processos de distribuição.

A linguista propõe uma análise da linguagem de programação sob a ótica da trajetória histórica da escrita enquanto um “letramento plataforma” (VEE, 2017, p. 29, tradução nossa)⁴. Para compreender essa tese de Vee (2017) é necessário, portanto, voltar no tempo em que a escrita ainda estava dando seus primeiros passos diante de uma língua oral já consolidada.

A HISTÓRIA DA ESCRITA E DA LINGUAGEM DE PROGRAMAÇÃO: DE INTELIGÊNCIAS MATERIAIS A LETRAMENTOS?

Higounet (2003) explica que a escrita originou-se da fala. Mas essa técnica foi

⁴ “platform literacy” (VEE, 2017, p. 29).

muito além do registro de textos orais. A escrita criou uma grande autonomia em relação à língua oral, substituindo-a em muitas práticas sociais. Atento à ligação entre texto falado e escrito, o autor (HIGOUNET, 2003, p. 9) fez uma síntese da história da escrita sob a perspectiva da “expressão gráfica da linguagem”, explicando como o homem criou símbolos que culminaram em alfabetos usados até hoje. Nesse sentido Besserat (1992) comenta que os signos que antecederam a escrita foram criados, em muitos casos, para representar quantidades em acordos comerciais. Os dois autores enfatizam (HIGOUNET, 2003; BESSERAT, 1992) a importância dos suportes materiais para evolução da escrita.

Historiadores da Ciência da Computação como Augarten (1984) consideram que o desenvolvimento e idealização das máquinas de computar possuem uma relação direta com os códigos computacionais para elas desenvolvidos. O aprimoramento de uma máquina permitiria o desenvolvimento de uma linguagem mais elaborada (AUGARTEN, 1984). Desse modo, a história dos hardwares⁵ e softwares estão entrelaçadas e possuem, ambos, suas raízes mergulhadas na matemática. Essa relação profunda com a matemática surgiu por meio dos anseios de inventores como Babbage que idealizou uma máquina analítica para evitar erros de cálculo (AUGARTEN, 1984). Dessa invenção nasceu o primeiro programa computacional escrito pela Condessa Lovelace em 1843 (STEIN, 1985).

Ve (2017) também sublinha que, desde o início da computação, os cientistas procuram meios para traduzir ou compilar as linguagens de máquina, aproximando-a da linguagem humana. Percebe-se aqui duas similaridades históricas importantes. A primeira seria o parentesco com a matemática dessas duas práticas sociais. Alguns dos primeiros rudimentos de escrita ocidental seriam abstrações de quantidades (BESSERAT, 1992). E a linguagem de máquina ainda é uma série números binários que representam a intensidade da voltagem da corrente elétrica pelos transistores de uma máquina (AUGARTEN, 1984). A segunda semelhança seria a grande influência da materialidade sobre a transformação e evolução desses tipos de escritas.

Para Goody (1963)⁶ essa materialidade da escrita tornou-a, de início, uma atividade que requeria um alto grau de especialização. Da antiguidade até o surgimento

⁵ Entende-se aqui hardware, como a parte física de um computador e software como os programas contidos nele (AUGARTEN, 1984).

⁶ Ao falar de materialidade refiro-me ao conceito de “material externo” usado por diSessa (2000, p.7), já descrito na introdução desse artigo.

da tipografia, a escrita era uma habilidade altamente especializada, reservada a poucos. Por exemplo, na idade média um escriba deveria saber preparar tintas, pergaminhos, fazer a manutenção das ferramentas e instrumentos de escrita, entre outros saberes específicos. Esses especialistas serviam a comunidade, tornando a escrita uma tecnologia comunicacional reservada a poucos e usada em contextos sociais precisos como a redação de documentos oficiais ou textos religiosos (HIGOUNET, 2003; GOODY, 1963).

A necessidade de escribas para produzir os textos antigos, pode ser outro paralelo histórico entre a escrita e linguagem de programação desenvolvida dentro dos computadores (VEE, 2017). Durante a Segunda Guerra Mundial, entre 1939 e 1945, o uso de códigos computacionais era predominantemente militar. A computação era uma prática que exigia habilidades altamente especializadas tanto para operar quanto para montar as máquinas. Normalmente quem conseguia operar essas máquinas possuía um grande conhecimento em matemática, física e mecânica (AUGARTEN, 1984). A partir dessa ótica, o letramento, estaria, segundo Vee (2017), sendo regido pela materialidade do suporte ou dispositivo de escrita.

