

EDUCAÇÃO INTEGRAL POR MEIO DO PENSAMENTO COMPUTACIONAL

**Letramento em Programação:
Relatos de Experiência e Artigos Científicos**

Editora Appris Ltda.

1ª Edição - Copyright© 2019 dos autores

Direitos de Edição Reservados à Editora Appris Ltda.

Nenhuma parte desta obra poderá ser utilizada indevidamente, sem estar de acordo com a Lei nº 9.610/98. Se incorreções forem encontradas, serão de exclusiva responsabilidade de seus organizadores. Foi realizado o Depósito Legal na Fundação Biblioteca Nacional, de acordo com as Leis nos 10.994, de 14/12/2004, e 12.192, de 14/01/2010.

Catálogo na Fonte

Elaborado por: Josefina A. S. Guedes

Bibliotecária CRB 9/870

E244e
2019 Educação Integral por meio do pensamento computacional:
Letramento em programação: relatos de experiência e artigos científicos /
Amilton Rodrigo de Quadros Martins, Adelmo Antonio da Silva Eloy
(Organizadores). - 1. ed. - Curitiba: Appris, 2019.
363 p. ; 23 cm - (Educação, tecnologias e transdisciplinaridade)

Inclui bibliografias
ISBN 978-85-473-4017-9

1. Tecnologia educacional. 2. Programação (Computadores) – Educação básica.
3. Programação (Computadores) – Escolas públicas. I. Martins, Amilton Rodrigo
de Quadros (org.). II. Eloy, Adelmo Antonio da Silva (org.). III. Título. IV. Série.

CDD - 371.334

Livro de acordo com a normalização técnica da APA

Appris
editora

Editora e Livraria Appris Ltda.
Av. Manoel Ribas, 2265 – Mercês
Curitiba/PR – CEP: 80810-002
Tel. (41) 3156 - 4731
www.editoraappris.com.br

Printed in Brazil
Impresso no Brasil

EDUCAÇÃO INTEGRAL POR MEIO DO PENSAMENTO COMPUTACIONAL

**Letramento em Programação:
Relatos de Experiência e Artigos Científicos**

Amilton Rodrigo de Quadros Martins
e Adelmo Antonio da Silva Eloy
(Organizadores)

Appris
editora

FICHA TÉCNICA

EDITORIAL	Augusto V. de A. Coelho Marli Caetano Sara C. de Andrade Coelho
COMITÊ EDITORIAL	Andréa Barbosa Gouveia - UFPR Edmeire C. Pereira - UFPR Ireneide da Silva - UFC Jacques de Lima Ferreira - UP Marilda Aparecida Behrens - PUCPR
DIREÇÃO DE PRODUÇÃO	Sara C. de Andrade Coelho
ASSESSORIA EDITORIAL E REVISÃO	Alana Cabral
PRODUÇÃO EDITORIAL	Lucas Andrade
DIAGRAMAÇÃO	Bruno Ferreira Nascimento
CAPA	Fernando Nishijima
LIVRARIAS E EVENTOS	Estevão Misael
GERÊNCIA DE FINANÇAS	Selma Maria Fernandes do Valle

COMITÊ CIENTÍFICO DA COLEÇÃO EDUCAÇÃO, TECNOLOGIAS E TRANSDISCIPLINARIDADE

DIREÇÃO CIENTÍFICA	Dr.ª Marilda A. Behrens (PUCPR)	Dr.ª Patrícia L. Torres (PUCPR)
CONSULTORES	Dr.ª Ademilde Silveira Sartori (Udesc)	Dr.ª Iara Cordeiro de Melo Franco (PUC Minas)
	Dr. Ángel H. Facundo (Univ. Externado de Colômbia)	Dr. João Augusto Mattar Neto (PUC-SP)
	Dr.ª Ariana Maria de Almeida Matos Cosme (Universidade do Porto/Portugal)	Dr. José Manuel Moran Costas (Universidade Anhembi Morumbi)
	Dr. Artieres Estevão Romeiro (Universidade Técnica Particular de Loja-Ecuador)	Dr.ª Lúcia Amante (Univ. Aberta-Portugal)
	Dr. Bento Duarte da Silva (Universidade do Minho/Portugal)	Dr.ª Lucia Maria Martins Giraffa (PUCRS)
	Dr. Claudio Rama (Univ. de la Empresa-Uruguai)	Dr. Marco Antonio da Silva (Uerj)
	Dr.ª Cristiane de Oliveira Busato Smith (Arizona State University /EUA)	Dr.ª Maria Altina da Silva Ramos (Universidade do Minho-Portugal)
	Dr.ª Dulce Márcia Cruz (Ufsc)	Dr.ª Maria Joana Mader Joaquim (HC-UFPR)
	Dr. Edméa Santos (Uerj)	Dr. Reginaldo Rodrigues da Costa (PUCPR)
	Dr.ª Eliane Schlemmer (Unisinos)	Dr. Ricardo Antunes de Sá (UFPR)
	Dr.ª Ercília Maria Angeli Teixeira de Paula (UEM)	Dr.ª Romilda Teodora Ens (PUCPR)
	Dr.ª Evelise Maria Labatut Portilho (PUCPR)	Dr. Rui Trindade (Univ. do Porto-Portugal)
	Dr.ª Evelyn de Almeida Orlando (PUCPR)	Dr.ª Sonia Ana Charchut Leszczynski (UTFPR)
	Dr. Francisco Antonio Pereira Fialho (Ufsc)	Dr.ª Vani Moreira Kenski (USP)
	Dr.ª Fabiane Oliveira (PUCPR)	

SOBRE O INSTITUTO AYRTON SENNA

O Instituto Ayrton Senna, criado há 25 anos, trabalha para ampliar as oportunidades de crianças, adolescentes e jovens brasileiros desenvolverem seus potenciais por meio de uma educação de qualidade, equitativa e integral. Sua missão é gerar oportunidades para que os estudantes do século 21 se desenvolvam por completo, tornando-se capazes de construir seus projetos de vida e criar novas possibilidades de futuro para si mesmos e para a sociedade 5.0.

Para isso, o Instituto atua como um centro de inovação em educação, trabalhando em parceria com gestores públicos, educadores e pesquisadores para a produção e sistematização de conhecimentos para apoiar a construção de políticas públicas e práticas educacionais com base em evidências. O Instituto também desenvolve, implementa e avalia propostas educacionais diversificadas para apoiar redes de ensino e educadores que queiram promover a educação integral para todos os estudantes, desde a alfabetização até o ensino médio. Todo o conhecimento produzido é compartilhado com mais atores por meio de iniciativas de formação, difusão, cooperação técnica e transferência de tecnologia.

Dentre as propostas educacionais desenvolvidas pelo Instituto Ayrton Senna, destacamos o Letramento em Programação, cuja temática é central no desenvolvimento desta obra.

SOBRE A IMED

A Imed é uma Instituição de Ensino Superior com sede em Passo Fundo-RS e campus em Porto Alegre-RS, com apenas 15 anos de história, e há seis anos tem como foco estratégico investir em projetos de impacto na educação básica, por meio do Laboratório de Ciência e Inovação para a Educação (InovaEdu). Em 2016, o InovaEdu/Imed iniciou a parceria com o Instituto Ayrton Senna, criando o Núcleo Norte Gaúcho do Letramento em Programação, que até 2018 contou com a participação de 14 municípios e suas redes municipais de educação, construindo um momento ímpar e histórico na região, criando uma nova perspectiva rumo à educação para o século XXI no norte do RS.

Ainda em 2017, o InovaEdu/Imed, juntamente com o Instituto Ayrton Senna e parceiros regionais, articula a criação do Arranjo de Desenvolvimento de Educação (ADE) Norte Gaúcho, com a proposta de apoiar a efetivação de um regime de colaboração territorial, aproximando os 29 municípios participantes em uma pauta coletiva de educação pública de qualidade.

No contexto desta obra, a Imed ampliou o escopo de sua parceria com o Instituto Ayrton Senna para a produção de um livro que sistematize e dissemine conhecimentos em torno do Letramento em Programação. A visão da Imed aponta que a parceria com institutos e empresas para impacto efetivo na educação básica é missão de toda instituição de ensino superior realmente comprometida com o avanço do mundo pela educação, e que almeja inspirar quem transforma.

PREFÁCIO

Preparar as pessoas para as demandas deste século requer uma educação que seja capaz de potencializar as qualidades humanas na perspectiva de um desenvolvimento pleno em consonância com a Constituição Federal. Em seu artigo 205, o documento defende que “a educação, direito de todos e dever do Estado e da família, será promovida e incentivada com a colaboração da sociedade, visando ao pleno desenvolvimento da pessoa, seu preparo para o exercício da cidadania e sua qualificação para o trabalho”.

Isso significa dizer que precisamos ir além do desenvolvimento cognitivo tradicional e buscar mecanismos e estratégias que possam promover o desenvolvimento das chamadas competências socioemocionais. Nessa direção, estudos revelam que o uso de tecnologias digitais no ambiente escolar, integradas a um currículo que vise, de forma intencional, o desenvolvimento dessas habilidades, pode ser uma estratégia assertiva para promover, entre outras habilidades, a colaboração, o pensamento crítico, a criatividade, a inovação e a comunicação.

Ensinar às crianças e aos jovens do século 21 conceitos de programação, habilitando-os a se comunicar, produzir conhecimentos, resolver problemas e serem protagonistas, pode ser considerado um componente curricular importante para a promoção da educação integral. Foi pensando assim que há cinco anos o Instituto Ayrton Senna iniciou o projeto de Letramento em Programação nas escolas municipais de Itatiba (SP), em parceria com a Fundação Lemann, a Secretaria Municipal de Educação e a Universidade São Francisco (USF), com o apoio decisivo das empresas Boeing e Linx.

O que se viu ao longo dos anos foram estudantes cada vez mais engajados e motivados, de tal forma que o

projeto se tornou política pública do município de Itatiba mediante a Lei nº 5.079, de 8 de dezembro de 2017. O sucesso do projeto foi além dos limites do município, contagiando cidades vizinhas que incluíram o Letramento em Programação no currículo escolar, como foi o caso de Morungaba e Vinhedo. Ao mesmo tempo, uma rede nacional de Letramento em Programação começou a ser estruturada, com destaque para Passo Fundo, no Rio Grande do Sul, e Caruaru, no estado de Pernambuco, como polos estratégicos. Hoje, o Letramento em Programação está em quatro regiões do Brasil, formando uma rede colaborativa que tem no seu DNA a participação do terceiro setor, por meio do Instituto Ayrton Senna, de universidades locais, empresas apoiadoras e das secretarias municipais de Educação.

Tal iniciativa vem ajudando a levar ao chão de escola a quinta competência da Base Nacional Comum Curricular (BNCC), que trata da compreensão, utilização e criação de tecnologias digitais para que as linguagens de programação e de conceitos e práticas de computação possam cada vez mais promover a construção de uma escola que responda às demandas do século 21. Ou seja, que a cultura digital seja utilizada de forma crítica, significativa, reflexiva e ética nas diversas práticas sociais, incluindo as escolares, para se comunicar, acessar e disseminar informações, produzir conhecimentos, resolver problemas e exercer protagonismo e autoria na vida pessoal e coletiva.

As crianças e jovens precisam de uma escola que traga significado para suas vidas, e o Letramento em Programação é um dos exemplos que promove tal significado por meio do uso de jogos e aplicativos construídos por eles próprios na perspectiva não só de resolver problemas reais no ambiente escolar, mas também do dia a dia do município. Um dos momentos mais marcantes é o Tech Oscar – evento de encerramento anual do projeto –, no qual os alunos apresentam para a comunidade escolar

e suas famílias os projetos desenvolvidos com base em programação. Dá gosto ver o brilho nos olhos desses alunos, assim como o encantamento e o engajamento que dão significado aos seus estudos.

Este livro retrata não só o que aprendemos sobre o tema até aqui, mas também sobre como estamos aprendendo. Os relatos de experiências de professores da Rede Nacional do Letramento em Programação representam o caráter prático de nossa atuação. Os artigos convidados e selecionados, a partir de edital aberto, demonstram nossa abertura para aprender e colaborar em rede. Estamos aprendendo muito com instituições que desenvolvem trabalhos similares, e acreditamos que a transformação da escola pública será muito mais rápida se unirmos forças em torno das mesmas causas.

Boa leitura!

Mozart Neves Ramos

*Diretor de Articulação e
Inovação do Instituto Ayrton Senna*

SUMÁRIO

INTRODUÇÃO	16
-------------------------	-----------

SEÇÃO 1

ARTIGOS DE REFERÊNCIA SOBRE O TEMA.....	22
--	-----------

CAPÍTULO 1

ENTENDENDO A PROGRAMAÇÃO DE COMPUTADORES COMO UM LETRAMENTO (TRADUZIDO)	24
--	-----------

Annette Vee – University of Pittsburgh

CAPÍTULO 2

DIRETRIZES DE ENSINO DE COMPUTAÇÃO NA EDUCAÇÃO BÁSICA	62
--	-----------

*Leila Ribeiro, Lisandro Zambenedetti Granville,
Dalton Serey, Simone André da Costa Cavalheiro*

SEÇÃO 2

ARTIGOS CIENTÍFICOS SOBRE PENSAMENTO COMPUTACIONAL NA EDUCAÇÃO BÁSICA	82
--	-----------

CAPÍTULO 3

ENSINO DE PENSAMENTO COMPUTACIONAL COM JOGOS E INTERNET DAS COISAS	86
---	-----------

*Bruno da Silva Rodrigues, Ana Grasielle Dionísio Corrêa
Nizam Omar, Valéria Farinazzo Martins*

CAPÍTULO 4

DETECT - UM MODELO PARA A AVALIAÇÃO DE UNIDADES INSTRUCCIONAIS PARA O ENSINO DE COMPUTAÇÃO NA EDUCAÇÃO BÁSICA	120
--	------------

Christiane Gresse von Wangenheim, Giani Petri, Adriano F. Borgatto

CAPÍTULO 5

A MODELAGEM MATEMÁTICA COMO METODOLOGIA PARA O DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL POR MEIO DA CRIAÇÃO DE OBJETOS DE APRENDIZAGEM..... 144

Márcia Regina Kaminski, Clodis Boscarioli

CAPÍTULO 6

DESENVOLVIMENTO E IMPACTO DA INTEGRAÇÃO CURRICULAR DE PROGRAMAÇÃO E PENSAMENTO COMPUTACIONAL EM TODAS AS DISCIPLINAS E SÉRIES DO ENSINO FUNDAMENTAL DA FUNDAÇÃO TORINO ESCOLA INTERNACIONAL172

Ana O. Rodrigues, Ana Márcia A. M. Paiva, Sabrina M. de Oliveira, Márcia L. A. A. Naves

CAPÍTULO 7

PENSAMENTO COMPUTACIONAL E PROGRAMAÇÃO: IMPACTOS NA FORMAÇÃO DE PROFESSORES E CONTRIBUIÇÕES PARA PRÁTICAS PEDAGÓGICAS INTERDISCIPLINARES206

Eliel Constantino da Silva, Maria Teresa Zampieri, Sueli Liberatti Javaroni

CAPÍTULO 8

ESTIMULANDO O PENSAMENTO COMPUTACIONAL POR MEIO DE OFICINAS LÚDICAS NO LITORAL NORTE GAÚCHO 232

*Vitória de Souza Fabricio, Mauricio Braga Julio,
Augusto Weiland, Anelise Lemke Kologeski*

SEÇÃO 3

RELATOS DE EXPERIÊNCIAS INOVADORAS.....260

CAPÍTULO 9

LETRAMENTO DIGITAL NORTE GAÚCHO: UMA POSSIBILIDADE PARA O DESENVOLVIMENTO DE HABILIDADES SOCIOCOMPORTAMENTAIS EM ALUNOS COM TDAH E COMORBIDADES264

Irani Bernadete Roani, Ivânia Campigotto Aquino

CAPÍTULO 10

EMEF ANTONIO ZAMBRZYCKI DE GETÚLIO VARGAS ABRINDO AS PORTAS PARA O LETRAMENTO EM PROGRAMAÇÃO 270

Vanessa Polli

CAPÍTULO 11

RELATO DA EXPERIÊNCIA DE UTILIZAÇÃO DA FERRAMENTA SCRATCH E APP INVENTOR NO PROCESSO DE ENSINO-APRENDIZAGEM DE ALUNOS DO MUNICÍPIO DE MARAU/RS..... 278

*Diego Fernando Grzibowski, Ramon Fioravanso,
Simone Costenaro Ribeiro, Maikyeli Orsato Decesaro*

CAPÍTULO 12

A CULTURA MAKER INSERIDA NO LETRAMENTO EM PROGRAMAÇÃO.....286

Peter Rodrigo Trento

CAPÍTULO 13

REALIZANDO A ATIVIDADE DA CONSTRUÇÃO DO JOGO DE LABIRINTO... 294

Érica Luiza de Oliveira

CAPÍTULO 14

AS DISTINTAS EXPERIÊNCIAS DOCENTES QUE FAVORECEM AS PRÁTICAS PEDAGÓGICAS..... 302

Eduardo Mauricio Moreno Pinto

CAPÍTULO 15

UM OLHAR SOBRE AS CONSTRUÇÕES – ANALISANDO INTERAÇÕES E APRENDIZAGENS312

Sonia Mara Martins de Mattos Souza

CAPÍTULO 16

“TROLÓLÓ: QUE BARULHO É ESSE?” 322

Daniel Pereira de Souza

CAPÍTULO 17

TECNOLOGIA, DESENVOLVIMENTO COGNITIVO E AUTOCONHECIMENTO CAMINHANDO JUNTOS, É POSSÍVEL? 330

Eliana Maria Fattori Calza

CAPÍTULO 18

A DESAFIANTE E ENCANTADORA MISSÃO CHAMADA LETRAMENTO EM PROGRAMAÇÃO.....340

Nariana de Oliveira Tavares

SOBRE OS AUTORES.....348

INTRODUÇÃO

Você consegue imaginar como o mundo estará daqui a 15 anos? Que habilidades o estudante que começa a ser alfabetizado hoje terá que desenvolver ao longo do seu percurso escolar para estar preparado para os desafios do futuro?

Inserida em um mundo cada vez mais complexo, a escola do século 21 deverá enxergar seus estudantes como indivíduos multifacetados, criando oportunidades para que cada um deles torne-se um cidadão autônomo capaz de construir novos futuros e desenvolver todo o seu potencial. Para enfrentar esse desafio, é preciso encarar a educação sob uma nova perspectiva: a partir de um olhar de educação integral.

Sob essa perspectiva, a educação deve criar oportunidades para que a criança, o adolescente e o jovem possam desenvolver habilidades necessárias à sua vida em sociedade e ao seu protagonismo, em todas as dimensões – incluindo tanto os aspectos cognitivos, quanto os socioemocionais (saiba mais no box a seguir) e outros. Assim, do mesmo modo que são estimulados a aprender o conteúdo das disciplinas, os estudantes também podem receber apoio explícito e intencional para o desenvolvimento de competências como colaboração, comunicação, responsabilidade e empatia. Dessa forma, estarão mais preparados para criar uma nova realidade mais sustentável, igualitária e criativa no futuro.

Para alcançar esses objetivos, a educação integral deve ser a linha norteadora de todas as práticas e políticas educacionais, dentre elas o letramento em novas linguagens necessárias ao século 21. Entre esses novos códigos, estão os do mundo digital, aprendidos por meio do desenvolvimento do pensamento computacional, que é definido como a análise e resolução de problemas por meio da construção de algoritmos. O letramento em linguagens de computação e o pensamento computacional colaboram **não apenas para o desenvolvimento**

cognitivo dos estudantes, mas também para o seu desenvolvimento pleno ao promover, entre outras habilidades, o raciocínio lógico.

Quando o propósito da educação deixa de ser a mera transmissão ou o acúmulo de informações e passa a ser esse desenvolvimento pleno, é que se pode dizer que o que se busca é uma educação integral. É com ela que se pode assegurar às novas gerações o direito de aprender e produzir tanto quanto de se conhecer e conviver com os outros e com o mundo ao redor. Nesse **olhar sobre o desenvolvimento pleno, que é o que se valoriza na educação integral**, a aprendizagem de conhecimentos e habilidades cognitivas segue sendo relevante e precisa ser garantida com qualidade, mas tem que conviver com um trabalho igualmente intencional para o desenvolvimento de competências para a vida, também chamadas de competências socioemocionais.

COMPETÊNCIAS SOCIOEMOCIONAIS PODEM SER ENTENDIDAS COMO

Características que todas as pessoas possuem, em níveis diversificados, que envolvem a capacidade de mobilizar, articular e colocar em prática conhecimentos, valores, atitudes e habilidades para se relacionar consigo e com os outros, para estabelecer e atingir objetivos, para tomar decisões e enfrentar situações adversas ou novas. Por exemplo: a colaboração, a assertividade e a resiliência.

*Essas competências manifestam-se no **modo como pensamos, sentimos e nos comportamos** e são maleáveis, ou seja, **podem ser desenvolvidas por meio de experiências** formais e informais de aprendizagem – mas, para isso, é crucial que exista intencionalidade, pois quando se tem consciência das competências que se busca desenvolver, é possível apropriar-se do aprendizado e ampliar o repertório de habilidades. Estudos indicam que elas são importantes **impulsionadoras de realizações** ao longo da vida, como na saúde e no bem-estar de cada um, na qualidade de relações sociais e também nos processos de aprendizagem (pois estão ligadas a funções como a memória e a atenção, podendo tanto potencializar, quanto prejudicar seu desempenho).*

Assim, a educação brasileira possui hoje duas demandas igualmente prioritárias:

De um lado, acelerar os esforços para saldar a dívida educacional que se revela nos baixos índices de aprendizado, no abandono e na distorção idade-ano escolar, reflexos da fragilidade nos processos de alfabetização inicial.

De outro lado, investir com ousadia para ampliar as fronteiras de qualidade, preparando o estudante como um ser integral.

A ampliação da jornada escolar pode até ser uma condição de apoio, no entanto o conceito de educação integral que se coloca está relacionado **menos ao aspecto de tempo e mais ao propósito das atividades de ensino e aprendizagem**. Estamos falando de ampliar as oportunidades e situações de aprendizagem significativas e conectadas com a vida do estudante.

Estão envolvidas aí diversas esferas da educação: a organização das áreas de conhecimento, o uso intencional de metodologias e **práticas de ensino e aprendizagem como pontes para o desenvolvimento integral do estudante**, a formação de professores, bem como práticas e **instrumentos de gestão** adotados pelas equipes e redes de ensino, que vão desde o ensino e a aprendizagem, até a rotina das escolas e da secretaria de educação.

Para isso, é importante **reunir os conhecimentos que já vêm sendo produzidos** — tanto pelos próprios professores nas escolas, quanto por pesquisadores de diversas áreas da Ciência — para construir propostas de educação integral com base em evidências. **Antes de desenhar uma iniciativa, é preciso identificar quais as novidades** sobre os mecanismos envolvidos na aprendizagem, os fatores que mais contribuem para o desenvolvimento intencional de competências e as práticas mais eficazes, bem como o impacto da educação integral sobre o desenvolvimento humano ao longo de toda a vida.

A Sociedade Brasileira de Computação, que assina artigo na página 62, destaca que “a compreensão e domínio dos conceitos fundamentais da Computação permitirá que estudantes entendam de forma mais completa o mundo e tenham, conseqüentemente, maior autonomia, flexibilidade, resiliência, pró-atividade e criatividade”. Bonnie Nardi, professora de Ciências da Computação na Universidade da Califórnia, por sua vez, defende que é importante que os usuários finais de tecnologia saibam programar “para que as muitas decisões com as quais a sociedade democrática se depara sobre o uso de computadores, incluindo questões complexas de privacidade, liberdade de expressão e liberdades civis, possam ser abordadas por cidadãos comuns partindo de uma perspectiva de maior conhecimento”.

Letrados em programação e pensamento computacional, portanto, crianças e jovens poderão ser não apenas usuários, mas também criadores de tecnologias capazes de atender às novas demandas do mundo contemporâneo e transformar realidades. A partir dessa visão, o tema vem sendo discutido entre estudiosos de áreas diversas, como Computação, Pedagogia e Letras, e inspirado iniciativas de letramento em programação e pensamento computacional na educação básica. Entre essas experiências está o Letramento em Programação, programa desenvolvido pelo Instituto Ayrton Senna e parceiros, que visa ampliar a inserção de estudantes no universo digital.

O Letramento em Programação busca promover o letramento em linguagens de programação voltadas ao desenvolvimento do pensamento computacional em estudantes de escolas públicas do ensino fundamental. De 2015, ano da sua criação, até 2018, a proposta já chegou a milhares de estudantes, em 19 municípios de 4 estados brasileiros, por meio da formação de centenas de educadores que puderam expandir suas competências e impactar ainda mais escolas. Para além do desenvolvimento de habilidades computacionais e outras competências necessárias para a vida no século 21, a proposta vem aumentando o engajamento dos alunos com a sua educação, contribuindo para a sua aprendizagem em todas as áreas.

Por meio da formação de educadores das redes públicas, a proposta amplia a inserção dos alunos no universo digital utilizando

ferramentas de programação computacional gratuitas, que permitem a autoria de narrativas em plataformas digitais com base em conteúdos temáticos escolhidos pela escola e pelos estudantes, como cyberbullying, sonhos ou projeto de vida. São criados também jogos baseados em conteúdos didáticos em parceria com os professores, que podem ser utilizados pela escola em outras turmas, além de aplicativos de smartphone que resolvem problemas reais da escola ou da família dos estudantes. Também são desenvolvidos sistemas inteligentes de robótica educacional, que podem automatizar tarefas simples do cotidiano, como regar plantas quando a umidade da terra está baixa, por exemplo, trabalhando então conteúdos de matemática, física, química e geografia de forma interdisciplinar.

Reunindo artigos científicos sobre o tema, selecionados a partir de um edital aberto ao público, e também de experiências reais do projeto de Letramento em Programação, a presente obra tem por objetivo disseminar conhecimento sobre o pensamento computacional como estratégia para uma educação integral e transformadora. O livro também apresenta relatos de experiências inovadoras do Letramento em Programação em quatro estados onde este está em operação.

SEÇÃO 1

ARTIGOS DE REFERÊNCIA SOBRE O TEMA

Nesta seção serão apresentados dois artigos convidados para esta edição do livro, com o objetivo de trazer referenciais sobre a programação de computadores e pensamento computacional como meio para promoção da Educação Integral. Por apresentarem ideias e iniciativas relevantes a partir de instituições e pesquisadores de referência, esperamos que sirvam de base para o desenvolvimento e/ou revisão de iniciativas relacionadas ao tema.

Iniciaremos apresentando o artigo “Entendendo a programação de computadores como um letramento”, de Annette Vee – University of Pittsburgh –, que utiliza dados históricos e sociais de estudos de letramento para enquadrar a programação de computadores como “letramento computacional”, e explica que a programação e a escrita seguiram trajetórias históricas semelhantes enquanto tecnologias materiais, e estão interligados em ambientes de redação contemporânea.

Em seguida, será apresentado o artigo Diretrizes de Ensino de Computação na Educação Básica, de autoria de Leila Ribeiro, Lisandro Zambenedetti Granville, Dalton Serey e Simone André da Costa Cavalheiro, que apresentam as diretrizes para o ensino de Ciência da Computação na Educação Básica definidas pela Sociedade Brasileira de Computação (SBC).

CAPÍTULO 1

ENTENDENDO A PROGRAMAÇÃO DE COMPUTADORES COMO UM LETRAMENTO (TRADUZIDO)

Annette Vee – University of Pittsburgh¹

¹ Artigo original em: <http://licsjournal.org/OJS/index.php/LiCS/article/view/24/26>.

“Comparamos a capacidade das massas de ler e escrever software com o letramento em massa e prevemos mudanças igualmente difundidas para a sociedade. Atualmente, os hardwares são suficientemente rápidos e acessíveis para possibilitar a educação computacional em massa: a próxima grande mudança irá acontecer quando a maioria dos usuários de computador tiverem o conhecimento e o poder de criar e modificar softwares.”

- Guido van Rossum, extraído de um pedido de subvenção à *Defense Advanced Research Projects Agency* (Darpa) de 1999 para apoiar o ensino de programação de computadores.

Em seu pedido de subvenção à Darpa, Guido van Rossum (1999), o desenvolvedor da linguagem de programação Python,² quase que certamente esperava obter acesso a associações culturais positivas de letramento, a fim de assegurar o financiamento para seu projeto. Ao mesmo tempo em que invocar o “letramento” é retoricamente oportunista por ser uma causa para a obtenção de financiamentos, diz Cynthia Selfe, a comparação entre programação e letramento tem sido tão frequentemente ecoada que é mais do que apenas uma expressão retórica. Esse paralelo entre a programação e o letramento teve início imediatamente após a invenção dos computadores programáveis. Desde a década de 1960, os entusiastas da Computação vêm empregando o conceito de “letramento” para ressaltar a importância, flexibilidade e poder da escrita para os computadores. O cientista da computação Alan Perlis alegou, em

² Python é uma linguagem popular desenvolvida para ser amplamente acessível e muito utilizada na educação.

1961, que todos os estudantes deveriam aprender programação, da mesma forma que aprendem a escrever, nos cursos de escrita de primeiro ano acadêmico. Na Dartmouth University, na década de 60, os matemáticos John Kemeny e Thomas Kurtz conceberam a linguagem de programação Basic para alunos e não especialistas. Mais tarde, Kemeny (1983, p. 216) escreveu:

Um dia, o letramento em computadores (“computer literacy”, em inglês) será uma condição para a obtenção de emprego, possivelmente para sobrevivência, pois os iletrados em computadores serão cortados da maioria das fontes de informação.

O paralelo entre a programação e o letramento conseguiu, finalmente, um espaço nos comentários populares: Douglas Rushkoff (2010, s/p) diz que aprender programação provê às pessoas “acesso ao painel de controle da civilização” (1), e Marc Prensky (2008, s/p) defende que,

[...] conforme a programação se torna cada vez mais importante, ela vai deixando o quarto dos fundos e se tornando uma competência e atributo fundamentais das nossas classes intelectuais e sociais mais altas, da mesma forma que a leitura e a escrita o foram no passado.

Code.Org, uma entidade sem fins lucrativos iniciada em 2013 e apoiada por Mark Zuckerberg e Bill Gates, apresenta em seu site uma série de citações de educadores, tecnólogos e personalidades públicas, alegando que aprender a codificar é uma questão de “direitos civis”, o “4º letramento” e uma forma de “controlar seu destino, ajudar sua família, sua comunidade e seu país”.

A promoção da programação de computadores como um tipo de escrita adequado para as massas está presente em muito mais lugares do que aqueles aqui relacionados. Porém, infelizmente, quando o “letramento” tem ligação com a programação, essa ligação

se dá, com frequência, de formas não sofisticadas: letramento limitado à leitura e escrita de textos; letramento separado do contexto histórico ou social; letramento como uma forma não mitigada de progresso. No entanto, apesar desses usos irrisórios do conceito de letramento, defendo que esses especialistas em informática têm algo a mais em mente. O que essa ligação persistente da programação à escrita significa para os especialistas em letramento?

A programação de computadores tem muito em comum com o letramento textual – trajetória histórica, formação social, possibilidades de comunicação e conexões com o discurso cívico. Neste artigo, defendo que a associação de “letramento” em referência à programação de computadores não é apenas adequada por causa desses paralelos, mas que nossas definições de letramento devem mudar para absorver essa nova forma de escrita digital. Ainda veremos se a programação de computadores será ou não um letramento em massa. No entanto, assim como o código e os computadores se tornaram elementos essenciais na nossa vida diária, a programação certamente se tornou um poderoso modo de comunicação escrita. Os estudos de letramento podem nos ajudar a melhor compreender a dinâmica social, técnica e cultural da escrita dessa importante tecnologia.

Minha abordagem à ligação entre a programação e o letramento vai além desses breves exercícios comparativos e utiliza estes dados históricos e sociais de estudos de letramento:

1. Historicamente, o letramento ganhou importância quando o texto ganhou importância para a governança (Clanchy, 1993) e o letramento passou a ser uma infraestrutura crucial quando a vida passou a depender deste (Gilmore, 1989);
2. Letramento não se trata apenas de processos técnicos de leitura e escrita: ele é moldado por fatores sociais e ideologias (Street, 1984; diSessa, 2000);
3. Os tipos de identidades letradas disponíveis para as pessoas podem dar forma à maneira como elas aprendem o letramento (Heath, 1983; Purcell-Gates, 1997; Banks, 2006).

Tais princípios de estudos de letramento contemporâneo fornecem perspectivas úteis sobre as formas como a programação de computadores se tornou essencial para as nossas práticas de comunicação ao longo dos últimos 60 anos, bem como o que isso pode significar para a escrita do século XXI.

Em primeiro lugar, defino e analiso reclamações sobre “letramento computacional” (“*computational literacy*”, em inglês) e delineio algumas características destacadas de código, incluindo como ele se tornou uma espécie de escrita. Em seguida, lanço mão dos precedentes da história do letramento, a fim de nos ajudar a considerar o futuro do letramento conforme este se intersecta com a escrita de código. Após isso, forneço uma visão geral de alguns dos contextos sociais da programação de computadores que, de certa forma, se assemelham aos contextos sociais de letramento. Por fim, defendo que os educadores em letramento podem querer ajudar a moldar os valores e abordagens à programação devido à principal função da programação em nossos ambientes de escrita contemporânea. Além de produtivos em contextos profissionais, os valores adotados nas ciências da computação são demasiadamente limitados para um futuro no qual a programação pode se tornar uma prática especializada generalizada e não uma prática especializada – um *letramento*. Com base nessas informações, afirmo que os estudiosos do letramento devem cultivar uma compreensão mais profunda da complexa relação entre letramento computacional e textual.

LETRAMENTO COMPUTACIONAL

Os argumentos sobre o que devemos considerar “letramento” têm se proliferado ao longo das duas últimas décadas. Além do letramento de leitura e escrita de textos, os estudiosos propuseram o letramento visual (Kress & van Leeuwen, 2006), letramento gráfico (Cope & Kalantzis, 2000), letramento quantitativo (Wolfe, 2010) e letramento em videogames (Squire, 2008), entre muitos outros tipos de letramento. As estruturas oferecem ajuda aos professores na gestão

da demanda pela integração desses novos “letramentos” em aulas de escrita (Selber, 2004; Cope & Kalantzis, 2000; DeVoss, McKee, & Selfe, 2009). Esse conjunto de novos lettramentos e abordagens pedagógicas sugere o poder do “letramento” como um termo descritivo que implica a urgência, bem como indica a crescente complexidade das representações e comunicações contemporâneas de informações. Temendo uma diluição do poder explicativo do lettramento, Anne Frances Wysocki e Johndan Johnson-Eilola (1999) nos advertiram contra o uso do termo para descrever todo e qualquer sistema de competências. Por conseguinte, serei cautelosa no que tange à utilização excessiva do termo *letramento* e sobre inclusão de outra competência considerada essencial para a redação do século XXI. Sendo assim, o que é lettramento, exatamente? E por que ele pode ser útil para expandir seu aparato conceitual a fim de descrever a programação de computadores?

Definição de “letramento computacional”

Defino “letramento” como uma facilidade humana com uma tecnologia simbólica e infraestrutural, como um sistema de escrita de texto, que pode ser utilizado para fins criativos, comunicativos e retóricos. O lettramento permite que as pessoas representem suas ideias em textos que fogem aos contextos imediatos e interpessoais (escrever), além de possibilitar a interpretação de textos produzidos por outras pessoas (ler).

A diferença crucial entre o lettramento e um sistema de competências comunicativas dependentes de tecnologia – ao que Andrea diSessa (2000) denominou “inteligência material” – está no posicionamento das tecnologias utilizadas por tais competências: as tecnologias que, subjacentes aos lettramentos, são mais vitais do que aquelas que sustentam inteligências de material (5). Ao mesmo tempo em que as pessoas se beneficiam das inteligências materiais, elas também necessitam de

letramentos para negociar seu mundo. Para que uma inteligência material se torne um letramento, primeiro, é preciso que um componente material ou uma tecnologia se torne essencial, ou infraestrutural para as práticas de comunicação de uma sociedade. Em seguida, a capacidade de interpretar ou compor com essa tecnologia deve se tornar essencial e difundida, o que depende, essencialmente, da facilidade de uso da tecnologia (diSessa, 2000). No caso do letramento textual, o timing desses dois eventos irá variar consideravelmente de acordo com a sociedade em questão. Na Grã-Bretanha, por exemplo, os textos escritos se tornaram essenciais para a sociedade séculos antes da capacidade de lê-los e escrevê-los. E, em sociedades mais isoladas, a capacidade de interpretar textos pode ainda não ser um letramento por não ser difundida ou porque os textos não são importantes. Em extensão ao esquema de diSessa (2000), proponho que: determinar se um sistema de competências é ou não um letramento depende do seu contexto social. Uma pessoa pode ser competente no uso de tecnologias específicas de comunicação, mas o letramento aproveita tecnologias infraestruturais e é necessário na vida quotidiana. Embora muitas inteligências materiais sejam alcunhadas de “letramento”, tal definição de letramento restringe seu campo e faz com que o termo retenha seu potencial.

Assim como o letramento textual, a programação de computadores também é um código que permite que as pessoas representem e interpretem ideias à distância. Em grande parte do mundo, o código é atualmente infraestrutural. Estratificado sobre e sob a tecnologia da escrita, atualmente, o código de programação estrutura em grande parte nossas comunicações contemporâneas, o que inclui processamento de texto, e-mail, a *World Wide Web*, as redes sociais, a produção de vídeo digital e a tecnologia de telefonia móvel. Nossa situação profissional, prontuários médicos e status de cidadania – quando exclusivamente registrados em texto – são catalogados em bases de dados de códigos de programação. Contudo, enquanto a tecnologia de código é atualmente infraestrutural à nossa sociedade, a capacidade de ler e escrever código ainda não é difundida. Por essa definição, a programação de computadores é uma *inteligência material* e ainda não é *letramento*. No entanto, conforme o código se torna mais infraestrutural e à medida que mais pessoas aprendem a

escrevê-lo, a programação de computadores está se assemelhando cada vez mais ao letramento. Conforme descrito em mais detalhe a seguir, a programação está deixando o domínio exclusivo das ciências da computação e tornando-se cada vez mais essencial para profissões como jornalismo, biologia, design e, até mesmo, por meio das humanidades digitais, o estudo de literatura e história.

A forma como a programação está deixando gradativamente o domínio especializado das ciências da computação e migrando para outros campos sugere que ela está se tornando um letramento, ou seja, uma capacidade generalizada de compor e interpretar textos simbólicos e comunicativos em um meio infraestrutural. Os estudiosos das ciências da computação e da educação já utilizaram vários termos para descrever a forma de letramento que a programação pode representar, incluindo “letramento procedimental” (Bogost, 2007; Mateas, 2005), “letramento computacional” (diSessa, 2000) e “pensamento computacional” (Wing, 2006). No campo da retórica e da redação, os termos “letramento fonte” (Stolley, 2012) e “processualidade” (Vee, 2013) também têm sido utilizados. Aqui, utilizo o termo “letramento computacional”, por fazer uma ligação entre o aparato teórico do letramento e o cálculo que é essencial para a programação de computadores; contudo discordo desses estudiosos de várias maneiras significativas.

O “letramento procedimental”, por Ian Bogost (2007, p. 245), “implica a capacidade de reconfigurar os conceitos e regras com o intuito de entender processos, não apenas no computador, mas de forma mais geral”. Nessa definição, Bogost equipara o letramento mais à leitura do que à escrita – “entender[dimento]” de processos, e não os representar. Ele atribui a “autoria[de] argumentos por meio de processos” ao conceito de “retórica procedimental” (Bogost, 2007, p. 29). Para Bogost (2007, p. 257), entender os procedimentos dos artefatos digitais ou recombinar blocos de significado também pode ser letramento procedimental, e não apenas a aprendizagem de programação. Discordo de Bogost no que diz respeito ao meu maior interesse na escrita como sendo essencial para as práticas de letramento, bem como no meu enfoque em contextos sociais e de programação em si. Embora concorde que os artefatos digitais, como jogos, podem oferecer uma janela sobre os processos que garantem o software,

acredito que esse afastamento da programação escamoteia a poderosa dinâmica social e histórica da escrita de código.

Em outra discussão sobre “letramento procedimental”, o cientista Michael Mateas (2005) foca nos novos profissionais da mídia e observa a escrita de código com mais especificidade. No entanto seu tratamento de “letramento” é breve e deixa margem para maior exploração. Jeannette Wing (2006, p. 33) enfoca seu campo da ciências da computação (CC) ao invés da programação propriamente dita, argumentando que “pensamento computacional... é uma competência fundamental para todos e não apenas para cientistas da computação”. Ela explicitamente relaciona seu conceito de “pensamento computacional” à leitura, escrita e aritmética. Discordo de Wing por acreditar que a CC, apesar de importante, não passa de um guia para se pensar sobre essa nova forma de letramento em potencial. Conforme a programação ruma para além da CC, devemos ampliar o aparato conceitual utilizado para compreender suas funções no mundo.

O modelo de “pilares” sociais, cognitivos e materiais do estudioso da educação Andrea diSessa (2000), que apoia o letramento, é compatível com a minha definição de letramento computacional. O termo e conceito de “letramento computacional” reconhece o computador como a base material do letramento, mas também rompe com o termo “letramento em computadores” baseado em competências, que limitou a teoria educacional nas décadas de 1980 e 1990. Computação é, evidentemente, a principal função do computador. Porém, à medida que se torna mais profundamente integrada a dispositivos digitais, a programação está se diversificando para além daquilo que poderíamos tradicionalmente considerar um “computador”. Esse letramento vai mudar conforme a informática muda, assim como o letramento textual mudou com as possibilidades de novas tecnologias de inscrição. Por isso, precisamos de um conceito de programação como letramento que o abstraia para longe das suas atuais tecnologias. Sou a favor do termo “letramento computacional” de diSessa (2000) nesta discussão, pois ele aponta para os mecanismos subjacentes do letramento da programação de computadores – computação –, bem como aponta para além de qualquer instrumento específico.