Já na batalha entre protestantes e católicos, iniciada no século XVI, a leitura autônoma da bíblia passou a ser considerada uma virtude entre os seguidores de Lutero. Surgiu, assim, a semente ideológica do letramento ligado à uma qualidade moral do indivíduo. A ascensão do nacionalismo e o crescimento de indústrias no século XIX impulsionaram essa ideia, transformando o ato de letrar-se em um dever cívico. Quem não fosse letrado estaria, portanto, à margem da sociedade. No século XX as campanhas de alfabetização relacionavam explicitamente os indicadores de progresso de um país ao seu índice de analfabetismo. Escolas proliferaram-se ao redor do globo em uma verdadeira corrida pela universalização da escrita (RESNICK; RESNICK, 1977).

Com as mudanças tecnológicas, o letramento começou a ser associado às novas habilidades necessárias para produzir tecnologias, dentre elas a principal era a linguagem de programação. A tecnologia digital passou a conduzir o significado do letramento. A Linguagem de programação nasceu dentro do contexto ideológico no qual o letramento era um bem moral cívico, uma contribuição cidadã (VEE, 2017).

Vee (2017) destaca quatro argumentos, emprestados da retórica das Campanhas de Alfabetização, que balizam os discursos favoráveis ao ensino massivo da linguagem

de programação: o empoderamento individual; a aprendizagem de novas formas de pensar por meio de resolução de problemas e tomada de decisão; o progresso coletivo da sociedade e o exercício individual da cidadania; e questões econômicas de empregabilidade.

Nos anos setenta e oitenta o empoderamento do indivíduo diante da tecnologia era a principal motivação dos que defendiam o ensino de códigos computacionais. Aliada à questão da cidadania estava, igualmente, a ideia de usar a linguagem de programação como um meio de desenvolver a capacidade de resolver problemas. Por outro lado, hoje o argumento central repousa sobre o mercado e a capacitação de pessoas empregáveis (VEE, 2017).

Visto somente pela lente da produtividade profissional, o letramento em linguagem de programação poderia tornar-se tão excludente quanto a escrita já foi no passado (VEE, 2017). Vee (2017) alerta que a ideologia implícita às campanhas de massa sobre o letramento computacional, em particular a “Hora do Código” (A HORA... 2017), objetiva a criação de força de trabalho para as grandes corporações. Essas iniciativas podem, portanto, servir como um reforço do mercado global dominante. Uma das consequências desse fenômeno seriam os empregados contratados à distância para trabalhar em suas casas e a precarização do trabalho. Muitas vezes esses funcionários são de outros países, possuem pouca autonomia e ganham um salário mais baixo (VEE, 2017).

Ao propor olhar para a linguagem de programação sob a ótica do seu uso social e através da lente do conceito de letramento, Vee (2017) traçou um paralelo histórico entre a escrita e os códigos computacionais, ressaltando a peso ideológico do termo letramento. Em paralelo, a autora (VEE, 2017) sublinha a importância da materialidade da tecnologia para moldar o pensamento dos indivíduos bem como a estrutura social na qual eles vivem, resgatando as ideias de Goody e Watt (1963). Os autores (GOODY; WATT, 1963; MCLUHAN, 1962) atribuem o impacto civilizacional da escrita à invenção dos alfabetos bem como à sua reprodução em diferentes suportes. Logo qualquer outro tipo de comunicação de massa que registrasse materialmente a fala poderia suplantar a supremacia dos textos escritos da “Gutenberg Galaxy” (MCLUHAN, 1962 p. 1).

Esta linha de pensamento foi duramente criticada a partir de 1980 por estudiosos dos Novos Letramentos que defendiam um olhar socialmente e culturalmente

contextualizado do letramento que levasse em conta a diversidade das mídias criadas pelo advento do computador (COPE e KALANTIZ, 2000; CAZDEN, B.; COPE; FAIRCLOUGH et al. 1996). Em 1990 o *New London Group* (doravante NLG) propôs uma perspectiva pedagógica do letramento ancorada nos multiletramentos. No campo da educação, destacou-se a proposta de uma pedagogia dos multiletramentos, pensada pelo NLG, que preocupava-se, já nos anos 90, com o impacto educacional das mudanças tecnológicas oriundas do advento dos computadores domésticos (CAZDEN, B.; COPE; FAIRCLOUGH et al. 1996). Afim de propor um caminho possível para a transformação educacional o NLG publicou um Manifesto da Pedagogia dos Multiletramentos (CAZDEN, B.; COPE; FAIRCLOUGH et al. 1996). Mais tarde Cope e Kalantzis (2000) escreveram um livro intitulado: “A Pedagogia dos multiletramentos, aprendendo por meio do design.” (COPE; KALANTIZIS, 2000). O título do livro já sintetiza a ideia central da Pedagogia dos Multiletramentos: o letramento em design que implica a consideração das dimensões multilinguísticas e multimodais dos letramentos. Mesmo sem negar a materialidade dos chamados novos letramentos, os grupos ligados a esse conceito de letramentos digitais (*New literacies e New London Group*) focam nos desdobramentos sociais e culturais no qual esses estão envolvidos (VEE, 2017)