Influenciada por Bogost (2007), Mateas (2005), Wing (2006) e diSessa (2000), defino “letramento computacional” como sendo a

constelação de competências que decompõe um processo complexo em pequenos procedimentos e, em seguida, expressa – ou “escreve” – esses procedimentos com o uso da tecnologia de códigos que podem ser “lidos” por uma entidade não humana, como um computador. Para escrever códigos, uma pessoa deve ser capaz de expressar um processo em termos e procedimentos que podem ser avaliados recorrendo a regras explícitas. Para ler código, uma pessoa deve ser capaz de traduzir as orientações extremamente explícitas em um modelo de trabalho daquilo que o computador está fazendo. Meu uso do termo “letramento” aqui é estratégico; por tal definição de “letramento”, essa capacidade ainda é apenas uma “inteligência material”. No entanto o “letramento” sugere a função que essa competência irá assumir no futuro, evoca os frequentes paralelos feitos entre a programação e a escrita, e possibilita acesso a teorias de letramento.

O letramento computacional tem base no letramento textual, pois implica escrita e leitura textuais, mas também é bastante distinto do letramento textual. A programação requer a construção de estruturas a partir de componentes explicitamente definidos. Como argumenta Wittgenstein (1997), a linguagem humana funciona de forma diferente: por meio de utilização e intercâmbio e não de definições explícitas. Essa propriedade de linguagem em uso, como sabemos, facilita a literatura e a comunicação humana. Mas também faz com que a linguagem fique suscetível a erros: como Austin (1962) nos lembra, um leitor ou ouvinte pode ter pouca informação ou pode não querer ser persuadido, o que torna uma ação no discurso “inadequada”. A explicitação necessária na programação é uma fonte de crítica dos estudiosos em ciências humanas, pois força definições distintas – por exemplo, Haefner (1999). Mas essa característica do código também permite a construção de procedimentos complexos e encadeados com a confiança de que o computador irá interpretá-los conforme são escritos. O código pode aumentar e realizar a mesma operação milhões de vezes em sequência – uma perfeita possibilidade de perlocutório que é impossível em linguagem humana. Por essas razões, o letramento computacional não é simplesmente uma *prática* de letramento, um subconjunto de letramento textual. Ao invés disso, é um letramento (potencial) próprio, com complexa relação com o letramento textual.

A evolução da programação da engenharia para a escrita

A programação tem uma relação complexa com a escrita; trata-se de escrita, porém sua conexão com a tecnologia de código e computadores³ também a distingue da escrita textual. Ao mesmo tempo, o sistema de escrita de código distingue os computadores de outras tecnologias infraestruturais, como os automóveis⁴. Contudo, nem sempre essa fusão entre texto e tecnologia foi o que aconteceu com computadores; os primeiros computadores mecânicos e elétricos contavam com a engenharia e não a escrita para programá-los. Segue um exemplo: o Mark I de Harvard, concluído em 1944, foi programado por circuitos de computação ou pela conexão física de fios a tubos de vácuo. Cada cálculo novo exigia reconectar a máquina, o que, basicamente, tornava o computador uma máquina própria para cada nova situação. Em 1945, com o desenvolvimento do “conceito de programa armazenado”⁵, o programa do computador podia ser armazenado na memória da mesma maneira que seus dados eram armazenados. Embora de aparência simples, esse design foi revelador: ele trouxe o conceito de “programação” da engenharia física para a representação simbólica. A programação passou a ser uma manipulação de *código*, ou seja, um texto simbólico que fez parte de um sistema de escrita. Dessa forma, assim como a engenharia, os computadores se tornaram tecnologias de escrita.

Nos anos seguintes, o controle do computador por meio de código continuou a se distanciar da materialidade do dispositivo, tendendo para a abstração dos processos que o controlam (Graham). Para ilustrar isso, cada nova revisão do popular computador PDP da

³ Quando escrevo “computadores”, me refiro à classe geral de máquinas que podem executar cálculos, e não apenas *mainframes*, computadores portáteis etc.

⁴ A maioria dos automóveis modernos contém computadores e, portanto, com base em alguns aspectos, separei, artificialmente, essas duas tecnologias. Porém o carro parece ser a tecnologia infraestrutural mais comumente invocada para refutar a importância singular dos computadores, e, portanto, escolho trabalhar com essa tecnologia.

⁵ Esta é por vezes chamada de “Arquitetura de von Neumann”, em homenagem a John von Neumann, um membro da equipe do Eniac da Pennsylvania, e autor nomeado da inovadora “Primeira Versão de um Relatório sobre o EDVAC”. Como a origem desse projeto era colaborativa e complicada, não me refiro a ela, aqui, como “Arquitetura de von Neumann”

Digital Equipment Corporation nos anos 60 exigia uma nova linguagem de programação, pois o hardware havia mudado. No entanto, até a década de 90, a “máquina virtual” da linguagem de programação Java ofereceu um ambiente de programação efetivamente independente de plataforma. Ao longo dos últimos 60 anos, muitos desenvolvedores de linguagens de programação tentaram criar mais linguagens acessíveis ao escritor, que aumentavam o valor semântico do código, isentando os escritores da necessidade de conhecer detalhes do hardware do computador. Esse processo incluía algumas mudanças importantes no design das linguagens de programação, como: uso de palavras em vez de números; gerenciamento automático de memória; organização de programas estruturados; comentários do código, e o desenvolvimento de ambientes de programação para aprimorar a legibilidade do código. Com o desenvolvimento da sintaxe dos códigos de programação até se tornar semelhante à linguagem humana (especialmente o inglês), os requisitos de expressões precisas na programação foram alterados, porém não eliminados.

Tais desenvolvimentos de linguagem levaram muitos a crer que a programação logo se tornaria obsoleta, isto é, uma vez que o computador pode responder à linguagem humana natural, não haverá necessidade de escrever código. Já em 1961, Pedro Elias alegou que a formação em linguagens de programação cessaria em breve, pois os

[...] universitários irão se deparar o console com um teclado tão natural e uma linguagem tão natural, que restará pouco, ou nada, para ser ensinado em termos de programação. [Nesse ponto,] deveríamos esperar que ela desaparecesse de todos os currículos, com exceção daqueles de um grupo moderado de especialistas (Perlis, 1964, p. 203).

À primeira vista, a reivindicação de Elias parece ser apoiada por modernas interfaces, tais como o iPad. Milhares de aplicativos, menus e interfaces prometem oferecer o poder da programação àqueles que não sabem escrever códigos. Coletivamente, eles sugerem que estamos simplificando o processo de resolução de problemas em softwares.

O argumento de Elias é talvez o mais convincente contra a ideia de que a programação se tornará um letramento: as interfaces e linguagens de computador irão evoluir e se tornar tão sofisticadas que pouquíssimas pessoas precisarão saber escrever códigos. Mas isso ainda não aconteceu – pelo menos até agora. Apesar de as linguagens de programação terem continuado a se desenvolver desde 1961, elas ainda têm um longo caminho a percorrer para se tornar uma “língua natural”. Linguagens altamente legíveis, como Python, Ruby e Javascript ainda exigem raciocínio lógico e atenção às expressões explícitas de procedimentos. Interfaces e modelos despojados, como o iPad, podem apenas acomodar opções limitadas. Elas são criadas para o consumo e não a produção de softwares. Isso significa que os programadores e designers de software (ou as empresas para as quais trabalham) ainda imperam. Sendo assim, a trajetória histórica do desenvolvimento das linguagens de programação que aqui descrevi sugere que é improvável que a importância essencial da programação diminua com o aumento da sofisticação das linguagens de computador. Na verdade, se a história do letramento textual puder ser tomada como modelo – e, na próxima seção, defendo que realmente é –, o desenvolvimento de linguagens de programação mais acessível pode *aumentar* em vez de diminuir as pressões sobre o letramento computacional. Mais sofisticadas e mais amplamente distribuídas, as tecnologias da escrita, na verdade, parecem ter colocado mais pressão sobre o letramento individual, elevando, gradativamente, o nível de competência exigido até o nível de competência necessário para que alguém possa ser considerado “letrado” (diSessa, 2000; Brandt, 2001). Da mesma forma, conforme uma tecnologia de “inteligência material” se torna cada vez mais fácil de dominar, tal competência ganha cada vez mais importância para o local de trabalho e se integra cada vez mais à vida quotidiana, tornando-se cada vez mais semelhante a um *letramento*. Igualmente, à medida que os computadores se tornaram mais acessíveis e os idiomas mais fáceis de aprender e usar, a programação parece estar se *afastando* do domínio de especialistas – ao contrário daquilo que Elias espera.

HISTÓRIAS DE LETRAMENTO TEXTUAL E COMPUTACIONAL

Conforme sugerido por Guido van Rossum (1999) em seu pedido de subvenção à Darpa (citado anteriormente), é possível compreender algumas das formas como a programação irá funcionar na sociedade referindo-nos à história do letramento. Nesta seção, visitaremos dois períodos de transição na história do letramento: a primeira é *quando os textos se tornaram infraestruturais* para a vida quotidiana das pessoas, e a segunda é *quando teve início o letramento*. Durante a primeira transição, quando visitamos a Inglaterra medieval para fins de observação, os textos se tornaram essenciais para a vida das pessoas, pois ajudavam no desenvolvimento das instituições – burocracia governamental, lei dos contratos escritos e a empresa da publicação – para o ajuste e absorção do crescimento populacional e do aumento da quantidade de informações. Na segunda transição – o longo século XIX no Ocidente –, instituições, como o serviço de correios, leis tributárias escritas, sinalização pública e educação em massa, foram criadas com base no pressuposto de que a maioria dos cidadãos sabia ler e escrever. Apresento, a seguir, uma história comparativa e abrangente dessas transições para ilustrar o que podemos aprender sobre a trajetória do letramento computacional.

Quando os textos se tornaram infraestruturais

É possível notar um padrão histórico semelhante nas formas como as tecnologias de letramento de texto e código têm se espalhado em toda a sociedade: inicialmente surgidas a partir de iniciativas do governo central, elas se expandiram para outras grandes instituições e empresas e, finalmente, se propagaram, reestruturando a vida doméstica. Apesar da longa existência dos textos, entre os séculos XI e XIII, na Inglaterra, os textos se tornaram comuns

no governo, na organização social e no comércio, realizando uma mudança gradual, porém profunda, na vida quotidiana das pessoas. A escrita, que, inicialmente, era uma ferramenta ocasional, evoluiu até se tornar uma prática infraestrutural de comunicação e registro de informações. De acordo com os escritos do historiador Brian Stock (1983, p. 4), durante esse tempo, as “pessoas passaram a viver textos”. Essa transição sobrevalorizou as competências que, antes, eram detidas apenas por escribas e escrivães e aqueles que sabiam ler e escrever passaram a adquirir um status distinto daquele de outros artesãos. De forma semelhante, as iniciativas do governo americano em tecnologia computacional prefiguraram o uso de computadores nas empresas, na educação e na vida doméstica.

Tanto para o texto como para o código, a mudança na infraestrutura social teve início com a luta do governo central para gerenciar um acentuado aumento das informações. Os normandos que invadiram a Inglaterra do século XI lutaram para controlar uma terra vasta e estranha e, como resultado, o novo governante ordenou a realização de um censo – o qual ficou conhecido como “*Domesday Book*”. Apesar de o *Domesday Book* não ter sido abrangente, ele exigiu das autoridades locais a elaboração de um texto escrito em resposta à solicitação da coroa, o que incentivou e forçou a utilização de textos para registrar informações nas províncias (Clanchy, 1993). Da mesma forma como o *Domesday Book* pretendia registrar a população inglesa recentemente conquistada, o censo americano do final do século XVIII ajudou a recrutar soldados e contribuintes dos Estados Unidos. Porém, como já não era possível contar com a memória para registrar a Inglaterra medieval, a escrita e a matemática implementadas por humanos atingiram o ápice na América do final do século XIX. Como os Estados Unidos cresceram em população e os censos cresceram em ambição, uma nova forma de tabulação de informações foi necessária. Herman Hollerith, um engenheiro mecânico e estatístico de um Escritório de Censo, inventou um computador eletrônico analógico prevendo as informações do Censo 1890. Diferentes versões da “máquina de Hollerith” foram usadas até o censo de 1950, quando foi utilizado o primeiro computador digital – o UNIVAC I (Departamento de Censo dos EUA). Mais uma vez, o censo motivou uma tecnologia de letramento mais sofisticada.

Após essas e outras iniciativas centralizadas, o texto e código foram utilizados em outros grandes projetos de gestão de informações. O computador Eniac, criado para produzir tabelas de disparos para os americanos na Segunda Guerra Mundial, foi concluído tarde demais para ajudar nos esforços de guerra; no entanto, em 1950 e 1960, essa pesquisa abriu caminho para a utilização de computadores por grandes indústrias e instituições, tais como companhias aéreas e universidades (Campbell-Kelly & Aspray, 2004). De maneira semelhante, o governo inglês providenciou a adoção da escrita nas províncias por meio de novas leis e políticas. Até o final do século XIII, as leis da terra começaram a priorizar contratos escritos em relação a testemunhas pessoais. Para participar dessa nova sociedade documental, os indivíduos tinham que saber assinar seus nomes ou utilizar selos para indicar sua anuência aos contratos (Clanchy, 1993). A difusão dos textos do governo central para as províncias repercutiu até chegar ao conhecimento dos programadores, que, adquirindo experiência em grandes projetos de software financiados pelo governo, circularam por entre as pequenas indústrias, divulgando, assim, seu conhecimento do código escrito. O historiador da computação Martin Campbell-Kelly (2004) sugere que os projetos do governo central americano eram, em essência, um campo de treinamento para programadores e incubadoras de tecnologia de computação, que logo se tornaram primordiais para áreas como bancos, companhias aéreas e serviços administrativos. Isso traz à mente aquilo que o historiador Michael Clanchy (1993) chama de mudança “da memória para o registro escrito” – nas décadas de 60 e 70, os Estados Unidos experimentaram uma mudança de escrita para o registro *computacional*.

Foi somente na década de 80 que os computadores se tornaram acessíveis o suficiente para permitir que a maioria das pessoas se familiarizassem com eles. Foi então que os computadores atingiram seu ápice, “de modo que as pessoas com competências comuns pudessem usá-los e desejar usá-los” (Campbell-Kelly & Aspray, 2004, p. 231). Essa é a fase final da difusão do texto e do código – sua expansão do governo centralizado e do comércio para a vida doméstica. Antes do século XI, quando a escrita era apenas ocasional e não poderosamente essencial para negócios ou transa-

ções jurídicas, a capacidade de ler e escrever era uma habilidade não muito diferente da capacidade de esculpir em madeira ou trabalhar com cerâmica; o conceito de “letramento” não existia, pois a leitura e a escrita eram competências especializadas, ou “inteligências materiais”, nas palavras de diSessa (2000). Mas, quando a tecnologia da escrita se tornou infraestrutural, ou seja, quando se tornou essencial para instituições como o governo e o comércio, a capacidade de lidar com tal tecnologia já não podia ser relegada para uma classe especializada. Os leigos que viviam na Inglaterra do século XIII se familiarizaram com as formas como os textos podiam gravar ações, fazer promessas e definir seu lugar na sociedade (Clanchy, 1993). Naquele momento, aqueles que não sabiam ler passaram a ser considerados “iletrados”, e o poder passou a migrar para aqueles que sabiam.

Da mesma forma que a tecnologia do texto começou a fazer parte da vida cotidiana na Inglaterra medieval, a vida das pessoas passou a ser circunscrita por códigos de programação. Nas palavras de Brian Stock (1983), passamos a “viver o código”. Muitos centros de poder econômico e comercial estão ligados aos códigos – os fundadores do Google, Microsoft e Facebook são apenas algumas das pessoas que reestruturaram nosso trabalho e vida pessoal por meio da sua capacidade de programar computadores. Em menor escala, porém de forma profundamente coletiva, nossos rudes sistemas de gestão de cursos, aplicativos de telefonia móvel e softwares de produtividade moldam a maneira como atualmente ensinamos, nos comunicamos e, até mesmo, nos entendemos. Por trás de todos esses softwares estão os programadores – pessoas que fazem desse trabalho uma profissão. Desde o início dos softwares, nunca houve programadores o suficiente para satisfazer a necessidade da sociedade. Vivemos uma perpétua “crise de software”, como observado pelo historiador da computação Nathan Ensmenger (2010) e lembrado pelo recente vídeo promocional do site Code.org. **Embora os códigos de programação estejam em toda parte, são poucos – muito poucos – aqueles que sabem ler ou escrever código. E se esses poucos se tornassem muitos? E se não houvesse programadores de forma limitada, mas, ao invés disso, pessoas que sabem programar?**

Quando teve início o letramento

Conforme os computadores e os códigos se tornam mais essenciais para a nossa formação enquanto cidadãos e para a nossa comunicação, educação e práticas comerciais, a programação de computadores está deixando de ser uma competência especializada e se tornando generalizada, ou seja, passando de uma “inteligência material” para um letramento. Podemos ver paralelos a esse momento nas formas como a leitura e a escrita se tornaram essenciais para o emprego e a cidadania nos séculos XIX e XX.

Com início no final do século XVIII, nos Estados Unidos, a capacidade de escrita e, principalmente, a de leitura se tornaram cada vez mais comuns em consequência de campanhas de letramento em massa, da retórica da construção da nova república, da atividade econômica vigorosa e da motivação pessoal. O dramático aumento nos níveis de letramento no século XIX foi atribuído à importância crescente dos textos – jornais que catalogavam eventos locais e globais, almanaques que ofereciam aconselhamento aos agricultores, cartas divulgadas pelo correio e contas que monitoravam dívidas. Os textos se tornaram transmissores essenciais de cultura e conhecimento entre determinados grupos da Vermont rural do século XIX, para citar um exemplo, de tal forma que “a leitura se tornou uma necessidade da vida”, segundo o historiador William Gilmore (1989). A onipresença do texto também afetou a governança democrática à medida que as cédulas impressas e as mudanças no direito contratual exerciam pressão sobre o letramento (Stevens, 1988). Essas mudanças na governança contribuíram para justificar as campanhas de escolarização em massa nos Estados Unidos do século XIX (Soltow & Stevens, 1981) e estruturou retoricamente o trabalho das mães de letrar seus filhos enquanto cidadãos da nova república (Gilmore, 1989, p. 49). Nos séculos XIX e XX, a necessidade de letramento aumentou rapidamente: após tornar-se comum, o letramento passou a ser mais necessário e, portanto, ainda mais prevalente. Segue um comentário do historiador Lawrence Cremin (1982, p. 493) sobre esse período: “o letramento, em um ambiente no qual está em expansão, tende a criar uma demanda por mais letramento”.

Da mesma forma, os códigos de programação estão se tornando uma competência essencial em profissões fora das ciências da computação, graças ao aumento da sua prevalência e da capacidade de escrevê-los. Clay Shirky (2004, s/p) descreve essa pressão geral da programação sobre o emprego como “*downsourcing*”, ou a generalização dessa prática anteriormente especializada: “apesar de toda a atenção estar voltada para a terceirização, também tem ocorrido muito *downsourcing*, o movimento da programação partindo de uma descrição de trabalho para uma competência mais amplamente praticada”. Embora a necessidade do uso de softwares tenha permeado quase todas as listas de descrições de trabalho, a necessidade de programar computadores vem seguindo logo atrás. Atualmente, os cientistas, economistas, estatísticos, produtores de mídia ou jornalistas que entendem um pouco de programação podem simplificar ou enriquecer suas pesquisas e produção.

A pressão exercida pelo letramento computacional sobre a área de jornalismo merece maior detalhamento, pois ilustra algumas das formas mais interessantes em que a escrita está permeada pela programação. Hoje em dia, a redação para o jornalismo online – seja em blogs ou sites de órgãos de imprensa tradicionais – envolve a integração de elementos programáticos, visuais e de áudio. Paralelamente à escrita tradicional, vemos gráficos interativos e exibição de informações em sites de empresas como *New York Times*, *OK Cupid* e *Five-Thirty-Eight*. Tais composições multimodais onipresentes estão abrindo o caminho para uma abordagem de transmissão de notícias baseada em códigos. A indústria do jornalismo, que ansiosamente experimenta e fornece informações sobre a sua própria situação, acompanhou essa mudança de texto alfabético para mídia digital baseada em código que ocorreu na transmissão de informações. Um escritor da revista online *Gawker* descreve a “Ascensão do Programador Jornalista”:

É possível que um blogger profissional típico tenha que desempenhar tarefas que exigem conhecimento funcional de HTML, Photoshop, gravação, edição e captura de vídeo, podcasts e CSS para poder concluir tarefas que costumavam ser pro-

blemas dos outros, se houvesse [...] Codificação é a próxima etapa lógica nesse processo [...] Não é preciso ir muito longe para ver como a programação pode naturalmente se desenvolver a partir da escrita. (Tate, 2010, s/p).

Em outras palavras, agora, as tarefas que antes faziam parte das descrições do trabalho de outras pessoas foram incluídas por “*downsourcing*” na rotina diária de um jornalista típico. As listas de composições da *Gawker* diferem em seus requisitos técnicos (ex.: HTML é uma “linguagem de marcação extensível”, e não uma linguagem de programação completa), mas, de alguma forma, todas exercem pressão sobre as competências computacionais. Respondendo a essa mudança na profissão, as escolas de jornalismo têm focado a formação de uma nova geração de jornalistas que serão escritores de código, bem como de texto. Por exemplo, recentemente, a Columbia University anunciou um novo Programa de Mestrado em Ciências da Computação e Jornalismo que integrará seu programa de jornalismo tradicional à programação de computadores (van Buskirk, 2010), e a Northwestern’s Medill School of Journalism vem, há vários anos, oferecendo bolsas de estudo a alunos de mestrado cujo histórico inclui ciências da computação ou programação (Medill, s/d).

Vários exemplos recentes de letramento computacional utilizado para aplicações cívicas também ilustram como ele vem esbarrando na escrita, bem como nas preocupações tradicionais dos educadores em letramento. Nos “Crisis Camps” criados nas principais cidades do mundo após o terremoto de 2010 no Haiti, equipes de programadores utilizavam dados geográficos disponibilizados em mapas do Google e pela Nasa para escrever uma base de dados do tipo *Craigslist* que fizesse a correspondência entre as doações e as necessidades e ajudassem a localizar pessoas desaparecidas (American Public Media). Lançada em 2009, a organização *Code for America* utiliza o modelo *Teach for America* para inserir programadores em prefeituras locais para ajudar a simplificar alguns de seus processos burocráticos específicos (“About”, *Code for America*). No nível comunitário, Michele Simmons e Jeff Grabill apresentam um estudo de caso do site e da base de dados de um grupo de ação em

cidadania que revelam os perigos dos PCBs em um abastecimento local de água, o que demonstra como os grupos comunitários podem lutar e obter sucesso com uma tecnologia baseada em código para transmitir suas mensagens. Devido à sua essencialidade para a retórica cívica, Simmons e Grabill (2007, p. 441) alegaram que esse tipo de manipulação programática de bases de dados já não pode ser relegado para disciplinas técnicas: “escrever em interfaces complexas de computador, bem como por meio delas, é um letramento necessário à cidadania no século XXI”. A maioria dessas atividades cívicas não exige grandes habilidades em programação, mas ainda utiliza conceitos básicos de construção de base de dados e de computação baseada em código, o que pode ser considerado um letramento computacional básico. Tal conhecimento permite que um escritor saiba quando e onde a programação é melhor integrada, mesmo que ele não construa o programa individualmente.

Nesses espaços cívicos, a programação apoia uma escrita que pode fazer a diferença no mundo. Talvez, por esses motivos, as justificativas para o ensino da programação enquanto competência generalizada são muitas vezes pronunciadas em linhas cívicas, e não nas forças morais e religiosas por trás das campanhas de letramento textual no século XIX. Bonnie Nardi (1993, p. 3-4) defende que é importante para os usuários finais saber programar,

[...] para que as muitas decisões com as quais a sociedade democrática se depara sobre o uso de computadores, incluindo questões complexas de privacidade, liberdade de expressão e liberdades civis, possam ser abordadas por cidadãos comuns partindo de uma perspectiva de maior conhecimento.

Em momentos como os debates do Congresso sobre as leis *antispam* para e-mails ocorridos em meados de 1990 (Graham, 2004) e sobre a Lei de Combate à Pirataria Online (Sopa, *Stop Online Piracy Act*) de 2012, vimos o que acontece quando os agentes públicos dos Estados Unidos não possuem o conhecimento geral defendido por Nardi. Nesses casos, erros de entendimento fundamentais de

programação de computadores obscureceram os termos do debate e quase levaram a leis deletérias ou inexequíveis.

Nessa crescente necessidade dos jornalistas, cidadãos comuns e agentes públicos por entender um pouco de programação, é possível notar um acúmulo de tecnologias de letramento, como descrito por Débora Brandt (2001) no seu estudo etnográfico das práticas de letramento na América do século XX. Um tanto paradoxalmente, o aumento da importância do letramento acompanhou um aumento da *complexidade* do letramento. Os indivíduos entrevistados por Brandt (2001) notaram que seus locais de trabalho passaram a exigir uma comunicação mais sofisticada, um vasto conhecimento jurídico e a capacidade de escrever com computadores. À medida que as novas tecnologias de letramento se tornavam mais acessíveis e prevalentes, elas foram introduzidas nas práticas de comunicação previamente estabelecidas, aumentando, assim, a complexidade das práticas de letramento. Parece que a maior facilidade no uso das tecnologias digitais multiplicou o letramento novamente: agora, a programação é um novo letramento nos complexos ambientes profissionais.

LETRAMENTO COMPUTACIONAL PARTINDO DA PERSPECTIVA DOS ESTUDOS DE LETRAMENTO

A trajetória histórica descrita sugere que o letramento do século XXI é um fenômeno cada vez mais complexo que inclui habilidade tanto com a tecnologia computacional como com a textual. Apesar de os aplicativos e modelos poderem ajudar indivíduos e organizações a buscar seus interesses em softwares, sem ter, necessariamente, que saber programar, as informações específicas e necessidades de comunicação das empresas, bem como de instituições governamentais e organizações sociais, estão levando a uma maior personalização dos softwares. Com frequência, os serviços voltados ao consumidor não são flexíveis o suficiente para absorver as preocupações locais,

como aqueles descritos por Simmons e Grabill (2007). Além disso, deixar que decisões importantes sobre design de software sejam tomadas pela população pequena (e relativamente homogênea) que sabe programar desempodera aqueles que apenas consomem e não produzem softwares.

Enquanto os achados históricos de Harvey Graff (1991) indicam que, independentemente de outros fatores, ser letrado não confere poder às pessoas, aumenta sua renda ou as faz ascender socialmente, não ser letrado pode significar um impedimento em um mundo onde o texto e o letramento são infraestruturais para a vida cotidiana. Da mesma forma, nota-se que as pessoas não *letradas computacionalmente* deverão, cada vez mais, contar com a ajuda alheia para conduzir suas vidas profissionais, cívicas e pessoais. Tanto na computação como no texto, uma pessoa não letrada “tem menos controle sobre seu destino do que uma pessoa letrada”, conforme observado por Edward Stevens (1988, p. 64) sobre a Nova Inglaterra colonial. Com o aumento do uso de códigos nas comunicações, na organização social, nas funções governamentais e no comércio – e conforme o letramento computacional se torna mais infraestrutural –, o equilíbrio de poder está novamente tendendo para aqueles que possuem competências nessa nova tecnologia de letramento.

Essa alteração no equilíbrio de poder deve alertar os educadores socialmente sintonizados para a importância da integração de práticas de letramento computacional aos seus cursos de escrita e de retórica. Esses cursos já estão sobrecarregados com o excesso de ensino dos letramentos que mencionei no início deste artigo. No entanto podemos começar a pensar sobre como nossas aulas de escrita podem incorporar expressão e conhecimento computacional. Uma concepção específica da forma como isso poderia funcionar está fora do escopo deste artigo, porém a introdução de web design e programação em aulas de redação é um bom começo. Em outras palavras, ensinar alguns aspectos do letramento computacional em aulas de redação não significa que os departamentos de inglês deveriam ensinar ciência da computação. Assim como os cientistas da computação muitas vezes enfatizam que a programação é apenas um aspecto de sua disciplina (Wing, 2006; Denning, 2008), podemos pensar na ciência da computação como uma perspectiva

importante, porém incompleta sobre letramento computacional. A seguir, apresento algumas perspectivas sobre o letramento computacional enquanto fenômeno social *fora* da ciência da computação e defendo que os educadores em letramento podem fornecer valiosas perspectivas pedagógicas sobre a programação.

Aspectos sociais do letramento computacional

Enfatizando os fatores sociais do letramento que se intersectam e excedem suas possibilidades tecnológicas, Brian Street (1984, p. 97) escreve:

[...] obviamente, o letramento é mais do que meramente a “tecnologia” em que é manifesto. Nenhum atributo material serve para definir o letramento propriamente dito. O letramento é um processo social no qual são utilizadas determinadas tecnologias socialmente criadas em estruturas institucionais para fins sociais específicos.

O “modelo ideológico” de Street sintetiza os aspectos tecnológicos e sociais do letramento e traz à mente suas complexas interações. Essa lente tecnológica e social dos estudos de letramento pode nos ajudar a compreender as possibilidades materiais do código e dos computadores, bem como as formas como os valores sociais, contextos e comunidades de programação moldam as práticas de letramento computacional e as identidades relacionadas a tais práticas.

Como o computador é um objeto tecnológico e a programação requer explicitação, de uma forma que frequentemente difere da comunicação humana, a programação de computadores é frequentemente retratada como sendo insocial, ou puramente tecnológica.

Como o trabalho na história da tecnologia tem demonstrado, no entanto, os computadores são tecnologias sociais em sua concepção e implantação (Ensmenger, 2010; Campbell-Kelly & Aspray, 2004). As linguagens de programação são escritas por pessoas, e os programadores escrevem código não apenas para computadores, mas também para outros programadores. Embora, *com frequência*, o código seja escrito *principalmente* para a sua função (conforme lido pelo computador) ao invés de seu valor estético (conforme lido por outros programadores), os públicos duplos do código exercem uma tensão nos valores relativos à sua redação. O computador requer expressão precisa, mas os programadores precisam de legibilidade e desejam um código esteticamente agradável. Enfatizando o valor estético do código para os públicos humanos, o influente cientista Donald Knuth (1992, p. ix) concebeu a famosa “programação letrada”, argumentando que a “literatura do gênero ‘programação’ é realizada por máquinas, mas esse não é seu principal objetivo. Os programas de computador que são realmente bonitos, úteis e rentáveis devem poder ser lidos por pessoas”. O conceito de “programação letrada” de Knuth só é possível pois a programação é feita em espaços sociais com públicos humanos. Para entender algumas dessas influências sociais sobre a programação, devemos separá-las das exigências reais e técnicas do computador. Restrições relativas a como controlar o fluxo do programa, como nomear variáveis, qual deve ser a extensão das funções e qual quantidade de códigos deve ser escrita por linha são socialmente estabelecidas para ajudar os programadores a trabalhar em conjunto, especialmente em equipes muito grandes, mas pouco importam para o computador. Em outras palavras, existem formas de organização de código que o computador entende perfeitamente bem, mas que são evitadas por determinados sistemas de valores humanos na programação. Aspectos depreciados do código são, por vezes, descritos com o termo carinhoso “*code smells*” (Atwood, 2012a), o que evidencia a tensão entre os públicos humanos e de computadores do código.⁶

⁶ Outro exemplo de código comumente denegrido é a declaração “*goto*”, famosamente ridicularizada e “considerada nociva” pelo cientista da computação Edsger Dijkstra (1968), pois permite que os programas saltem para fora de sequência ad hoc e, portanto, violem regras de fluxo de programa. Embora poucas linguagens modernas ainda apoiem tecnicamente a capacidade da declaração *goto* de tornar a estrutura formal do programa, ela permanece uma pedra-de-toque nesse conflito de valores.

Podemos pensar sobre a falácia do código certo-ou-errado como sendo semelhante à da deturpação mecanicista do letramento, de que a leitura e a escrita são meramente uma questão de gramática correta e decodificação precisa. De fato, a escrita textual geralmente requer algum nível de aderência a padrões para facilitar sua leitura. Enquanto a concisão, as transições claras e os verbos ativos podem constituir um bom estilo de escrita, em determinados contextos, esses valores são socialmente moldados (Prendergast, 2009). Eles dependem grandemente do gênero, do público e do contexto, e uma descrição da escrita como sendo meramente aderência a padrões ignora os espaços sociais complexos nos quais ela é produzida e interpretada. A programação requer aderência a normas mais explícitas do que a escrita textual, mas também não pode ser reduzida a estas ou removida de seus contextos sociais. Por exemplo, um programador que trabalha por si mesmo em um aplicativo pequeno não precisa atender a comentários, formatação de códigos e nomenclatura de variáveis que sejam “adequados”, exatamente como definido por Strunk e White: com frequência, uma “gramática adequada” é inadequada para contextos não acadêmicos ou criativos. Em outras palavras, os sistemas de valores para códigos podem falhar quando aplicados fora dos contextos sociais nos quais se desenvolveram (Prendergast, 2009).

À medida que a programação ganha cada vez mais relevância para áreas fora da ciência da computação e da engenharia de software, é possível perceber o desdobramento dessa tensão entre os valores para o código escrito nesses contextos tradicionais e os valores para o código escrito fora deles. Nas ciências, em que o código e algoritmos permitiram que os pesquisadores processassem conjuntos de dados enormes e complexos, essa tensão a respeito do que significa “código adequado” é bastante acentuada. Por exemplo, uma recente história da *Scientific American* relatou que o código não está sendo lançado juntamente com o restante dos métodos utilizados em experimentos científicos, em parte porque os cientistas podem ter “vergonha do código ‘feio’ que escrevem para a sua própria pesquisa” (Hsu, 2012, s/p). Uma discussão sobre o artigo publicado no *Hacker News*, um popular fórum online para programadores, sintetiza algumas das principais tensões na apli-

cação de valores de engenharia de software no código científico. Conforme argumentado em um comentário, o contexto para o qual código está escrito tem importância: “Há uma enorme diferença entre o código único descartável produzido por um cientista que esteja tentando testar uma hipótese e um código de produção produzido por um engenheiro para servir na qualidade comercial” (*Hacker News*, 2012). Um código que possa ser bom para um experimento – que contenha, por exemplo, algumas funções excessivamente longas, duplicação ou outros tipos de “*code smells*” –, pode não ser adequado para um software comercial que seja, digamos, composto por uma grande equipe de programadores ou mantido por décadas em diversos sistemas operacionais. Embora os valores dominantes da programação possam denegri-lo, ele pode ser bom apenas para um contexto científico.

A prática de “ofuscar” códigos, tornando-os ilegíveis para humanos apesar de ainda serem processados pelo computador, também destaca valores de diferentes contextos utilizados na programação. Os programadores ofuscam códigos quando desejam lançar um software de trabalho, mas não querem que as pessoas leiam (e, potencialmente, copiem ou modifiquem) seu código. É ainda possível utilizar ofuscação de formas lúdicas, como no “*Underhanded C Contest*”, em que as pessoas escrevem um código deliberadamente enganoso: ele parece (para humanos) executar uma função, mas, na realidade (para o computador), executa outra (Mateas & Montfort, 2005). O *Underhanded C Contest* solicitou que os programadores escrevessem um código que desviasse bagagem ou contasse votos incorretamente, mesmo parecendo verificá-los. Outro contexto próprio para código ofuscado é o das “linguagens estranhas”, cuja função, com frequência, é fazer comentários irônicos sobre o design e a implementação da linguagem. Embora possam tecnicamente ser utilizadas para escrever softwares, essas linguagens são mais destinadas para brincar do que para usar. Por exemplo, a linguagem apropriadamente denominada *Brainfuck* explora a ofuscação de código tirando espaços em branco e usando apenas alguns caracteres simbólicos, ao invés de letras (Mateas & Montfort, 2005). Se pensarmos no código como sendo escrito em contextos sociais e para outros programadores, faz sentido o fato dele poder ser criativo

e, até mesmo, lúdico para o benefício do público humano. O código ofuscado e as linguagens estranhas sugerem que o valor estético da programação varia de acordo com o seu contexto, da mesma forma como para a escrita.

A história e as formas específicas das tecnologias das linguagens de programação dão forma ao valor e usos do letramento computacional, exatamente como Street (1984, p. 96) alega que elas fazem para o letramento textual. De fato, como Mateas e Montfort (2005, s/p) argumentam sobre linguagens estranhas e ofuscadas, seu inerente “jogo refuta a ideia de que a tarefa do programador é automática, tem valor neutro e é desconectada dos significados das palavras no mundo”. Essa conexão entre o código e as “palavras no mundo” sugere que seu respectivo letramento está imbricado no mundo onde os programadores aprendem e praticam seu ofício. Em outras palavras, o letramento computacional engloba não apenas as competências técnicas de leitura e escrita de código, mas é mais bem entendido como conectado aos seus contextos sociais.

Identidades letradas

Como a leitura é um ato interpretativo que utiliza conhecimentos adquiridos em contextos sociais específicos, Street (1984, p. 180) defende que: “a aquisição de letramento é, na verdade, um processo de socialização e não um processo técnico”. No modelo ideológico de letramento de Street, uma pessoa que tenha adquirido letramento em um contexto pode não ser funcionalmente letrada em outro contexto, pois o letramento não pode ser desligado da sua ideologia. De acordo com o reconhecido estudo etnográfico de letramento realizado por Shirley Brice Heath (1983), os aprendizes de letramento veem que utilizar e valorizar o letramento pode ter impacto sobre a maneira como eles a absorvem ou se eles a absorvem. Crianças que crescem em ambientes onde o texto está ausente e o letramento é marginalizado têm poucas formas de assimilar o

letramento em suas vidas, como sugerido por Victoria Purcell-Gates (1997), em sua obra com ciclos de baixo letramento.

O impacto das identidades disponíveis no desenvolvimento de práticas de letramento também parece ser válido para o letramento computacional. Como demonstrado nos exemplos anteriores sobre programação científica e criativa, as identidades que a ciência da computação disponibiliza para programação são demasiadamente limitadas para poder se tornar letramento computacional – uma competência generalizada e não especializada. Problematicamente, grupos historicamente desfavorecidos no domínio do letramento textual também estão em desvantagem no letramento computacional. Por exemplo, de acordo com o relatório de 2011 do Gabinete de Estatística do Trabalho, somente 20,8% dos programadores de computador eram mulheres.⁷ Embora a programação tenha sido inicialmente uma área dominada por mulheres, ela tendeu para a dominação masculina quando se tornou mais poderosa e complexa (Ensmenger, 2010), e resistiu a uma tendência mais geral de aumento da taxa de participação das mulheres anteriormente evidenciada em áreas dominadas por homens, como direito e medicina. Como a programação é uma forma de escrita poderosa e potencialmente generalizável, quem programa e quem é computacionalmente letrado deve ser uma preocupação dos educadores em letramento.

Os estereótipos dos programadores parecem ter sido inseridos na profissão logo no início: Ensmenger (2010, p. 79) observa que a definição do perfil de personalidade foi utilizada nos anos 60 na seleção de programadores do “sexo masculino antissocial, com inclinação para a matemática”. Embora já não seja praticado explicitamente, esse perfil de personalidade ainda influencia a percepção dos programadores como brancos, do sexo masculino e socialmente desajustados (Ensmenger, 2010). Até mesmo publicações recentes de organizações profissionais como a *Association for Computing Machinery* (ACM) apresentam imagens sexistas: uma ilustração que acompanha o artigo de Peter Denning de 2008 sobre as diversas facetas das ciências da computação (CC) mostra cinco homens (aparentemente brancos) que representam a programação, a engenharia, a matemática etc.,

⁷ No desenvolvimento de softwares e aplicativos, 19% eram mulheres, e no desenvolvimento de web, 38,6% eram mulheres.

ao lado de uma mulher (aparentemente branca) que representa um *usuário* de computador. O sexismo apresentado em conferências de tecnologia e startups aceleradas parece, agora, estar agravando o problema, apesar de inúmeros homens e mulheres de tecnologia terem se manifestado contra ele (Raja, 2012). A recente utilização do termo “*programmer*”, associado à cultura de startups apenas parcialmente de brincadeira, sugere um novo tipo de identidade para programadores – como “*bros*”, ou seja, jovens, do sexo masculino, altamente sociais e ousados (Raja, 2012). A chamada “*ascensão do programmer*” sugere que a programação *pode* acomodar um conjunto mais amplo de identidades, mas essas identidades são ainda muito limitadas.

Além dos persistentes e limitados estereótipos dos programadores de computador, confinar a programação em sua profissão pode restringir os estilos e contextos de programação “aceitável” e desmotivar novos alunos. Isso pode ser observado na discussão anterior sobre o código científico “feio”. A combinação entre a programação profissional e a competência mais generalizada da programação é também evidente nessas críticas recentes do movimento “aprenda o código” como a de Jeff Atwood (2012b). Atwood, o cofundador do popular fórum de programação online *Stack Overflow* e influenciador social, alegou que não precisamos de uma nova geração de pessoas que pensam que podem codificar softwares profissionais – pessoas como o Prefeito da Cidade de Nova Iorque, Bloomberg, que, em 2012, prometeu participar da campanha semanal de aprender a programar da *Codecademy*:

Para aqueles que defendem que a programação é uma competência essencial que devíamos estar ensinando às nossas crianças, bem ao lado da leitura, escrita e aritmética: **podem me explicar como Michael Bloomberg seria melhor no seu trabalho diário de administrar a maior cidade dos EUA se ele acordasse um dia como programador de Java?** (“Por favor, não”, ênfase no original)⁸.

⁸ A citação de Java por Atwood (2012b), a linguagem que domina contextos de software profissional, é outra indicação de seus pressupostos de que “o meme ‘todos devem aprender programação’, como ele o chama, se refere a contextos de programação casual e não profissional.

Conforme apontado por muitos dos seguidores de Atwood, seu argumento apresenta a programação como uma ferramenta apenas para a profissão e desconsidera os possíveis benefícios de se conhecer aspectos da programação em outras profissões ou atividades. Em paradigmas como o de Atwood, programar seria problematicamente e necessariamente limitado aos tipos de pessoas já bem-vindas em um contexto profissional.