Vee (2017) propõe uma abordagem de letramento que esteja entre a materialidade de Good e Watt (1963) e a inclusão dos diversos contextos socioculturais, defendida pelos Novos Letramentos e pelo NLG (CAZDEN, B.; COPE; FAIRCLOUGH et al. 1996). Nesse sentido um letramento seria uma inteligência material que passou a tecer a infraestrutura comunicacional de uma sociedade (VEE, 2017). Por isso, a pesquisadora prefere falar de letramento no singular. Ao pensar sob essa ótica a linguagem de programação estaria entrelaçada às formas de comunicação digitais que permeiam toda a contemporaneidade, tornando-se infraestrutural. Isso indicaria que códigos computacionais formam uma plataforma que, assim como a escrita, sustenta os outros letramentos. Do mesmo modo que a escrita, esse letramento estaria tecendo uma relação profunda com o poder. A linguista considera que deixar este poder somente nas mãos dos programadores seria desistir de uma parte significativa da compreensão de como o mundo atual funciona. Observar a escrita de códigos computacionais pelas lentes do letramento permitiria perceber o peso e a influência de seus usos sobre as relações profissionais, sociais, econômicas e culturais do mundo contemporâneo (VEE, 2017).

Vale ressaltar que o termo letramento computacional é usado por diSessa (2000) para definir uma inteligência material, relacionada à escrita de códigos computacionais, que estaria transformando-se em um letramento. Para diSessa (2000) os códigos computacionais deveriam ser ensinados universalmente. Mas essa tarefa seria restrita aos Cientistas da Computação.

Em contrapartida, a tese de Vee (2017) é de que a linguagem de programação encontra-se no meio de uma transformação histórica. A engrenagem dessa mudança seria a apropriação dessa ferramenta comunicacional por diferentes setores da sociedade. Assim, profissionais de áreas não correlatas à Ciência da Computação também deveriam poder ensinar e moldar os códigos computacionais de acordo com seus perfis e necessidades. Desse modo, Vee (2017) cunhou o termo em inglês “*Coding Literacy*”⁷⁸ afim de realçar seu posicionamento em relação a forma como a linguagem de programação pode ser aprendida, usada e ensinada.

No entanto, este trabalho não adota inteiramente a visão socio-material proposta por Vee (2017), ancorada na ideia de que uma inteligência material transforma-se em letramento quando passa a permear a infraestrutura e o cotidiano de uma sociedade. Por um lado, reconhece-se a influência dos dispositivos digitais e analógicos de escrita na construção do conhecimento humano assim como na configuração das relações sociais. Por outro lado, aferir o status de letramento somente às práticas de escrita amplamente usadas e valorizadas em uma sociedade poderia resultar no apagamento das vozes das minorias.

Prefere-se, então, ancorar o conceito de letramento sob a perspectiva de Soares (2001 [1998], p. 39) que define-o como sendo: “O estado ou condição que adquire um grupo ou indivíduo como consequência de se ser apropriado da escrita e de suas práticas sociais.” Ainda seguindo o raciocínio de Soares (2001 [1998]) saber ler e escrever é diferente de apropriar-se da escrita, usando-a para comunicar-se competentemente em uma sociedade letrada. Desta forma Soares (2004, p. 14) propõem a distinção entre a alfabetização que seria a “aquisição do sistema convencional de escrita” (SOARES, 2004 p. 14), e o letramento visto como o “desenvolvimento de habilidades de uso desse sistema em atividades de leitura e escrita, nas práticas sociais que envolvem a língua

⁷ Como o livro “*Coding Literacy: How Computer Programming is changing wrting*” de Annette Vee (2017) ainda não foi oficialmente traduzido, optou-se por não traduzir esse termo que é central na tese de Vee (2017).

escrita” (SOARES, 2004 p. 14).

Tomando por base nas definições de letramento e alfabetização de Soares (2001 [1998], 2004), sugere-se considerar letramento computacional o desenvolvimento das habilidades necessárias para o uso competente das práticas sociais de leitura e escrita de códigos de computadores. Já a alfabetização computacional seria o desenvolvimento da habilidade de ler e escrever uma ou mais linguagens de programação, compreendendo os princípios lógicos que regem esse sistema de escrita.