Em contraste com essas percepções limitadas sobre quem deveria programar, um conceito de letramento computacional nos ensina que, assim como a escrita pode ser útil para aqueles que não são escritores profissionais, a programação pode ser útil e enriquecedora para muitos grupos diferentes de pessoas. Podemos pensar sobre o que Adam Banks (2006) chama de “acesso transformador” ao letramento computacional, ou seja, o acesso que permite que as pessoas “tanto alterem as interfaces daquele sistema e fundamentalmente alterem os códigos que determinam a forma como o sistema funciona” (Banks, 2006, p. 45, ênfase no original). Alterar a “interface” de programação pode implicar um ensino da programação mais difundido, talvez, até mesmo, em nossas aulas de redação.

Contudo mudar a “forma como o sistema funciona” avançaria para além do acesso substantivo à educação e chegando ao exame crítico das ideologias incorporadas à educação. A programação, conforme definida pela ciência da computação ou pela engenharia de software, tende a ecoar as ideologias desses contextos. Isolar a programação dessas ideologias revela que o webmaster, o criador de jogos, o maker, o cientista e cidadão ativista também podem se beneficiar da programação como um meio de atingir seus objetivos. Inúmeras iniciativas recentes em escolas, como Harvey Mudd (Alvarado & Dodds, 2012), sites como Codecademy.com e Code.org, e organizações locais como *Girl Develop It*, objetivam ensinar programação em novos contextos, como uma competência mais generalizada que os cursos de CC normalmente incentivam. Pode-se dizer que estão todos colaborando para um “acesso transformador” (Banks, 2006) à programação. Um grande benefício de um conceito de letramento computacional é que ele estrutura a programação como uma prática de letramento com diversas aplicações e não como uma profissão definida por um conjunto limitado de valores.

CONCLUSÃO

Assim como a escrita foi gradualmente penetrando nas infraestruturas governamentais e sociais do Ocidente, a programação tem se infiltrado em muitos domínios nos quais, antes, a escrita prevalecia. Padrões semelhantes nessas trajetórias sugerem que a programação poderia vir a ser a base de um novo letramento computacional. No entanto, sem importar que rumo tomará no futuro, a programação já é uma inteligência material e uma poderosa forma de escrita. Devido ao papel central dos códigos no governo, educação, negócios e cidadania – pois o código é infraestrutural –, suas práticas de escrita devem ser consideradas por educadores em letramento. Esse é o argumento reacionário pelo letramento computacional – como disse Douglas Rushkoff (2010, s/p), “programe ou seja programado”. Essa é também a lógica por trás de alguns convites para ensinar programação para crianças do ensino fundamental: a finalidade do meu aprendizado da linguagem de programação Logo nos Estados Unidos na década de 1980 era, supostamente, para nos ajudar a vencer os russos, da mesma forma como o treinamento de programadores na década de 1960 tinha sido realizado com sucesso.

Mas um conceito de letramento computacional nos oferece mais do que estratégias de treinamento em tecnologias da Guerra Fria. Ele também nos ajuda a entender como a redação está mudando. **A programação não está substituindo a escrita, mas entrelaçando-se com ela e aumentando-a.** A programação desempenha um papel de apoio à escrita tradicional (incluindo, para este estudo, escrito em um computador com um software de processamento de texto) e facilita novas formas de comunicação, como tweets, textos, postagens no Facebook, e-mails e mensagens instantâneas. Os exemplos mencionados e extraídos do jornalismo, da literatura pertinente e de algumas aplicações cívicas demonstram algumas das mudanças oriundas dessa nova tecnologia de escrita híbrida. Considerar a escrita de código com base em um conceito de letramento computacional nos permite focar as práticas de escrita que respaldam nossos complexos ambientes contemporâneos de redação. Possibilita um envolvimento mais forte com o nosso soft-

ware, pois destaca as pessoas que o escrevem, bem como os padrões históricos que o precedem.

Por fim, entender a programação de computadores como uma forma de letramento computacional leva a uma pedagogia de escrita mais abrangente e inclusiva. É importante aumentar o acesso à programação devido ao seu poder e diversidade de aplicações, o que significa que a programação não pode ser relegada ao domínio exclusivo das ciências da computação. Ela também é importante para expandir os nossos conceitos de escrita e incluir a programação. Imagens e som, juntamente com outros modos de redação, já mudaram a maneira como nos comunicamos e como podemos expressar e processar as informações. Consequentemente, os estudiosos em letramento adicionaram esses modos de escrita ao nosso conceito de letramento e vêm discutindo sobre a forma como incorporá-los em salas de aula de redação. Porém a programação permite *todas* as formas de redação digital. Agora, devemos mudar os nossos modelos de letramento para justificá-los.⁹

NOTAS

² Para obter uma lista mais abrangente, consulte minha bibliografia anotada, disponível em: <http://www.scribd.com/doc/96309313/Computer-Programming-e-Literacy-An-Annotated-Bibliography>.

³ Aqui, estou utilizando a estrutura de Susan Leigh Star (1999, p. 381-2), que descreve a infraestrutura como sendo integrada e transparente, até que se quebre, tenha amplo alcance, seja moldada por padrões e seja difícil de mudar.

⁹ Gostaria de agradecer a Tim Laquintano, Kate Vieira, Débora Brandt, Steve Carr, Don, Bialostosky, Michael Bernard-Donals e aos revisores anônimos pelos relevantes comentários nos rascunhos do presente artigo.

OBRAS CITADAS

"About" (2012). *Code for America*. Code for America Labs, Inc., n.d. Web. 30 Jun.

Alvarado, C., & Dodds, Z. (2012). Women in CS: An Evaluation of Three Promising Practices. *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, 2010. 57-61. Web. *ACM Digital Library*. 30 Jun. 2012.

American Public Media (2010). Devising Aid Programs on Their Laptops. *Marketplace*. 18 Jan. 2010. Web. 31 Jan. 2010.

Atwood, J. (2012a). Code Smells. *Coding Horror Blog*. 18 May 2006. Web. 20 Jun. 2012.

Atwood, J. (2012b). Please Don't Learn to Code. *Coding Horror Blog*. 15 May 2012. Web. 15 May 2012.

Austin, J. L. (1962). *How to Do Things with Words*. Cambridge: Harvard UP. Print.

Banks, A. (2006). *Race, Rhetoric, and Technology*. Mahwah: Lawrence Erlbaum. Print.

Bogost, I. (2007). *Persuasive Games: The Expressive Power of Videogames*. Cambridge: MIT P. Print.

Brandt, D. (2001). *Literacy in American Lives*. Cambridge, UK: Cambridge UP. Print.

Bureau of Labor Statistics, United States Department of Labor (2012). Labor Force Statistics from the Current Population Survey. 1 Mar. 2012. Web. Retrieved 26 Jun. 2012, from <http://www.bls.gov/cps/cpsaat11.htm>.

Campbell-Kelly, M. (2004). *From Airline Reservations to Sonic the Hedgehog: A History of the Software Industry*. Cambridge: MIT P. Print.

Campbell-Kelly, M., & Aspray, W. (2004). *Computer: A History of the Information Machine*. The Sloan Technology Series. 2nd ed. Boulder: Westview P. Print.

Clanchy, M. (1993). *From Memory to Written Record: England 1066 - 1307*. Malden: Blackwell Publishing. Print.

Code.org (s/d). Code.org. Web. Retrieved 7 April 2013, from <http://www.code.org/>.

Codecademy(s/d). Codecademy. Web. Retrieved 7 April 2013, from <http://www.codecademy.com>.

Cope, B., & Kalantzis, M. (Ed.)(2000). *Multiliteracies: Literacy Learning and the Design of Social Futures*. London: Routledge. Print.

Cremin, L. (1982). *American Education: The National Experience, 1783-1876*. New York: Harper and Row. Print.

Denning, P. J. (2008). The Profession of IT: Voices of Computing. *Communications of the ACM* 51.8 (2008): 19-21. ACM Digital Portal. Web. 10 Apr. 2013.

DeVoss, D. N., McKee, H. A., & Selfe, R. (Dickie)(2009). *Technological Ecologies and Sustainability*. Logan: Computers and Composition Digital P. Web. 17 Jul. 2013.

Dijkstra, E. (1968). Go to Statement Considered Harmful. *Communications of the ACM* 11.3 (1968): 147-48. ACM Digital Portal. Web. 23 Apr. 2010.

diSessa, A. (2000). *Changing Minds: Computers, Learning and Literacy*. Cambridge: MIT P. Print.

Ensmenger, N. (2010). *The Computer Boys Take Over: Computers, Programmers, and the Politics of Technical Expertise*. Cambridge: MIT P. Print.

Gilmore, W. J. (1989). *Reading Becomes a Necessity of Life: Material and Cultural Life in Rural New England, 1780-1835*. Knoxville: U of Tennessee P. Print.

Gizmodo.com (2012). Retrieved 13 July 2019, from <https://gizmodo.com/please-dont-learn-to-code-5910497>

Graff, H. (1991). *The Literacy Myth: Cultural Integration and Social Structure in the Nineteenth Century*. 1979. New Brunswick: Transaction Publishers. Print.

Graham, P. (2004). *Hackers & Painters: Big Ideas from the Computer Age*. Sebastopol: O'Reilly. Print.

Hacker News (2012). Secret Computer Code Threatens Science (scientific-american). 13 Apr. 2012. Web. Retrieved 17 Apr. 2012, from <https://news.ycombinator.com/item?id=3844910>.

- Haefner, J. (1999). The Politics of the Code. *Computers and Composition* 16.3 (1999): 325–39. *ScienceDirect*. Web. 19 Aug. 2008.
- Heath, S. B. (1983). *Ways with Words: Language, Life and Work in Communities and Classroom*. New York: Cambridge UP. Print.
- Hsu, J., & Innovation News Daily (2012). Secret Computer Code Threatens Science. *Scientific American, Inc.*, 13 Apr. 2012. Web. 17 Apr. 2012.
- Kemeny, J. (1983). The Case for Computer Literacy. *Daedalus* 112.2 (1983): 211–30. *JStor*. Web. 13 Jan. 2013.
- Knuth, D. (1992). *Literate Programming*. CSLI Lecture Notes. United States: Center for the Study of Language and Information. Print.
- Kress, G., & Leeuwen, T. van. (2006). *Reading Images: The Grammar of Visual Design*. 1996. 2nd ed. London: Routledge. Print.
- Mateas, M. (2005). Procedural Literacy: Educating the New Media Practitioner. *Future of Games, Simulations and Interactive Media in Learning Contexts*. Spec. issue of *On The Horizon* 13.1 (2005): 110–11. Print.
- Mateas, M., & Montfort, N. (2005). A Box, Darkly: Obfuscation, Weird Languages, and Code Aesthetics. *Proceedings of the 6th Digital Arts and Culture Conference, IT University of Copenhagen* (2005): 144–53. NickM.com. Web. 1 Mar. 2009.
- Medill School of Journalism (s/d). *Admissions*. Northwestern U, n.d. Web. 24 Apr. 2010.
- Nardi, B. (1993). *A Small Matter of Programming: Perspectives on End User Computing*. Cambridge: MIT P. Print.
- Perlis, A. (1964). The Computer and the University. *Computers and the World of the Future*. Ed. Martin Greenberger. Cambridge: MIT P. Print.
- Prendergast, C. (2009). The Fighting Style: Reading the Unabomber's Strunk and White. *College English* 71.1 (2009): 10–28. Print.
- Prensky, M. (2008). Programming is the New Literacy. *Edutopia*. Web. Retrieved 23 Apr. 2010, from <http://www.edutopia.org/literacy-computer-programming>.
- Purcell-Gates, V. (1997). *Other People's Words: The Cycle of Low Literacy*. 1995. First paperback ed. Cambridge: Harvard UP. Print.

Raja, T. (2012). 'Gangbang Interviews' and 'Bikini Shots': Silicon Valley's Programmer Problem. *Mother Jones*. (26 Apr. 2012). Web. 18 Jun. 2012.

Rushkoff, D. (2010). *Program or Be Programmed: Ten Commands for a Digital Age*. N.p.: OR Books. Kindle file.

Selber, S. A. (2004). *Multiliteracies for a Digital Age*. Carbondale: Southern Illinois UP. Print. Studies in Writing and Rhetoric Series.

Selfe, C. (1999). *Technology and Literacy in the 21st Century: The Importance of Paying Attention*. Carbondale: Southern Illinois UP. Print. Studies in Writing and Rhetoric Series.

Shirky, C. (2004). Situated Software. *Clay Shirky's Writings about the Internet*. Web. 15 Jan 2010.

Simmons, W. M., & Grabill, J. (2007). Toward a Civic Rhetoric for Technologically and Scientifically Complex Places: Invention, Performance, and Participation. *College Composition and Communication* 58.3 (2007): 419-48. Print.

Soltow, L. & Stevens, E. (1981). *The Rise of Literacy and the Common School in the United States: A Socioeconomic Analysis to 1870*. Chicago: U of Chicago P. Print.

Squire, K. D. (2008). Video Game Literacy: A Literacy of Expertise. *Handbook of Research on New Literacies*. Eds. Julie Coiro, Michele Knobel, Colin Lankshear, and Donald J. Leu. New York: Lawrence Erlbaum. 635-70. Print.

Star, S. L. (1999). The Ethnography of Infrastructure. *American Behavioral Scientist* 43.3 (1999): 377-91. *Sage Journals*. Web. 3 Jul. 2013.

Stevens, E. W., Jr. (1988). *Literacy, Law, and Social Order*. DeKalb: Northern Illinois UP. Print.

Stock, B. (1983). *The Implications of Literacy*. Princeton: Princeton UP. Print.

Stolley, K. (2012). Source Literacy: A Vision of Craft. *Enculturation*. 10 Oct. 2012. Web. Retrieved 11 Apr. 2013, from <http://enculturation.gmu.edu/node/5271>.

Street, B. (1984). *Literacy in Theory and Practice*. Cambridge, UK: Cambridge UP. Print.

Tate, R. (2010). Hack to Hacker: Rise of the Journalist-Programmer. *Gawker: Valleywag*. 14 Jan. 2010. Web. 20 Jan. 2010.

U.S. Census Bureau (2009). "History: Overview," "History: Univac I." 10 Dec. 2009. Web. 5 Jan. 2010.

van Buskirk, E. (2010). Will Columbia-Trained, Code-Savvy Journalists Bridge the Media/Tech Divide? *Wired*. 7 Apr. 2010. Web. 20 Apr. 2010.

van Rossum, G. (1999). *Computer Programming for Everybody* (Revised Proposal). Corporation for National Research Initiatives, Jul 1999. Web. 23 Apr. 2010.

Vee, A. (2013). Ideologies of a New Mass Literacy. *Conference on College Composition and Communication Convention*. The Riviera, Las Vegas. 14 Mar. 2013. Address.

Wing, J. (2006). Computational Thinking. *Communications of the ACM* 49.3 (2006): 33-35. *ACM Digital Library*. Web. 7 Apr. 2007.

Wittgenstein, L. (1997). *Philosophical Investigations*. Trans. G.E.M. Anscombe. 2nd ed. Cambridge: Blackwell, 1997. Print.

Wolfe, J. (2010). Rhetorical Numbers: A Case for Quantitative Writing in the Composition Classroom. *College Composition and Communication* 61.3 (2010): 452-75. Print.

Wysocki, A. F., & Johnson-Eilola, J. (1999). Blinded by the Letter: Why Are We Using Literacy as a Metaphor for Everything Else? *Passions, Pedagogies, and Twenty-First Century Technologies*. Ed. Gail Hawisher and Cynthia Selfe. Logan: Utah State UP. 349-68. Print.

CAPÍTULO 2

DIRETRIZES DE ENSINO DE COMPUTAÇÃO NA EDUCAÇÃO BÁSICA

Leila Ribeiro

Lisandro Zambenedetti Granville

Dalton Serey

Simone André da Costa Cavalheiro

SOCIEDADE BRASILEIRA DE COMPUTAÇÃO

INTRODUÇÃO

Computação é uma ciência: possui fundamentos e princípios, organizando de forma sistemática parte do conhecimento da humanidade. Computação pode ser considerada uma ciência natural (Denning, 2007): já existia muito antes de computadores (máquinas) serem inventados. Por exemplo, em Biologia, informação é codificada de forma digital no DNA (usando quatro símbolos) e processada usando procedimentos e métodos computacionais bem definidos. David Baltimore, ganhador do prêmio Nobel de Biologia em 1975, afirmou em 2001 que os mecanismos celulares são “processos computacionais naturais para ler DNA e modificar/construir células vivas, ou seja, a Biologia é uma ciência da informação” (Baltimore, 2001). Os fluxos de informação da Economia e Administração são processos computacionais. Uma receita de bolo, um roteiro turístico, as instruções de como usar uma cafeteira são processos computacionais. Stephen Wolfram, na sua contribuição *A new kind of science* (Wolfram, 2002; 2017), afirma que a natureza é escrita em uma linguagem computacional. O universo é permeado de processos computacionais.

Computação está em todos os lugares, em tudo que fazemos. A Ciência da Computação explica uma parte (abstrata) do mundo real: os *processos de informação*. Mas Computação também é uma ciência do artificial, porque ela pode ser usada para investigar problemas e construir soluções, gerando processos que não existiam no mundo

real, criando um mundo artificial, virtual, um mundo que é hoje presente e fundamental na vida de grande parte das pessoas (por exemplo, a Internet é um componente do mundo virtual). Alfred Aho (2011) afirma que a Computação explica os processos que ocorrem no mundo por meio de modelos, chamados modelos computacionais, e esses modelos podem ser construídos de diferentes formas, usando diferentes linguagens e descrevendo vários aspectos em níveis de abstração diversos. Mas esses modelos não são apenas descrições abstratas da realidade, a Computação também estuda formas de implementar esses modelos usando máquinas (computadores) para que os processos sejam mais bem compreendidos ou, no caso do mundo virtual, passem a existir. A Computação provê, portanto, não somente explicação, como também ferramentas para transformar o mundo.

Mesmo sendo uma área altamente inovadora e tecnológica, os princípios da Computação são os mesmos há décadas. A compreensão e domínio dos conceitos fundamentais da Computação permitirá que estudantes entendam de forma mais completa o mundo e tenham, conseqüentemente, maior autonomia, flexibilidade, resiliência, proatividade e criatividade.

Como nas outras áreas do conhecimento, é muito importante que o professor saiba quais são as competências e habilidades que podem ser desenvolvidas a partir da Computação, se não o ensino fica sem objetivo. Portanto, o primeiro passo para inserir Computação nas escolas é a definição de diretrizes curriculares para orientar professores e escolas, definindo as competências específicas desenvolvidas pela Computação, e as habilidades e objetos de conhecimento que podem ser trabalhados em cada etapa escolar para atingir essas competências.

A proposta de Diretrizes de Ensino de Computação na Educação Básica foi elaborada por uma comissão de professores especialistas na área. Versões intermediárias foram apresentadas e discutidas com a comunidade em eventos da SBC. Na elaboração, foram considerados os currículos de vários países, como Estados Unidos (CSTA, s/d), Inglaterra (Guerra, Kuhnt, & Blöchliger, 2012), Austrália (ACARA, s/d), Alemanha (GOV.UK, 2013), entre outros (CFE, 2015; OPS, 2016), além de vários projetos-piloto existentes no Brasil (Bordini *et al.*,

2016). O documento completo pode ser encontrado em <http://www.sbc.org.br/educacao/diretoria-de-educacao-basica>.

O restante deste capítulo está organizado da seguinte forma: a seção 1.2 apresenta os eixos nos quais os conhecimentos da área de Computação foram organizados; a seção 1.3 discute como a Computação auxilia no desenvolvimento das competências gerais definidas na Base Nacional Comum Curricular (BNCC, s/d); as diretrizes da SBC são apresentadas na seção 1.4; e a seção 1.5 tece algumas considerações finais.

OS EIXOS DA COMPUTAÇÃO

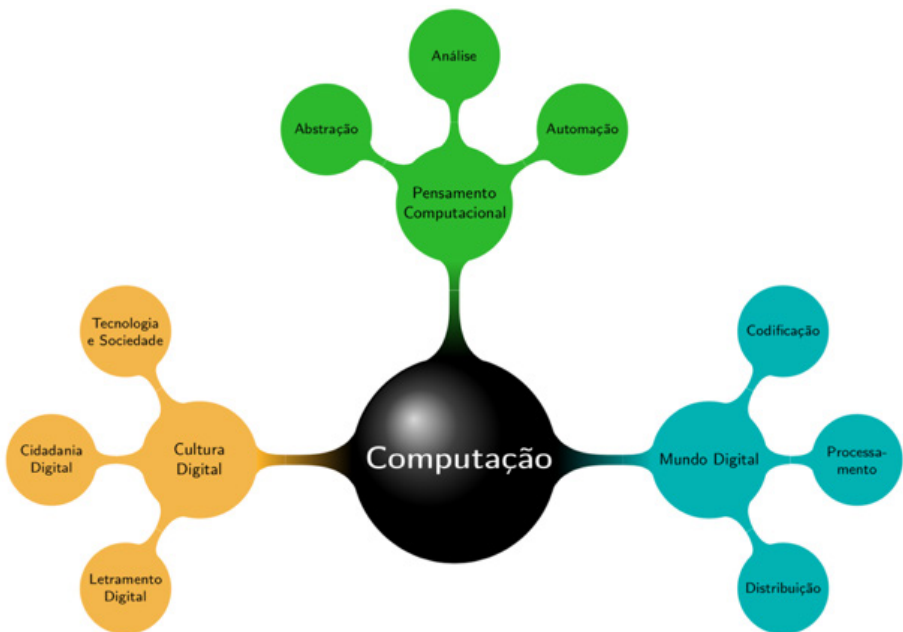


Figura 1.1 – Eixos da Computação

Fonte: Sociedade Brasileira de Computação, 2008, p. 4

Os conhecimentos da área da Computação foram organizados em 3 eixos, mostrados na Figura 1.1. Os eixos estão detalhados nos quadros da Figura 1.2. O **Pensamento Computacional** se refere à capacidade de compreender, definir, modelar, comparar, solucionar, automatizar e analisar problemas (e soluções) de forma metódica e sistemática, por intermédio da construção de algoritmos. O conceito de algoritmo está presente em todas as áreas e está intrinsecamente ligado à resolução de problemas, pois um algoritmo é uma descrição de um processo (que resolve um determinado problema), incluindo os recursos necessários e/ou gerados pela execução do processo. O Pensamento Computacional envolve, portanto, abstrações e técnicas necessárias para a descrição e análise de informações/recursos (dados) e dos passos dos processos, bem como para a automação de soluções usando máquinas (computadores). Assim, o eixo do Pensamento Computacional é fundamentado em três pilares (Wing, 2008; Ribeiro, Foss, & Cavalheiro, 2017): abstração, automação e análise. Por sua vez, a compreensão do **Mundo Digital** envolve três pilares principais, que chamamos de codificação, processamento e distribuição. A codificação diz respeito à representação, no mundo digital, dos mais diferentes tipos de informação que possam nos interessar. A capacidade de processamento dos dados codificados no mundo digital confere extrema agilidade para desempenhar vários processos, assim como habilita vários outros a acontecerem. De forma indissociável nesse contexto está a capacidade de distribuição de informação no mundo digital. Essa capacidade é fator fundamental para tamanho impacto do mundo digital. O terceiro eixo diz respeito à compreensão mais ampla das tecnologias digitais e seu impacto no mundo. A **Cultura Digital** inclui as relações interdisciplinares da Computação com outras áreas do conhecimento, buscando promover a fluência no uso do conhecimento computacional para expressão de soluções e manifestações culturais de forma contextualizada e crítica. Também faz parte da Cultura Digital uma análise dos novos padrões de comportamento e novos questionamentos morais e éticos na sociedade que surgiram em decorrência do Mundo Digital.

CULTURA DIGITAL

Tecnologia e Sociedade

Compreender o impacto e decorrências da revolução digital e dos avanços do mundo digital na humanidade.

Letramento Digital

Utilizar de forma eficiente e crítica ferramentas que auxiliem a obter, analisar, sintetizar e comunicar informações de formatos e com fins diversos.

Cidadania Digital

Analisar de forma crítica questões éticas e morais que surgiram com o mundo digital, valorizando a diversidade e fortalecendo o olhar para a acessibilidade.

PENSAMENTO COMPUTACIONAL

Abstração

RESOLUÇÃO DE PROBLEMAS, REPRESENTAÇÃO DA INFORMAÇÃO E PROCESSOS:

Compreender e utilizar modelos e representações adequadas para organizar informação e descrever processos. Usar diferentes técnicas (decomposição, generalização, transformação, reuso, recursão, refinamento) para construir soluções para problemas, identificando as vantagens do uso destas técnicas e as condições para utilizá-las, aperfeiçoando e articulando saberes escolares.

Automação

PROGRAMAS E DADOS: Identificar problemas de diversas áreas do conhecimento e criar soluções, de forma individual e colaborativa, usando computadores, celulares e/ou outras máquinas processadoras de instruções para resolver todo ou partes do problema. Criar modelos computacionais para simular e fazer predições sobre diferentes fenômenos e processos.

Análise

ANÁLISE QUANTITATIVA E QUALITATIVA DE PROBLEMAS/ SOLUÇÕES:

Analisar criticamente os problemas e soluções para identificar não somente se existem soluções que podem ser automatizadas, mas também ser capaz de avaliar a eficiência e a correção destas soluções e de justificar tanto a adequação das soluções aos requisitos quanto as escolhas frente às diferentes possibilidades de soluções.

MUNDO DIGITAL

Codificação

DADOS - CODIFICAÇÃO DA INFORMAÇÃO:

Entender o conceito de informação e como informações podem ser descritas e armazenadas de diferentes formas. Compreender a importância da informação, e dos problemas e soluções relacionados a segurança de dados. Identificar diferentes formas de armazenar dados, e como representar visualizar e manipular dados massivos (big data).

Processamento

COMPUTADOR - PROCESSAMENTO DA INFORMAÇÃO:

Identificar os componentes básicos de um computador e compreender como a informação é processada por eles. Entender os diferentes níveis de relação entre hardware e software. Compreender os fundamentos da robótica e inteligência artificial.

Distribuição

REDES E INTERNET - DISTRIBUIÇÃO DA INFORMAÇÃO:

Entender como se dá a comunicação entre diferentes dispositivos digitais, como os dados são transmitidos. Compreender a estrutura e funcionamento da internet. Ser capaz de avaliar a confiabilidade de sistemas computacionais e empregar diferentes medidas de segurança digital.

Figura 1.2 – Eixos da Computação

Fonte: dos autores

COMPUTAÇÃO E AS COMPETÊNCIAS GERAIS DA BNCC

Formação em Computação é, hoje, comparável à alfabetização no passado. Pessoas que não têm a formação adequada em Computação compreendem de forma limitada o mundo de hoje (e do futuro), têm menos chances de expressar-se, inserir-se na sociedade, empregar-se, viver com qualidade: utilizar informação em seu benefício. O país que não oferece essa formação aos seus cidadãos terá a desigualdade social acentuada e perderá poder econômico, capacidade de produzir ciência e inovar em todas as áreas do conhecimento, assim como desenvolver suas indústrias.

Computação contribui na formação do jovem do século XXI, pois:

- Permite a **compreensão plena do mundo**, cada vez mais conectado e imerso em tecnologias digitais essencialmente;
- Aumenta a **capacidade de aprendizagem e resolução de problemas** dos alunos, provendo novas formas de expressão e pensamento;
- Serve como ferramenta de **apoio ao aprendizado das demais disciplinas**.

A Figura 1.3 explicita a relevância do ensino de Computação para o desenvolvimento das Competências Gerais da BNCC.



A compreensão do Mundo Digital é essencial para entender o mundo do século XXI.

O Pensamento Computacional desenvolve a capacidade de compreender, definir, modelar, comparar, solucionar, automatizar e analisar problemas (e soluções) de forma metódica e sistemática, através da construção de *algoritmos*.

Tecnologias digitais podem ser usadas para analisar e criar no mundo artístico e cultural.

Computação desenvolve uma maior compreensão do conceito de linguagem e do seu uso, e prove fluência em linguagens computacionais (usadas para representar informações e processos).

Computação prove fluência digital e a habilidade de criar soluções para diversos tipos de problemas (do mundo do trabalho e cotidiano) com o auxílio de computadores.

O domínio do conhecimento sobre o mundo, que é hoje imensamente influenciado pelas tecnologias digitais, e a capacidade de criar e analisar criticamente soluções neste contexto dá liberdade, autonomia e consciência crítica, além de preparar o aluno para o mundo do trabalho.

O Pensamento Computacional desenvolve a habilidade de construir argumentações consistentes e sólidas.

A construção e análise de algoritmos instiga questões sobre como o ser humano pensa e constrói soluções.

O desenvolvimento de soluções algorítmicas é um processo que permite exercitar a cooperação de forma sistemática.

A Computação, através do desenvolvimento do Pensamento Computacional, domínio do Mundo Digital e compreensão da Cultura Digital dá ao aluno condições de agir com consciência e cidadania no mundo do século XXI.

Figura 1.3 – Computação e as Competências Gerais da BNCC

Fonte: Sociedade Brasileira de Computação, 2018, p. 6

DIRETRIZES

Competências Específicas da Computação

Além de auxiliar no desenvolvimento das competências gerais da BNCC, o ensino de Computação desenvolve competências específicas, que são essenciais na formação do cidadão do século XXI. A evolução da sociedade exige que os jovens sejam preparados com as habilidades e competências do século XXI, as quais abrangem novas formas de pensar (integrando pensamento crítico, criatividade, inovação, resolução de problemas, tomada de decisão e metacognição), novas maneiras de trabalhar (incluindo trabalho colaborativo, interação virtual, flexibilidade e adaptabilidade), novos meios de viver no mundo (integrando cidadania digital e responsabilidade social) e novas ferramentas de trabalho (como literacia digital e informacional). A Figura 1.4 sumariza cinco competências específicas da Computação que impulsionam o desenvolvimento de tais habilidades. Para cada competência, é descrito seu objetivo principal e sua respectiva finalidade.

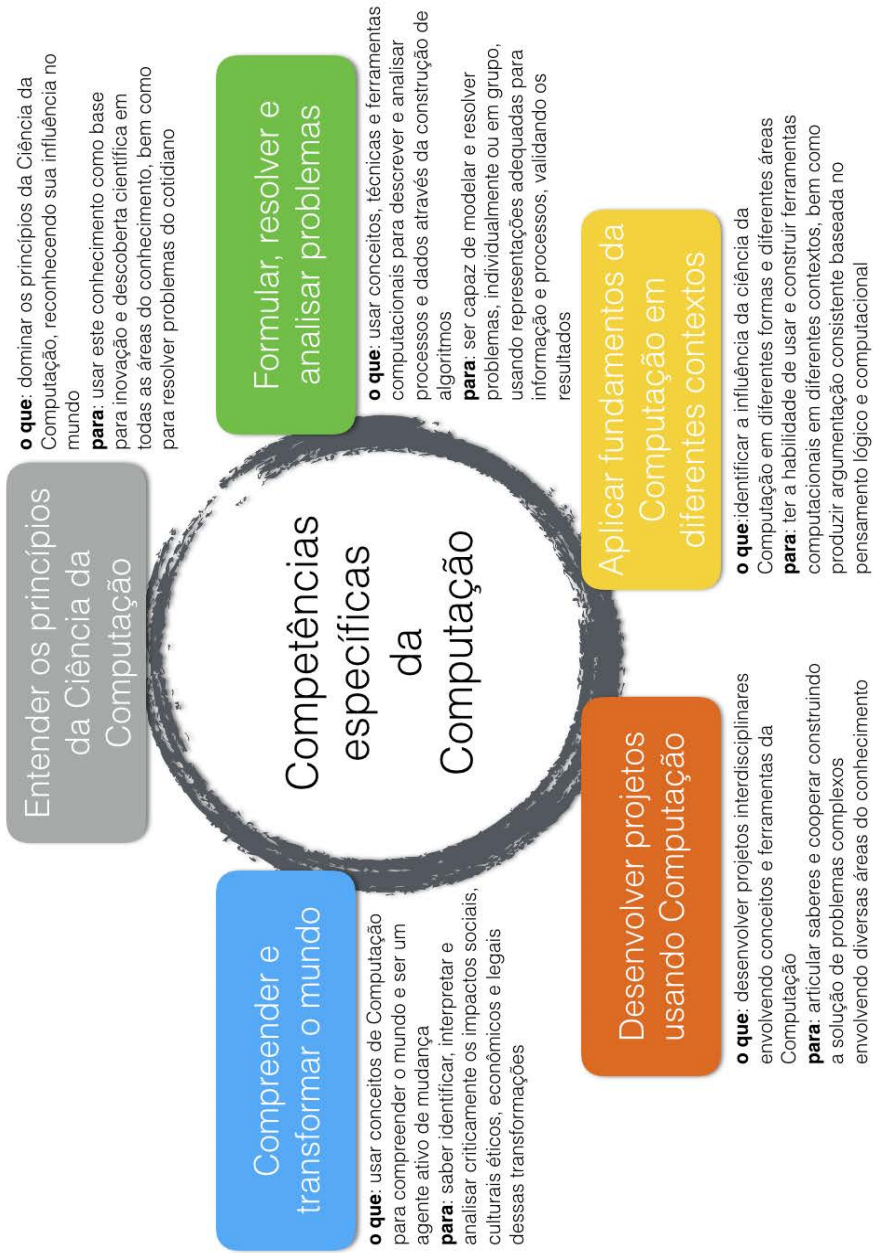


Figura 1.4 – Competências específicas da Computação

Fonte: dos autores

Concepção das diretrizes

A concepção das diretrizes está embasada nos fundamentos da Ciência da Computação. Nessa proposta, a ênfase é dada ao ensino de conceitos de Computação, e não ao uso de tecnologias, pois o domínio dos conceitos desenvolve a capacidade de utilizar e criar tecnologias.

Para a apropriação dos conceitos fundamentais discutidos na seção 1.3, sugere-se que eles sejam introduzidos do concreto ao abstrato. Ou seja, num primeiro momento, os conceitos computacionais devem ser desenvolvidos a partir de situações do cotidiano e materiais manipuláveis para posteriormente introduzir suas respectivas formalizações e abstrações. Dessa forma, permite-se que o estudante, ao trabalhar com materiais concretos, crie modelos mentais que, em etapa posterior, servirão de base para que ele consiga abstrair e formalizar os conhecimentos. Por exemplo, nos Anos Iniciais, pode-se trabalhar o conceito de lista e operações sobre listas utilizando-se baralhos, pilhas de figurinhas, fila de alunos, entre outros. Um segundo passo seria compreender a necessidade de descrever listas usando alguma linguagem, que até pode inicialmente ser uma linguagem visual, que é mais próxima da realidade concreta. Assim, quando esse conceito for formalizado usando-se linguagens textuais e linguagens de programação, o passo de aprendizado que o aluno precisará dar será menor, pois ele já compreende o conceito em si, só precisa aprender a representá-lo de outra forma. Essa proposta de evolução no aprendizado da construção de algoritmos, que é fruto de décadas de experiência de ensino de professores de Computação, parece ser simples, mas é de certa forma revolucionária, pois o ensino de algoritmos sempre teve grande ênfase na linguagem (sintaxe) ao invés dos conceitos de dados e processos (semântica), o que normalmente causa grandes dificuldades no aprendizado.

Organização das diretrizes

Nos anos iniciais, os alunos já são expostos à noção básica de algoritmos quando, por exemplo, ensinam-se as operações aritméticas básicas. A expectativa é que isso seja enfatizado, de forma que os estudantes tomem consciência da noção básica de algoritmo, sendo capazes de, a partir de conjuntos de instruções diversos, seguir e elaborar algoritmos para solucionar diferentes tipos de problemas, usando linguagem natural e linguagens pictográficas. Devem dominar as principais operações para a construção de algoritmos (composição sequencial, seleção e repetição) e ter noções de técnicas de decomposição de problemas. Além disso, espera-se que os estudantes reconheçam a necessidade de classificar objetos em conjuntos, cujos elementos podem ser atômicos (como números, palavras, valores-verdade) ou estruturados (como registros, listas e grafos), sendo capazes de trabalhar com elementos desses conjuntos e identificar situações concretas nas quais dados atômicos ou estruturados possam ser utilizados. O essencial, nessa etapa, é que os conceitos sejam dominados por meio de experiências concretas, que permitirão ao estudante construir modelos mentais para as abstrações computacionais, que serão formalizadas na próxima etapa do ensino fundamental (anos finais) com o uso de linguagens de programação. Ou seja, é muito importante que o Pensamento Computacional seja trabalhado de forma desplugada (sem o uso de computadores) nos anos iniciais.

No eixo de Mundo Digital, inicia-se trabalhando o conceito de informação: o que é, sua importância, por que descrevê-la, protegê-la, comunicá-la. Naturalmente, surge a noção de código e também de máquina, que pode ser usada para armazenar e processar informação (computador), bem como a relação entre a máquina e o algoritmo (software e hardware).

Nos anos finais, espera-se que os estudantes sejam capazes de selecionar e utilizar modelos e representações adequadas para descrever informações e processos, bem como dominem as principais técnicas para construir soluções algorítmicas. Além disso, devem conseguir descrever as soluções, de forma que máquinas possam executar partes ou todo o algoritmo proposto; construir modelos

computacionais de sistemas complexos e analisar criticamente os problemas e suas soluções. Nessa etapa, deve ser adquirido também um entendimento de como informações podem ser armazenadas, protegidas e transmitidas, e da estrutura e funcionamento da internet, permitindo que o aluno tenha plena compreensão do Mundo Digital, suas potencialidades e seus limites.

As Figuras 1.5, 1.6 e 1.7 descrevem, de maneira resumida, conceitos importantes a serem trabalhados nas etapas do ensino fundamental. Uma descrição mais detalhada envolvendo objetos de conhecimento e habilidades relacionadas a eles pode ser encontrada na proposta de Diretrizes de Ensino de Computação na Educação Básica (SBC, 2018).

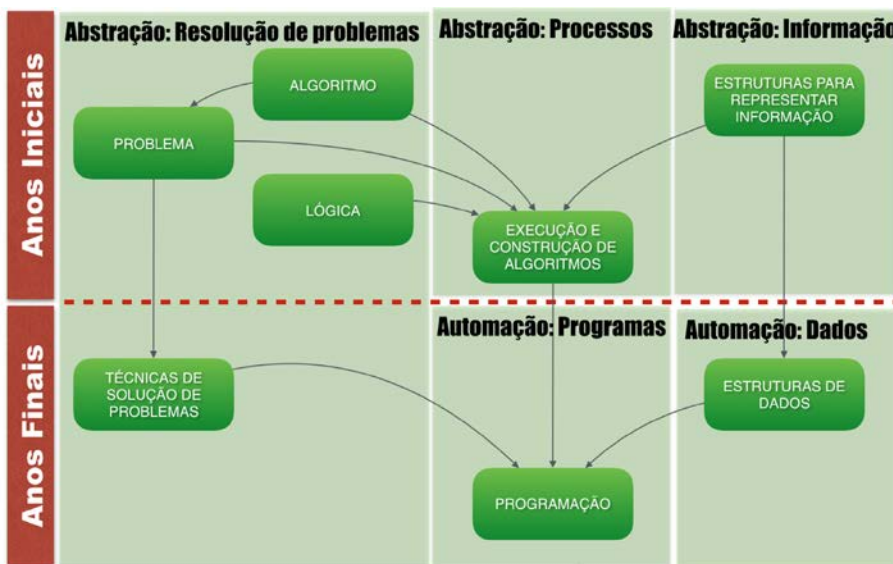


Figura 1.5 – Ensino fundamental: Eixo Pensamento Computacional

Fonte: Sociedade Brasileira de Computação, 2018, p. 8

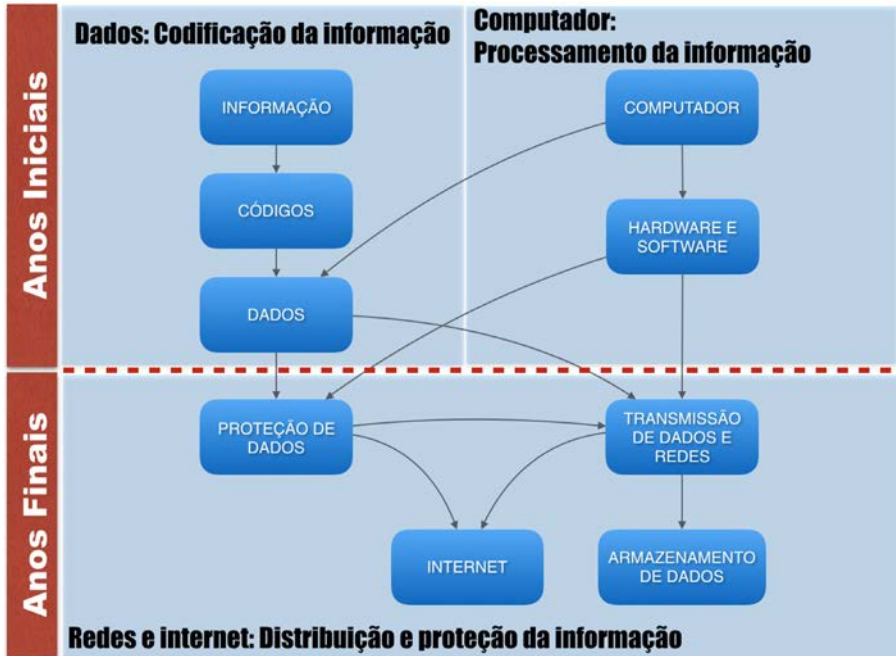


Figura 1.6 – Ensino fundamental: Eixo Mundo Digital
Fonte: Sociedade Brasileira de Computação, 2018, p. 8

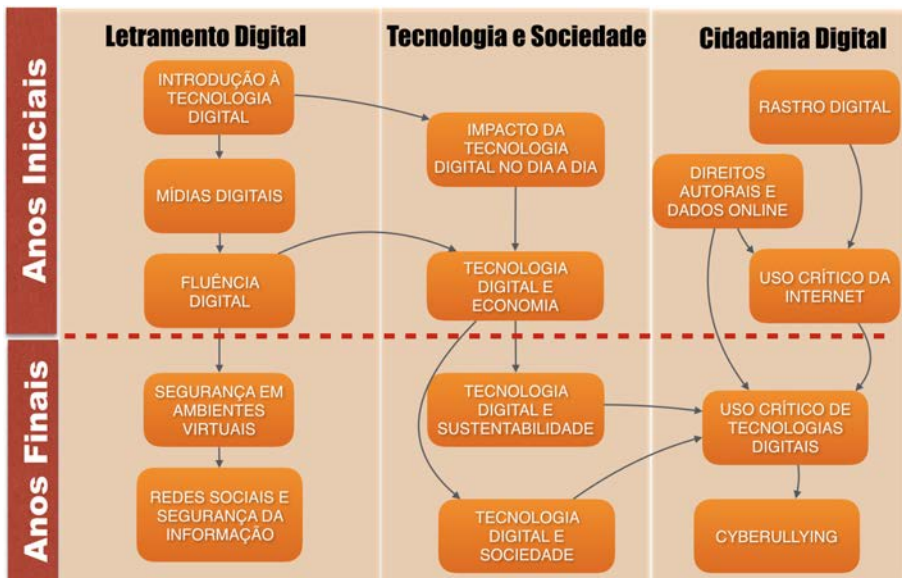


Figura 1.7 – Ensino fundamental: Eixo Cultura Digital
Fonte: = Sociedade Brasileira de Computação, 2018, p. 9

No ensino médio, a ênfase é na elaboração de projetos aplicando as diversas habilidades e conhecimentos adquiridos na etapa do ensino fundamental, e no desenvolvimento de habilidades relacionadas à análise crítica e argumentação, sob diferentes aspectos. Note que essa proposta de ensino médio assume que o aluno já trabalhou com conceitos de Computação durante nove anos, ou seja, todo o ensino fundamental. Assim, é possível que, no ensino médio, sejam trabalhados os conceitos apresentados a seguir, e resumidos na Figura 1.8.