A ideia de “letramentos plataforma”⁹ de Vee (2017, p.29), por sua vez, parece apresentar alguns pontos de questionamento. Ora, os multiletramentos propostos pelo GNL (CAZDEN, B.; COPE; FAIRCLOUGH et al. 1996 e COPE; KALANTIZIS, 2000) vão além da escrita. Eles propõem um olhar cuidadoso para outras formas de expressão humana fundamentadas no design como, por exemplo, esculturas, quadros, grafites ou porcelanas chinesas. Letramentos esses que não estão obrigatoriamente ligados ao mundo digital, nem à escrita. Assim os multiletramentos não podem ser reduzidos aos gêneros multimodais virtuais e, portanto, extrapolam a ideia da linguagem de programação enquanto um letramento plataforma.

Por outro lado, considerar a escrita como plataforma de letramentos em design externos ao mundo virtual seria ignorar seus possíveis aspectos topológicos e quantificadores e, portanto, matemáticos. Essa observação é válida para qualquer design, mesmo os que estão inseridos em gêneros discursivos multimodais¹⁰.

Também vale lembrar que o pensamento matemático expressa-se de acordo com contextos sociais, históricos e culturais diversos, incluindo as sociedades não letradas (D’AMBROSIO, 2008). Para incorporar a multiculturalidade dos aspectos matemáticos dos designs sugere-se, nesse trabalho, a perspectiva da etnomatemática, definida por Ambrósio (2008, p. 164) como sendo “o reconhecimento que as ideias matemáticas, substanciadas nos processos de comparar, classificar, quantificar, medir, organizar e de inferir e de concluir, são próprias da natureza humana.”

Além do mais, Vee (2017, p. 97, tradução nossa) considera a linguagem de programação “um sistema simbólico complexo que pode ser implementado simultaneamente em vários computadores bem como em redes globais de informação e

⁹ Platform literacies

¹⁰ Gêneros discursivos multimodais são textos híbridos que podem mesclar imagem, som, movimentos e escritos (BEAUDOIN, 2002).

comunicação”¹¹. Sob a ótica complexa de Morin (1984, p. 56), escolhida para guiar as reflexões desse artigo, um sistema opera dinamicamente e recursivamente seguindo um “tetragrama” composto pelos movimentos de ordem, desordem, interação e organização. Logo, a ideia de uma plataforma contradiz esses movimentos não lineares, pois leva a pensar que a linguagem de programação seria a causa e o mundo digital o seu efeito.

No entanto, uma das características mais marcantes da tecnologia contemporânea é a rapidez com que os produtos digitais adaptam-se às necessidades dos usuários e do mercado. Em outras palavras, a interação das pessoas com as interfaces do mundo digital determina as reprogramações dos algoritmos usados para criá-las, causando a desordem. De acordo com Morin (1984) todo sistema em desordem procura reorganizar-se. Isto posto, os programadores podem mudar a, se preciso for, a própria escrita de códigos computacionais para atingir um equilíbrio estável ou uma ordem. Em paralelo, o uso dessas tecnologias comunicacionais¹² molda o comportamento das pessoas que as usam e/ou programam, provocando sucessivos ciclos espiralados de interação, desordem, organização e ordem. Usuários, programadores e a linguagem de programação estabelecem, portanto, movimentos típicos de um sistema aberto.

Tal comportamento sistêmico da linguagem de programação pode ser relacionado ao pensamento complexo de Morin (1984, 1986, 1990) no intuito de entrever caminhos para a vivência da complexidade por meio do ensino de códigos computacionais sob a perspectiva da auto-heteroecoformação (FREIRE; LEFFA, 2016[2013]). Importa, conseqüentemente, explorar as possíveis conexões entre a Complexidade (MORIN, 1984, 1986, 1990, 2002), o letramento computacional (VEE, 2007) e auto-heteroecoformação tecnológica docente (FREIRE, 2009; FREIRE; LEFFA, 2016[2013]).

COMPLEXIDADE, LETRAMENTO COMPUTACIONAL E AUTO-HETEROECOFORMAÇÃO : CONEXÕES POSSÍVEIS

Edgard Morin (1990) explica que propôs o pensamento complexo para oferecer uma alternativa à fragmentação do saber e à linearidade da “relação causa-efeito”

¹¹ “a complex symbolic system that could be deployed across multiple computers and global networks”(VEE, 2017, p. 97).

¹² Refiro-me aqui ao conceito de tecnologia comunicacional de Vee (2017, p.97) que considera que alguns tipos de tecnologia, com o alfabeto, o livro ou a linguagem de programação, facilitam a construção de determinados tipos de conhecimentos. Por exemplo, a linguagem de programação facilita o conhecimento da escrita procedural.

defendida pelo paradigma¹³ tradicional (FREIRE; LEFFA, 2016[2013], p. 62). A partir de estudos da teoria da informação, da cibernética, da teoria dos sistemas e da ideia de auto-organização, Morin (1990) estabeleceu quatro operadores que por meio dos quais a Complexidade é engendrada. O primeiro é o dialógico. Para Morin (1990, p. 106-107) a apreensão da realidade precisa passar pela manutenção de duas lógicas antagônicas e complementares, como, por exemplo, a vida e a morte.

O segundo é o recursivo que propõe uma ruptura com a ideia de causalidade linear. Em contrapartida, pensa-se em um fluxo circular no qual “os produtos e os efeitos são ao mesmo tempo causa e produtores daquilo que os produziu” (MORIN, 1990 p. 108). O terceiro é o hologramático. Para explicar este princípio o autor (MORIN, 1990) exemplifica as células do corpo que contêm toda a informação genética do organismo que as encerra. Enfim, calcado na lógica recursiva, que, por sua vez, está ligada ao conceito dialógico, este pilar da complexidade considera que o todo está nas partes da mesma forma que as partes estão presentes no todo. O quarto operador é o sistema, concebido por Morin (2014, p. 78, tradução nossa) como “um todo organizado”. É por meio da organização que o sistema “favorece o surgimento de qualidades novas, ausentes das partes [quando essas estão] separadas: as emergências”¹⁴ (MORIN, 2014, p. 78). Por isso, o sistema na complexidade evidencia um todo que supera a ideia cartesiana de um conjunto de partes, pois constitui-se em “alguma coisa maior que a soma de suas partes” (MORIN, 2014, p. 78). Assim a ideia moriniana de sistema permite a conexão das partes ao todo e a religação “dos conhecimentos fragmentados” (MORIN, 2014, p. 78).

Fundamentado nessa visão de mundo, Morin (1986, p.37, tradução nossa) descreve a computação como “um complexo organizador/produzidor de caráter cognitivo comportando uma instância informacional, uma instância simbólica, uma instância memorial e uma instância lógica”¹⁵ No aspecto informacional a computação serve-se do sistema binário de sim ou não, verdadeiro ou falso para processar e tratar informações. Tais informações são codificadas na forma de”¹⁶ (MORIN, 1986, p. 38) que armazenam informações. Isso exige uma capacidade de memorização do computador que deverá

¹³ Para Morin (2002), o paradigma é um conjunto de regras, valores e crenças que comandam a maneira de pensar, e mesmo de cogitar, o mundo em uma determinada sociedade, inserida em contexto histórico específico.

¹⁴ “tout organisé” (MORIN, , 2014, p. 78)

¹⁵ “un complexe organisateur/producteur de caractère cognitif comportant une instance informationnelle, une instance symbolique, une instance mémorielle, une instance logicielle.” (MORIN, 1986, p. 37)

¹⁶ “signes/symboles” (MORIN, 1986, p. 37).

extrair, inscrever, duplicar, apagar ou modificar os dados conforme for necessário. Tais ações da instância memorial da computação precisam seguir uma determinada lógica que, de acordo com Morin (1986, p.38, tradução nossa), estabelecem as condições de funcionamento da atividade computacional “que é diversas formas de manipulação/tratamento de signos/símbolos”¹⁷ Ou seja, a linguagem de programação está inserida na instância simbólica da computação, mas relaciona-se de forma recursiva, dialógica, hologramática e sistêmica com os cerne da lógica, da memória e da organização.

No que toca a auto-heteroecoformação, Freire e Leffa (2016 [2013], p. 69) explicam que Pineau e Patrick (2005) estabelecem três polos formativos. A autoformação que consiste na “responsabilização do indivíduo pela sua própria formação”; a heteroformação que é “a ação dos indivíduos uns sobre os outros”; e a ecoformação que é “a ação do meio ambiente sobre os indivíduos” (LEFFA; FREIRE, 2016 [2013], p. 69). Os autores (FREIRE; LEFFA, 2016 [2013], p. 66, grifos dos autores) interpretaram essa teoria tripolar de formação de Pineau e Patrick (2005) à luz de “quatro dimensões distintas: *ação, sujeito, objeto e relações*” (FREIRE; LEFFA, 2016 [2013] p. 70, grifos dos autores).