No eixo de Pensamento Computacional, são trabalhadas a técnica de transformação de problemas e o paradigma de metaprogramação (algoritmos que recebem outros algoritmos como entrada), que são conceitos necessários para a compreensão dos limites da computação, ou seja, dos limites da formalização/racionalização. Esse entendimento, aliado aos fundamentos de inteligência artificial e robótica, provê a base necessária para uma discussão mais consubstanciada sobre o que é o Homem e o que é a Máquina, quais as similaridades e diferenças, não somente do ponto de vista físico, mas do ponto de vista filosófico, entendendo também as grandes questões éticas envolvidas na inteligência artificial. Outro conceito fundamental é a análise de algoritmos, tanto do ponto de vista de correção quanto de eficiência. Ao final do ensino médio, o aluno deve ter a habilidade de argumentar sobre algoritmos (processos), tendo meios de justificar por que a sua solução resolve de fato o problema, bem como analisar os tipos e quantidade de recursos necessários à sua execução. Além dessas habilidades, o aluno deverá dominar a técnica de refinamento, compreendendo que a solução de problemas complexos normalmente exige não somente decomposição, mas também que se utilizem várias camadas de abstração diferentes.

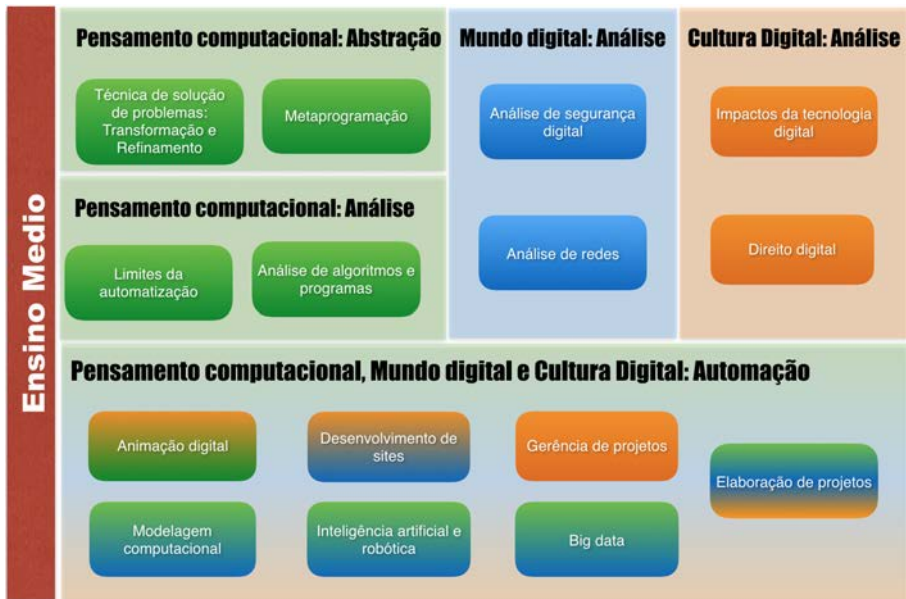


Figura 1.8 – Ensino fundamental: Eixo Cultura Digital

Fonte: Sociedade Brasileira de Computação, 2018, p. 19

Complementarmente, no eixo de Mundo Digital também serão trabalhadas habilidades que envolvem análise, nesse caso, análise crítica de redes e de segurança digital. Será trabalhado o conceito de Big Data, pois é importante que o aluno tenha a capacidade de utilizar dados massivos para resolver seus problemas.

Finalmente, a elaboração de projetos de modelagem computacional envolvendo tanto conceitos do Pensamento Computacional quanto do Mundo Digital e de Cultura Digital é de extrema relevância para os alunos trabalharem colaborativamente na busca de soluções para problemas de diversas áreas, compreendendo como a Computação pode ser usada tanto no seu dia a dia quanto no mundo do trabalho, provendo ferramentas e habilidades essenciais.

CONSIDERAÇÕES FINAIS

O papel da formação escolar é dar subsídios para os alunos não somente exercerem suas profissões, mas também para se adaptarem às transformações decorrentes dos avanços tecnológicos. Precisamos preparar os jovens para profissões que ainda não existem, usando tecnologias que ainda não foram inventadas e resolvendo problemas que ainda não sabemos que são problemas. Certamente, grande parte dessas profissões envolverá Computação de alguma forma, direta ou indiretamente. Porém a validade do conhecimento tecnológico é curta, por isso, o ensino de tecnologias digitais na Educação Básica precisa ser aliado aos conceitos fundamentais da Computação. Queremos que a Educação viabilize a formação de cidadãos e profissionais que compreendam as possibilidades e oportunidades de automação, ainda que não detenham as competências de implementá-las (infelizmente, hoje tipicamente restritas aos profissionais da Computação). Mas é necessário formar cidadãos capazes de compreender e perceber em que situações a Computação pode lhes ser útil, seja reduzindo esforços, custos, seja melhorando a qualidade dos resultados. A compreensão do mundo digital e as estratégias de resolução e análise de problemas desenvolvidas pela área da Computação proveem, portanto, aprendizagens essenciais para qualquer cidadão do século XXI, permitindo que ele crie e inove em todas as áreas do conhecimento.

A área que está mudando o mundo neste momento, tanto nos processos produtivos (por exemplo, indústria 4.0) quanto nas relações sociais (por exemplo, novas formas de comunicação e linguagens usadas nas redes sociais), é a Computação. Vários princípios da produção moderna são da área de Computação ou fortemente influenciados por ela, como Inteligência Artificial, Robótica, Bioinformática, análise de grandes volumes de dados (Big Data) etc. O impacto desses avanços no mercado de trabalho é comparável à Revolução Industrial: vários estudos apontam que grande parte dos postos de trabalho nos moldes atuais serão supridos por computadores. Ou seja, num futuro muito próximo (da geração que está atualmente na escola), os empregos exigirão ou que o profissional seja capaz de usar de forma crítica, segura e eficiente as ferramentas

computacionais ou que seja capaz de projetá-las. Além disso, a Computação provê variadas formas de expressão a partir de linguagens textuais e visuais que permitem descrever soluções de problemas em diferentes níveis de abstração. Dominar essas novas formas de expressão é essencial para interagir e controlar os artefatos computacionais que permearão ainda mais a sociedade. **Portanto, é fundamental que o aluno domine os princípios da Computação tanto para compreender a produção moderna quanto as formas contemporâneas de linguagens.**

A Sociedade Brasileira de Computação (SBC) acredita que:

- **Computação é essencial** na formação do cidadão do século XXI, e, portanto, deve fazer parte dos currículos de todas as escolas do Brasil;
- Os **fundamentos da Computação** (e não tecnologias) devem ser ensinados ao longo da Educação Básica com **intencionalidade**. Para isso, os objetos de conhecimento e habilidades relacionados à Computação precisam estar bem definidos e disponíveis para toda a rede escolar;
- Computação deve ser ensinada por **professores capacitados** na área, essa capacitação pode se dar por meio de licenciatura específica ou de formação complementar para professores de outras áreas;
- O ensino de **Computação** na Educação Básica deve ser entendido como **estratégico para o Brasil**, tanto para que sua população atinja melhores patamares de qualidade de vida quanto do ponto de vista **econômico, social e científico**.

REFERÊNCIAS

ACARA (s/d). Australian Curriculum. <https://www.australiancurriculum.edu.au/>.

Alfred, V. (2011). Aho 1. Ubiquity. January 2011. Retrieved from <https://doi.org/10.1145/1922681.1922682db/journals/ubiquity/ubiquity2011.html#Aho11>.

Baltimore, D. (2001). How biology became an information science. In P. Denning (Ed.). *The Invisible Future: The Seamless Integration of Technology into Everyday Life*, pp. 43–46. New York: McGraw-Hill.

Base Nacional Comum Curricular (s/d). Homologada pelo MEC. Recuperado de <http://basenacionalcomum.mec.gov.br>.

Bordini, A., Avila, C. M. O., Weissshahn, Y., Cunha, M. M., Cavalheiro, S. A. C., Foss, L., Aguiar, M., & Reiser, R. H. S. (2016). Computação na Educação Básica no Brasil: o Estado da Arte. *RITA*, 23(2): 210–238 (2016)

CSTA K12 (s/d). *Computer Science Standards*. Recuperado de <https://www.csteachers.org/page/standards>.

CFE ARGENTINA (2015). *Resolución No 263/15*. Recuperado de <http://www.me.gov.ar/consejo/resoluciones/res15/263-15.pdf>.

Denning, P. J. (2007). *Computing is a natural science*. Commun. ACM 50, 7 (July 2007), 13–18. DOI: <https://doi.org/10.1145/1272516.1272529>.

Guerra, V., Kuhnt, B., & Blöchliger, I. (2012). *Informatics at school - worldwide. An international exploratory study about informatics as a subject at different school levels*. Technical Report, University of Zurich. 67 p.

GOV.UK (2013). *Department for Education. National curriculum in England: computing programmes of study*. Recuperado de <https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study>.

OPS (2016). *Esi- ja perusopetuksen opetussuunnitelman perusteiden uudistaminen*. Recuperado de <https://www.oph.fi/ops2016>.

Ribeiro, L., Foss, L., & Cavalheiro, S. A. C. (2017). *Entendendo o Pensamento Computacional*. CoRR abs/1707.00338.

Sociedade Brasileira de Computação (2018). *Diretrizes para ensino de Computação na Educação Básica*. Recuperado de <http://www.sbc.org.br/educacao/diretrizes-para-ensino-de-computacao-na-educacao-basica>.

Wing, J. M. (2008). Computational thinking and thinking about computing. *Proceedings of the 2008 IEEE International Symposium on Parallel and Distributed Processing Symposium*, Miami, FL, USA.

Wolfram, S. (2002). *A new kind of science II-XIV*, 1-1197 Wolfram-Media 2002 978-1-57955-008-0.

Wolfram, S. (2017). A New Kind of Science: A 15-Year View. *Complex Systems*, 26(3).

SEÇÃO 2

ARTIGOS CIENTÍFICOS SOBRE PENSAMENTO COMPUTACIONAL NA EDUCAÇÃO BÁSICA

Nesta seção, serão apresentados seis artigos científicos, com objetivo de apresentar propostas de desenvolvimento de pensamento computacional na educação básica em âmbito nacional. Todos os artigos foram selecionados com o foco de demonstrar propostas práticas de desenvolvimento do Pensamento Computacional na Educação Básica, usando como base a implementação da Educação Integral. Em linhas gerais, a definição dos artigos envolveu a divulgação de chamada em fóruns relacionados ao tema, submissão livre de trabalhos e seleção dos seis artigos com base em revisão por pares cega.

Iniciaremos apresentando o artigo “Ensino de Pensamento Computacional com Jogos e Internet das Coisas”, de autoria de Bruno da Silva Rodrigues, Ana Grasielle Dionísio Corrêa, Nizam Omar e Valéria Farinazzo Martins, que propõe como o uso das Tecnologias de Informação e Comunicação no contexto escolar pode contribuir para o desenvolvimento de diversas habilidades, entre elas o raciocínio lógico e a capacidade de resolução de problemas.

O segundo artigo tem o título “dTECT – Um Modelo para a Avaliação de Unidades Instrucionais para o Ensino de Computação na Educação Básica”, de autoria de Christiane Gresse von Wangenheim, Giani Petri e Adriano F. Borgatto, e propõe um modelo confiável para a avaliação da qualidade de unidades instrucionais para o ensino de computação na educação básica com base na percepção dos alunos.

O terceiro artigo, intitulado “A Modelagem Matemática como Metodologia para o Desenvolvimento do Pensamento Computacional por meio da Criação de Objetos de Aprendizagem”, de autoria de Márcia Regina Kaminski e Clodis Boscarioli, objetiva apresentar a experiência de uma escola pública municipal que tem empregado a Modelagem Matemática e a criação de Objetos de Aprendizagem com Scratch como metodologia para o desenvolvimento do Pensa-

mento Computacional, na perspectiva de formação integral, no 5º ano fundamental.

O quarto artigo do capítulo tem o título “Desenvolvimento e Impacto da Integração Curricular de Programação e Pensamento Computacional em Todas as Disciplinas e Séries do Ensino Fundamental da Fundação Torino Escola Internacional”, de autoria de Ana O. Rodrigues, Ana Márcia A. M. Paiva, Sabrina M. de Oliveira e Márcia L. A. A. Naves, e apresenta o processo de implantação e o impacto de um projeto de três anos para integrar programação em todas as disciplinas e séries do ensino fundamental I e II da Fundação Torino.

O quinto artigo, intitulado “Pensamento Computacional e programação: impactos na formação de professores e contribuições para práticas pedagógicas interdisciplinares”, que tem como autores Eliel Constantino da Silva, Maria Teresa Zampieri e Sueli Liberatti Javaroni, tem por objetivo produzir conhecimento sobre a integração do Pensamento Computacional e programação por meio do uso do Scratch e outros recursos educacionais presentes na escola, em abordagens pedagógicas nos anos finais do ensino fundamental, envolvendo Geografia e Matemática.

Por fim, o sexto artigo, com o título “Estimulando o Pensamento Computacional por meio de Oficinas Lúdicas no Litoral Norte Gaúcho”, de autoria de Vitória de Souza Fabrício, Maurício Braga Julio, Augusto Weiland e Anelise Lemke Kologeski, apresenta o projeto de extensão “Programando Fácil: Conhecendo a Computação”, que oferece oficinas lúdicas, usando jogos digitais, no litoral norte do estado do Rio Grande do Sul.

CAPÍTULO 3

ENSINO DE PENSAMENTO COMPUTACIONAL COM JOGOS E INTERNET DAS COISAS

*Bruno da Silva Rodrigues
Ana Grasielle Dionísio Corrêa
Nizam Omar
Valéria Farinazzo Martins*

INTRODUÇÃO

Desde que houve a popularização dos computadores, há cerca de três décadas, mais usuários passaram a ser especialistas em computação e a trabalhar com programação. Assim, aprender programação passou a ser algo extremamente importante e não deve ser exclusividade dos estudantes de Computação. Inúmeros fatores apontam a relevância de desenvolver competências relacionadas à programação com estudantes em idade escolar (ensino fundamental e médio), culminando em habilidades úteis para a resolução de problemas: a melhoria da capacidade de raciocínio lógico desses estudantes e a visão correta do uso da tecnologia de que eles são consumidores (Scaico *et al.*, 2013).

De acordo com Wing (2006), pensamento computacional pode ser definido como sendo um conjunto de competências e habilidades envolvendo abstração e decomposição de problemas a fim de torná-los subpartes menores e menos complexas, focadas nos detalhes. Permite, assim, a resolução usando recursos computacionais e estratégias algorítmicas (Barcelos & Silveira, 2013).

O ensino do pensamento computacional foi incluso na Base Nacional Curricular Comum (BNCC), que define o que deve ser ensinado aos alunos em todas as escolas do país; assim, a abrangência do ensino da Computação passa a ser de um aprendizado básico auxiliando no desenvolvimento de conhecimentos e habilidades quanto a atitudes e valores aprimorando capacidades “de compreender, analisar, definir, modelar, resolver, comparar e automatizar problemas e suas soluções, de forma metódica e sistemática, por meio do desenvolvimento de algoritmos” (Brasil, 2018).

Por outro lado, a indústria de jogos digitais e o desenvolvimento de soluções de internet das coisas (do inglês, *Internet of things* – IoT) têm tido um enfoque bastante promissor. A indústria de jogos é uma das que mais cresce no mundo, com um faturamento global de quase US\$100 bilhões, conforme apontado por Fleury et al. (2015); já a consultoria da IDC (2017) estima que os gastos mundiais com a Internet das Coisas (IoT) devem chegar a US \$ 772,5 bilhões em 2018, o que torna ambos os temas presentes na vida das pessoas, sobretudo das crianças. Ainda, vários estudos (Martins & Guimarães, 2018; Perreira, Pacheco, & Fernandes, 2016; Savi et al., 2010) apontam o aumento de motivação nos estudantes, em cursos de Computação, quando o assunto dos projetos envolve jogos eletrônicos. No entanto ainda não existem muitos estudos sobre a motivação para o desenvolvimento de jogos por estudantes de ensino fundamental.

Este projeto faz parte do 2º ciclo de cursos sobre Pensamento Computacional concebido no conceito de projeto de extensão, numa parceria entre a Faculdade de Computação e Informática e o Colégio Presbiteriano Mackenzie. O objetivo deste projeto foi proporcionar aos alunos, que já tinham cursado o 1º ciclo, noções um pouco mais avançadas sobre princípios de lógica de programação de computadores, tais como: variáveis, repetição, geração de números randômicos, incremento de variáveis etc. Esse objetivo foi implementado por intermédio do desenvolvimento de jogos digitais nas três primeiras aulas e também de noções de Internet das Coisas e Arduino na última aula.

Este capítulo está estruturado como segue. Na seção 2, são apresentados os temas necessários para o entendimento deste capítulo: Scratch, Internet das Coisas e Arduino. Na seção 3, são apresentados os trabalhos relacionados. A seção 4 traz a metodologia do curso. A seção 5 apresenta a avaliação da metodologia. Na seção 6, estão os resultados da execução do 2º ciclo de cursos sobre Pensamento Computacional. Finalmente, na seção 7, são apresentados as conclusões e trabalhos futuros.

FUNDAMENTAÇÃO TEÓRICA

Scratch

Scratch é um ambiente para ensino de programação para iniciantes desenvolvido por pesquisadores do Instituto de Tecnologia de Massachusetts (Massachusetts Institute of Technology – MIT), nos Estados Unidos (Maloney *et al.*, 2008).

Inspirado pelo LogoBlocks (Wolz *et al.*, 2008) e pelo Etoys, o Scratch se baseia nas ideias de Logo (McCray *et al.*, 2003), que substitui o código digitado por uma abordagem de “arrastar-e-soltar” blocos de comandos (Figura 1). Cada tipo de bloco é diferenciado por uma cor característica que é utilizada para criar construções mais complexas aninhadas de repetição, condição, eventos, sons, variáveis, entre outros.

Sua ênfase em gráficos, animação, som e interação com o usuário permite que os estudantes foquem sua energia em problemas que envolvam o seu interesse (Margolis *et al.*, 2002). Esses códigos podem ser compartilhados e distribuídos, reutilizados e remixados por toda a comunidade, incentivando a ideia de software livre. Existem diversos softwares inspirados no Scratch, tais como o Pyonkee (uma versão do Scratch para iPad); GameMaker; Code.org, entre outros.

O ambiente Scratch traz o conceito de Computação Criativa (expressão utilizada para reconhecer que o conhecimento e as práticas dos estudantes precisam passar por seus interesses pessoais). Também foi projetado para proporcionar o engajamento interativo, experimental e lúdico, fazendo com que o estudante explore e aprenda, imaginando novas possibilidades (Scaico *et al.*, 2013). Devido à sua sintaxe fácil, o caráter mais didático da linguagem é utilizado para que o estudante se concentre no exercício do pensamento algorítmico e em sua criatividade durante a construção de soluções de seus problemas.