A dimensão da ação relaciona-se de forma recursiva e circular com os três polos de formação. Ao agir sobre si mesmo o sujeito pode também agir de forma recursiva e retroalimentadora “sobre os outros e sobre o ambiente” (FREIRE; LEFFA, 2016 [2013] p. 70).

Ainda ampliando o conceito de auto-heteroecoformação de Freire (2009), Freire e Leffa (2016 [2013], p. 71) consideram que a dimensão do sujeito movimenta-se sistemicamente de maneira bidirecionada. Ora os indivíduos situam-se intrassistemicamente “no plano interno, individual, dentro do sistema constituído pelo próprio sujeito;”. Ora esses estão operando movimentos “intersistêmicos quando se estabelecem em um plano social e/ou ecológico, interagindo com o sistema social e os vários sistemas que compõem o ambiente, em um sentido ecológico mais abrangente”. (FREIRE; LEFFA, 2016[2013], p.. 71)

Quando o sujeito age em direção a um objeto, esse também articula as dimensões anteriores bem como os pólos formativos da autoformação, da heteroformação e da ecoformação. Já a dimensão das relações “focaliza as interações

¹⁷ “qui est manipulation/traitement, sous formes e modes divers, de signes/symboles” (MORIN, 1986, p. 38).

que interligam o sujeito e sua ação quando direcionado ao objeto da formação (FREIRE; LEFFA, 2016[2013], p. 72)”. Uma vez que os movimentos do sujeito em direção ao seu objeto são intrassistêmicos, as relações ficam no plano individual. Por outro lado, quando os movimentos do sujeito são intersistêmicos, as relações “extrapolam os sistemas individuais e se exteriorizam, socializando-se, principalmente. (FREIRE; LEFFA, 2016[2013], p. 72)”. Essas relações também podem ir além da socialização quando o sujeito age sobre o ambiente, “pois assumem uma conotação ecológica” (FREIRE; LEFFA, 2013 p. 73).

Para os autores (FREIRE; LEFFA, 2013), os polos da autoformação, da heteroformação e da ecoformação seguem a dinâmica da complexidade (MORIN, 1990), pois movem-se dialogicamente, recursivamente, de forma hologramática e sistêmica, a partir das perspectivas das dimensões da ação, do sujeito, do objeto e das relações. Essas dimensões estão articuladas, evidenciando, assim, “os movimentos de personalização, socialização e ecologização, que a caracterizam”. (FREIRE; LEFFA, 2013 p. 75)

Seguindo o raciocínio de Morin (1986, 1990) a linguagem de programação está contida dentro da instância simbólica (signo) da computação, formando um sistema complexo que abarca outros três eixos: o da informação, o da memória e o da lógica (regras e condições de funcionamento). O autor (MORIN 1986, 1990) ainda ressalta que é nas relações, interações e interconexões dessas instâncias que o mundo digital é tecido.

Consequentemente a aprendizagem dos códigos, sob a perspectiva de suas práticas sociais, ou seja, do letramento segundo Soares (2001[1998], 2004) - bem como do sentido dado ao termo “*Coding Literacy*” por Vee (2017) - poderia facilitar a vivência dos aspectos dialógico, recursivo, hologramático e sistêmico da computação, estendendo essa experiência ao reconhecimento da complexidade (GALVANI; PINEAU, 2012).

Assim, os códigos permitiriam que o professor conseguisse visualizar, interconectar, tecer e religar, na forma de um holograma, o todo da computação (MORIN, 1986) e suas múltiplas interfaces. Ao mesmo tempo a vivência recursiva dos papéis de usuário, criador/cocriador de tecnologia ofereceria a oportunidade de movimentar-se entre os diversos polos e dimensões do processo formativo sugeridos por Freire e Leffa (2016[2013]). A aprendizagem da escrita dos códigos, dessa forma

conceituada, parece ter o potencial de permitir que o professor se construa e desconstrua enquanto “um cidadão cada vez mais pleno: um sujeito individual, social, tecnológico e, sem dúvida, planetário, protagonista e responsável pelo mundo em que vive” (FREIRE; LEFFA, 2016[2013]).

CONSIDERAÇÕES FINAIS

Este artigo propôs um olhar complexo (MORIN, 1986, 1990, 2002 e 2014) sobre a aprendizagem da linguagem de programação, religando-a à ideia de letramento computacional (*Coding Literacy*) (VEE, 2013, 2017) revisitado à luz do conceito brasileiro do termo descrito por Soares (2001[1998], 2004).

O presente artigo teve como escopo de reflexão o papel da linguagem de programação na tessitura da miríade de letramentos e habilidades inerentes ao universo digital. Pareceu conceitualmente factível abordar a experiência da aprendizagem da escrita de códigos por meio da auto-heteroconformação tecnológica complexa (FREIRE; LEFFA, 2013), a fim de propiciar ao professor as condições necessárias para a descoberta da complexidade (GALVANI; PINEAU, 2012). Por causa dos movimentos recursivos dos códigos computacionais, que permeiam as relações entre humanos e máquinas, pensou-se na linguagem de programação como um fio de Ariadne: um guia que conduz quem o encontra para dentro e para fora dos incontáveis labirintos de conexões possíveis do mundo virtual e, ao mesmo tempo, indica a porta de saída que leva ao mundo concreto. É sob a perspectiva da ampliação e democratização dos usos sociais da escrita de códigos computacionais que inscreve-se as aproximações desse artigo ao termo *Coding Literacy* de Vee (2017). Nesse sentido, o professor/programador/criador estaria, assim, apropriando-se da linguagem que, como alertou Vee (2017), está reconfigurando as estruturas sociais contemporâneas. Tal apropriação da linguagem de programação pelo professor, no sentido de “assumi-la como “sua propriedade”” (SOARES, 2001[1998], p. 39, grifo da autora), pode reconstruí-lo enquanto um cidadão crítico, capaz de interferir positivamente na sociedade e ajudar seus educandos a trilharem o mesmo caminho. Finalmente, esse artigo apenas procura horizontes de reflexão acerca da formação tecnológica docente, deixando ainda em aberto um tema que, devido à complexidade de tessituras possíveis, revela-se inesgotável.

REFERÊNCIAS

ATWOOD, J. Please Don't Learn to Code. *Coding Horror*, programming and human factors, 15 mai.2012. Disponível em: <https://blog.codinghorror.com/please-dont-learn-to-code/>. Acesso em: 10 nov. 2018.

AUGARTEN, S. *Bit by Bit: an illustrated history of computers*. Boston: Houghton Mifflin Co., 1984.

BEAUDOIN, V. De la publication à la conversation. Lecture et écriture électroniques. *Réseaux*, n. 119, p. 199-224, 2002. Disponível em: http://www.cairn.info/article.php?ID_REVUE=RESeID_NUMPUBLIE=RES_116eID_ARTICLE=RES_116_0199. Acesso em: 15 nov. 2018.

BESSERAT, D. *Before Writing, vol. 1: from counting to cuneiform*. Austin: University Texas Press, 1992.

CAZDEN, B.; COPE; FAIRCLOUGH *et al.* *A pedagogy of multiliteracies: Designing social futures*. *Harvard Educational Review*, 1996, p. 60 – 92.

CAPELO, R. Hora do Código estreia no País para ensinar programação a 1 milhão. *Estadão*, DEZ. 2014. Especial para o Estado. Disponível em: <https://link.estadao.com.br/noticias/geral,hora-do-codigo-estrela-no-pais-para-ensinar-programacao-a-1-milhao,10000030003>. Acesso em: 26/05/2019.

COPE, B.; KALANTZIS, M. (Eds.). *Multiliteracies: literacy learning and the design of social futures*. London/New York: Routledge, 2000.

CUNHA, M.B. *Para saber mais: fontes de informação em ciência e tecnologia*. Brasília : Briquet Lemos, 2001.

D'AMBROSIO, U. Para uma abordagem multicultural: o Programa Etnomatemática. *Revista Lusófona de Educação*, 11, 2008, p. 163 - 167. Entrevista concedida a Nunes Vieira.

DISESSA, A. *Changing Minds: Computers, Learning and Literacy*. Cambridge: MIT Press, 2000.

FREIRE, M. M; LEFFA, V. A auto-heteroecoformação tecnológica. In: MOITA, L. (Org.). *Linguística Aplicada na modernidade recente*. São Paulo: Parábola, 2013 [2016], p. 59-78.

FREIRE, M. Formação tecnológica de professores: problematizando, refletindo, buscando. In: SOTO, U.; MAYRINK, M.; GREGOLIN, I. *Linguagem, educação e virtualidade - experiências e reflexões*. São Paulo: Cultura Acadêmica, 2009, p. 13-28.

GALVANI, P.; PINEAU, G. Experiências de vida e formação docente: religando os saberes – um método reflexivo e dialógico. In: MORAES, M.C.; ALMEIDA, M.C (Org.). *Os sete saberes necessários à Educação do Presente*. Rio de Janeiro: WAK Editora, 2012, p. 185-225.

GOODY, J.; WATT, I. The Consequences of Literacy. *Comparative Studies in Society and History*, Cambridge: Cambridge University Press, Vol. 5, No. 3, 1963, pp. 304-345.

GROGAN, D. *Science and technology: an introduction to the literature*. London: Clive Bingley, 1970.

GUYNN, J. [Silicon Valley launches campaign to get kids to code](https://www.latimes.com/business/la-xpm-2013-feb-26-la-fi-tn-silicon-valley-launches-campaign-to-get-kids-to-code-20130226-story.html). *Los Angeles Times*, Los Angeles, FEV. 2013. Seção Business. Disponível em: <<https://www.latimes.com/business/la-xpm-2013-feb-26-la-fi-tn-silicon-valley-launches-campaign-to-get-kids-to-code-20130226-story.html>>. Acesso em: 26/06/2019.

HIGOUNET, C. *Uma história concisa da escrita*. São Paulo: Parábola, 2003.

ISAACSON, W. *Steve Jobs*. New York; Toronto: Simon e Schuster, 2011.

KAY, A. e GOLDBERG, A. Personal dynamic media. *IEEE Computer*, volume 10, número 3, 1972 p. 254–263.

MCLUHAN, M. *The Gutenberg galaxy: The making of typographic man*. Toronto: University of Toronto Press, 1962.

MORIN, E. *Science et conscience de la complexité: échanges avec Edgar Morin*. Aix-en-Provence: Librairie de l'Université, 1984.

_____. *La méthode 3: la connaissance de la connaissance*. Paris: Seuil, 1986.

_____. *Introduction à la pensée complexe*. Paris: Seuil, 1990.

_____. *O método 5. A humanidade da humanidade*. Porto Alegre: Sulina, 2002.

_____. *Enseigner à Vivre: um manifesto por changer l'Éducation*. Paris: Actes Sud, 2014.

OLIVEIRA, R. *Informática educativa: dos planos e discursos à sala de aula*. Campinas: Papirus, 2007.

PERLIS, A. *The Computer and the University, Computers and the World of the Future*. Cambridge: Massachusetts Institute of Technology (MIT) Press, 1964.

PAPERT, S. ; SOLOMON, C. *Twenty Things to Do With a Computer*. Cambridge: Massachusetts Inst. of Tech., Artificial Intelligence Lab., 1971. Disponível em: <https://files.eric.ed.gov/fulltext/ED077240.pdf>. Acesso em: 17 nov. 2018.

PINEAU, G.; PATRICK, P. *Transdisciplinarité et formation*. Paris: L'Harmattan, 2005.

PRESNSKY, M. Digital Natives, Digital Immigrants. *On the Horizon*, v.9 n. 5. West Yorkshire: MCB University Press out. 2001, p. 1-6.

PROGRAMAÊ. Disponível em: <http://fundacaotelefonica.org.br/projetos/programae/>. Acesso em: 25 nov. 2018.

RESNICK; D.P; RESNICK.L.B. The Nature of Literacy: an historical explanation. *Harvard Educational Review* 47, n° 3, 1997, p. 370-385.

RUSHKOFF, D. *Program or Be Programmed: Ten Commands for a Digital Age*. New

York: OR Books, 2010.

SAMPAIO, F.F; ELIA, M.F. *Projeto Um Computador por Aluno*, pesquisas e perspectivas. Rio de Janeiro: NCE/UFPR, 2012, p. 239-247.

SOARES, M. *Letramento: um tema em três gêneros*. Belo Horizonte: Autêntica, 2001[1998].

_____. Letramento e alfabetização: as muitas facetas. *Revista Brasileira de Educação*, Rio de Janeiro; v. 25, jan./fev./mar./abr. 2004.

STEIN, D. *Ada: a life and a legacy*. Cambridge: MIT Press, 1985.

VASQUEZ, V. Critical Literacy. In: *Oxford Research Encyclopedias, Education*. Disponível em: <http://oxfordre.com/education/view/10.1093/acrefore/>

[9780190264093.001.0001/acrefore-9780190264093-e-20](http://oxfordre.com/education/view/10.1093/acrefore/9780190264093.001.0001/acrefore-9780190264093-e-20). Acesso em: 12/11/2018.

VEE, A. Understanding Computer Programming as a Literacy. *Literacy in Composition Studies*, v.1, n.2, 2013 p. 42-64.

Disponível em: <http://licsjournal.org/OJS/index.php/LiCS/article/view/24>>. Acessado em: 10 dez. 2017.

_____. *Code Literacy: How Computer Programming ins changing writing*. Cambridge, Massachusetts: MIT Press, 2017.

WEBER, S. *The success of Open Source*. Cambridge: Harvard University Press, 2004.