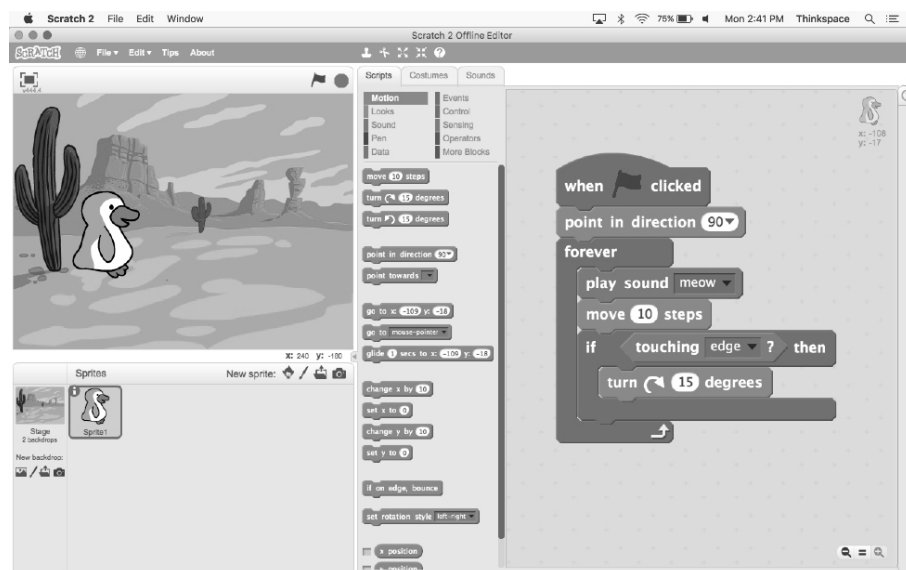


Figura 1 – O ambiente de programação Scratch

Fonte: Scratch, 2019

Em virtudes dos benefícios proporcionados pelo Scratch, diferentes soluções usando blocos de comandos foram propostas voltadas a diferentes objetos de aplicação, um exemplo de solução usando blocos de comandos é a plataforma Tinkercad (Autodesk, 2018). Desenvolvida pela Autodesk, essa plataforma possui um emulador de circuitos eletrônicos que possibilita ao usuário a montagem e simulação de circuitos usando Arduino, em que o Arduino pode ser programado usando Scratch. O ambiente de desenvolvimento do Tinkercad pode ser visto na Figura 12, em que o simulador de circuitos eletrônicos e Scratch são integrados na mesma ferramenta.

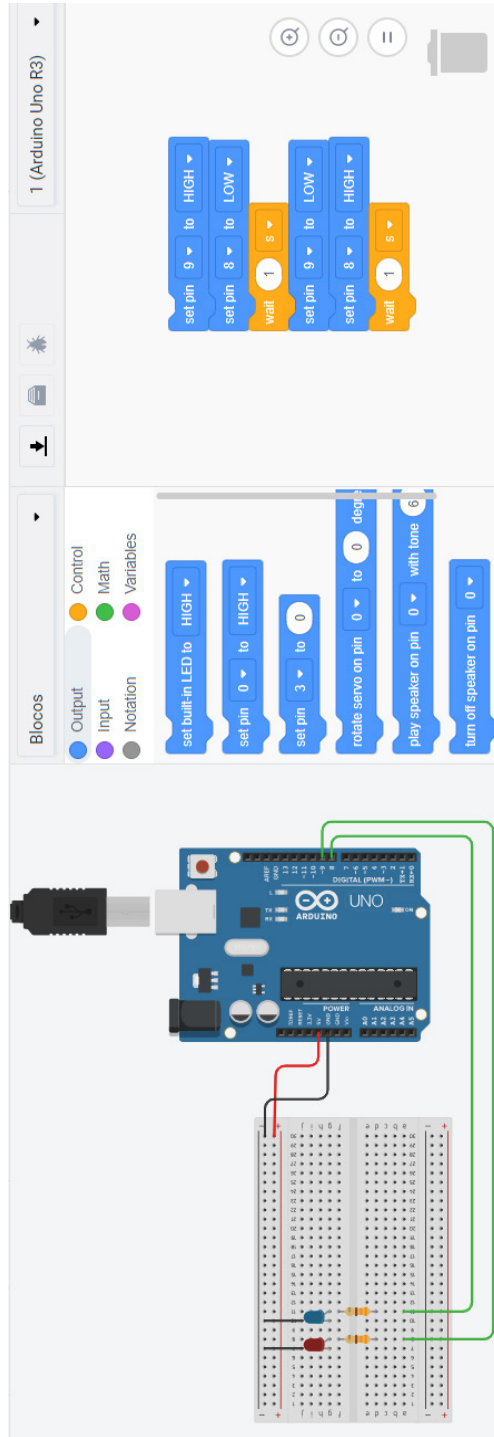


Figura 2 – O ambiente de programação Tinkercad

Fonte: Autodesk, 2018

Internet das Coisas

A criação e a consolidação da Internet que interconectou aplicações e processos em diferentes sistemas finais ao redor do mundo transformou a maneira como as pessoas interagem com a tecnologia. No início da década de 90, Weiser (1991) previa que os avanços nas telecomunicações e na computação móvel aumentaria a diversidade de serviços de computação, permitindo o acesso a esses serviços o tempo todo e independentemente da localização do usuário final. Esse conceito ficou conhecido como computação ubíqua ou computação pervasiva.

O termo Internet das Coisas (IoT) foi criado por Ashton e se refere à capacidade de objetos conectados à internet coletarem informações do ambiente a partir de tecnologia de sensores (Ashton, 2009). Além da coleta de informações, diferentes objetos IoT podem interagir em função das variações ambientais, permitindo uma melhor percepção do contexto no qual os objetos estão inseridos e, a partir de aplicações inteligentes, os objetos podem controlar ou ser controlados para responderem dinamicamente ao ambiente (Barnaghi et al., 2012).

Devido à versatilidade dos sistemas de IoT, soluções inteligentes em transporte, fornecimento de energia, segurança, atendimento médico domiciliar, entre outras, têm sido desenvolvidas. Além das aplicações citadas que proveem uma melhor qualidade de vida para os beneficiários do sistema (Kyriazis et al., 2013), no ensino, o uso de IoT aliado a estratégias de Aprendizagem Baseada em Projetos pode auxiliar na aprendizagem dos conteúdos curriculares (Rosa, 2017), aumentando o engajamento e motivação dos alunos (Fernandez, 2015).

Arduino

Criado por um grupo de professores no Interaction Design Institute de Ivrea (Itália) (Severance, 2014), o Arduino é uma plataforma de prototipagem eletrônica de hardware livre e de placa única

projetada com um microcontrolador Atmel AVR (Mcroberts, 2011), que permite criar projetos de baixo custo, flexíveis, de fácil implementação mesmo por profissionais com conhecimentos básicos em eletrônica (Banzi, 2015). Essas características contribuíram com a popularização do Arduino e, atualmente, existe uma variedade de plataformas (Figura 3) adaptadas às necessidades específicas, que vão desde plataformas de aprendizagem a dispositivos vestíveis e impressões 3D (Arduino, 2016).

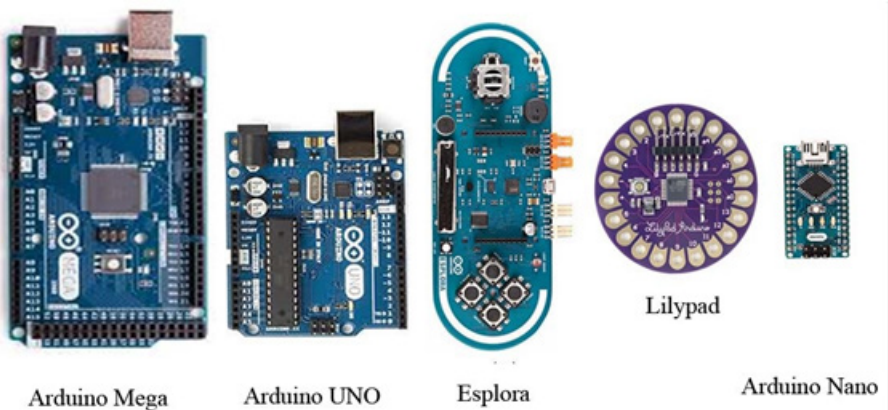


Figura 3 – Modelos de Arduino

Fonte: Rodrigues, Lima, & Oliveira, 2018

TRABALHOS RELACIONADOS

Vários trabalhos sobre o ensino de Pensamento Computacional com o uso de Scratch foram encontrados na literatura, no contexto brasileiro. Scaico *et al.* (2013) trazem um relato do uso de Scratch para o ensino de programação no ensino médio, em escolas no interior da Paraíba, com carga total de 20 horas. O curso foi implementado por meio de uma Olimpíada Interna de Programação com Scratch.

Oliveira *et al.* (2014) apresenta um relato de experiência sobre o ensino de programação utilizando Scratch para o ensino fundamental. Foi utilizada carga horária de 20 horas, em dez encontros semanais, de duas horas cada, para 20 alunos do 9º ano do ensino fundamental. O curso foi dividido em três etapas: conceitos computacionais, criação de animações e desenvolvimento de jogos simples.

Bezerra e Dias (2014) apresentam a experiência da oferta de oficinas de programação de computadores no ensino fundamental em duas escolas públicas de Belém, Pará, de LOGO com KTurtle e Scratch. As oficinas de Scratch tinham carga de 30 horas, duas vezes por semana, com aulas de 90 minutos. Os estudantes não foram categorizados em relação à idade, mas eram estudantes do 8º ano. Não foram abordados no trabalho os temas centrais que foram trabalhados na oficina.

Rodriguez *et al.* (2015) apresentam a experiência vivenciada em um projeto, realizado no âmbito de um programa de Pré-Iniciação Científica, cujo objetivo principal foi desenvolver noções básicas do Pensamento Computacional, junto aos alunos do primeiro ano do ensino médio – alunos entre 14 e 16 anos – de uma escola pública, por meio dos recursos do Scratch. Os estudantes desenvolveram três jogos educativos. O primeiro para o ensino de Matemática, envolvendo as quatro operações básicas. O segundo destinado aos alunos que vão prestar provas dos vestibulares. Finalmente, o terceiro para conscientização do meio ambiente.

Já em relação ao ensino de robótica, Kalil *et al.* (2013) apresentam um relato de experiência do uso das ferramentas Arduino, Scratch for Arduino (S4A) e Lego Mindstorms por sete estudantes de escolas públicas do ensino médio. O projeto teve como foco principal investigar quais ferramentas são mais efetivas para promover maior motivação a esses estudantes, assim como estimular o uso de robótica educacional.

Pode-se notar que existem vários estudos e relatos de experiência do uso de Scratch para o ensino de Pensamento Computacional. No entanto não foram encontrados trabalhos que vivenciassem o desenvolvimento de desafios com o uso de Arduino para um público ainda mais jovem, que faz parte dessa proposta.

METODOLOGIA DO CURSO

O curso “Estimulando o Pensamento Computacional com a Ferramenta Scratch” foi organizado em ciclos e ministrado por quatro professores universitários da Faculdade de Computação e Informática e corresponde a um projeto de extensão.

Este corresponde à segunda etapa do curso. No primeiro ciclo, com 20 horas ao todo, os estudantes aprenderam noções básicas de programação com Scratch e desenvolveram projetos de animações simples utilizando os sprites (personagens) e cenários disponibilizados pela ferramenta. No segundo ciclo (relatado neste capítulo), composto por 16 horas, os estudantes desenvolvem jogos, ou seja, são projetos mais bem elaborados. Assim, ao todo, o curso com os dois módulos é composto por 36 horas.

Três dos quatro encontros ocorreram numa sala de multimeios do Colégio Presbiteriano Mackenzie, entre os meses de outubro e novembro de 2018, com duração de 1h30min cada aula. Nessas aulas, os alunos usaram Ipads com o software Pyonkee (2018). Esse software foi adotado por se tratar de uma versão do Scratch para plataformas iPad, que é atualmente a plataforma adotada pelo colégio para introduzir os estudantes no meio digital. O Pyonkee é de uso gratuito, possui opção na língua portuguesa e está disponível para plataforma iPad. A introdução do Pyonkee no segundo ciclo ocorreu de forma natural, visto que os estudantes já haviam cursado o primeiro ciclo de Pensamento Computacional com Scratch em uma sala com computadores. A Figuras 4 ilustra momentos do curso em que os alunos utilizam o Pyonkee em um iPad para criar seus jogos.

A proposta executada em cada encontro está descrita nas seguintes subseções.



Figura 4 – Momentos de uso do Pyonkee em plataformas iPad

Fonte: autores

A Figura 5 ilustra momentos do curso nos quais os alunos utilizam a plataforma Arduino e o software Tinkercad (baseado em Scratch) para criar seus projetos de eletrônica. A proposta executada em cada encontro está descrita nas seguintes subseções.



Figura 5 – Momentos do uso do Arduino e Tinkercad

Fonte: autores

Primeiro encontro

No primeiro encontro, os alunos desenvolveram um jogo de boliche, tal como apresentado na Figura 6. Para tanto, noções de variáveis computacionais, geometria plana (noções de coordenadas 2D no plano cartesiano), variável acumuladora para pontuar o jogo e geração de números aleatórios foram necessárias.

O problema foi decomposto em passos mais simples a fim de solucionar o problema:

1. Apresentar a tela de abertura do jogo.
2. Depois de cinco segundos, mudar para a tela de jogo.
3. Posicionar os objetos na tela (bola de boliche e pinos).
4. Criar animação da bola de boliche.
5. Verificar se houve interação com a bola e, caso positivo, executar item 6.
6. Pontuar se a bola entra em contato com o pino, caso contrário, posicionar a bola na sua posição inicial.



Figura 6 – Cenário do Scratch para o jogo de boliche

Fonte: autores

Segundo encontro

Neste encontro, os alunos desenvolveram o jogo Space Invaders (Figura 7). Esse jogo é composto por uma nave que atira raios para tentar explodir os meteoros que caem. Há um botão de disparar raios e setas para a esquerda e direita para o jogador posicionar melhor a nave. Para tanto, noções de variáveis computacionais, geometria plana (noções de coordenadas 2D no plano cartesiano), variável acumuladora para pontuar o jogo e consumir vidas foram necessárias.

O problema foi decomposto em passos mais simples a fim de solucioná-lo:

- Apresentar a tela de abertura do jogo.
- Depois de cinco segundos, mudar para a tela de jogo.
- Posicionar os objetos na tela (meteoros e nave).
- Vincular à nave as setas para sua movimentação (esquerda e direita).
- Disparar raio ao clicar no botão.
- Criar animação do raio para cima.
- Criar animação do meteoro deslizando para baixo.
- Trabalhar com variáveis Pontos e Vidas.
- Trabalhar as regras de contatos entre meteoro, raio e nave.
- Finalizar o jogo quando a variável Vidas for 0.



Figura 7 – Cenário do Scratch para o jogo do Space Invaders

Fonte: autores

Terceiro encontro

Neste encontro, os alunos aprenderam a desenvolver o jogo Joquempô (também conhecido como jogo “papel, pedra, tesoura” – Figura 8). Em seguida, com a ajuda dos professores e alunos voluntários do curso de Ciência da Computação, os estudantes estabeleceram as lógicas do jogo. Os alunos também tiveram noções de geração de números aleatórios e variáveis computacionais.

O problema foi decomposto em passos mais simples a fim de solucionar o problema:

- Tirar foto das mãos e gerar as imagens a serem incluídas no jogo.
- Gerar o jogador 1 (humano) e o jogador 2 (máquina).
- Sortear a escolha da máquina (números aleatórios).

- Criar as regras de lógica do jogo: o papel ganha da pedra, a tesoura ganha do papel e a pedra ganha da tesoura.
- Trabalhar com variáveis Pontos para definir quem ganhou (humano ou máquina) depois de uma série de jogadas.

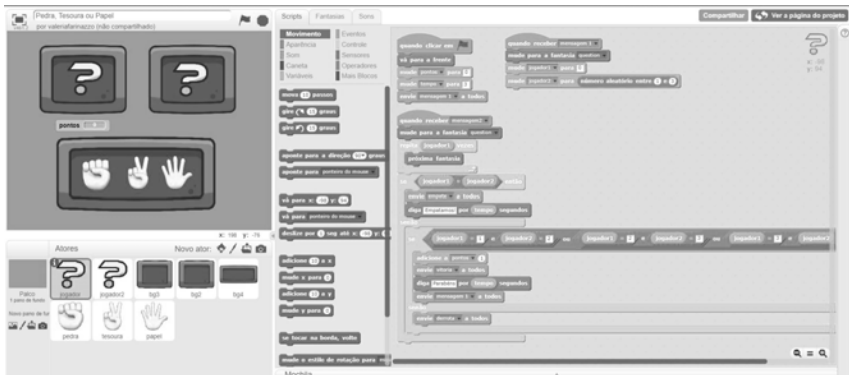


Figura 8 – Cenário do Scratch para o jogo Joquempô

Fonte: autores

Quarto encontro

Neste último encontro, foi introduzido aos alunos o conceito de Internet das Coisas e o Arduino que foi usado para controle de leds (*light-emitting diode*). Além do desenvolvimento do pensamento computacional a partir da programação do Scratch, foram introduzidas aos alunos noções básicas de eletrônica para simulação e implementação dos circuitos. Nessa aula, o conhecimento foi construído a partir da criação de um circuito no qual o Arduino foi programado para fazer um led oscilar com uma frequência de 0,5Hz, e posteriormente modificaram o circuito adicionando novos leds – o objetivo final da aula era a criação de um semáforo (Figura 9).

O problema foi decomposto em passos mais simples a fim de solucionar o problema:

- Criação do circuito eletrônico para acionamento do led pelo Arduino.
- Vincular leds às saídas digitais do Arduino.
- Acionamento temporizado dos leds.
- Simulação do circuito.
- Implementação do circuito simulado no Arduino.
- Gravação do código gerado a partir do Scratch no Arduino.
- Criação de novo circuito no simulador usando dois leds.
- Simulação do circuito.
- Implementação do circuito simulado no Arduino.
- Gravação do código gerado a partir do Scratch no Arduino.
- Criação do semáforo no simulador usando três leds.
- Simulação do circuito.
- Implementação do circuito simulado no Arduino.
- Gravação do código gerado a partir do Scratch no Arduino.

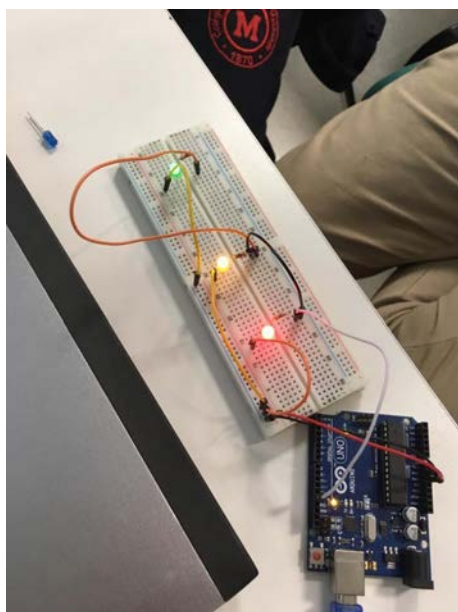


Figura 9 – Semáforo Arduino –
aplicação desenvolvida pelos alunos
Fonte: autores

AVALIAÇÃO DA METODOLOGIA

A fim de verificar a efetividade desse curso, foram criados dois questionários, sendo um de perfil e outro de satisfação para os estudantes, para serem respondidos no último dia de aula. As Figuras 10 e 11 apresentam esses questionários.

Questionário de Perfil dos Estudantes

Nome:

Série:

Idade:

Sexo:

Quais tecnologias você já usou?

(☐) Computador (☐) Tablet (☐) Smartphones (☐) Videogame

Quantas vezes por semana você usa o computador?

(☐) Todos os dias (☐) 3 a 5 vezes por semana (☐) Menos de 2 vezes por semana (☐) Nunca

Antes da aula, você sabia o que era Arduino?

(☐) Sim (☐) Não (☐) Não tenho certeza

Antes da aula, você já tinha ouvido falar de temas como robótica ou internet das coisas?

(☐) Sim (☐) Não (☐) Não tenho certeza

Figura 10 – Questionário de perfil dos estudantes

Fonte: autores

Questionário de Satisfação

Eu gostaria de aprender a mexer no Scratch um pouco mais

() Sim () Não () Mais ou menos

Foi fácil aprender a mexer no Scratch para fazer jogos?

() Sim () Não () Mais ou menos

O que você achou mais fácil no Scratch?

R: ____

O que você achou mais difícil no Scratch?

R: ____

Eu gostaria de continuar a mexer no Scratch para fazer mais jogos

() Sim () Não () Mais ou menos

Eu gostei de aprender a mexer no Arduino

() Sim () Não () Mais ou menos

Eu gostaria de continuar a mexer no Arduino

() Sim () Não () Mais ou menos

O que você achou mais difícil no Arduino?

R: ____

Figura 11 – Questionário de Satisfação

Fonte: autores

RESULTADOS E DISCUSSÕES

Nesta seção, serão apresentados o público-alvo desse curso, a satisfação dos estudantes e a avaliação subjetiva dos professores sobre o curso.

Público-alvo

Participaram dos encontros 10 alunos (quatro meninas e seis meninos) que já tinham passado pelo primeiro curso de Pensamento Computacional, com 20 horas ao todo, em que aprenderam noções básicas de programação com Scratch para a criação de histórias e animações. Essas crianças são do 4º ano, 5º ano e 6º ano do Colégio Presbiteriano Mackenzie, sendo que dois alunos estão com 9 anos, quatro com 10 anos e quatro com 11 anos.

As Figuras 12 e 13 apresentam os resultados relacionados à fluência digital desses estudantes.

Quais tecnologias você já usou?

10 responses

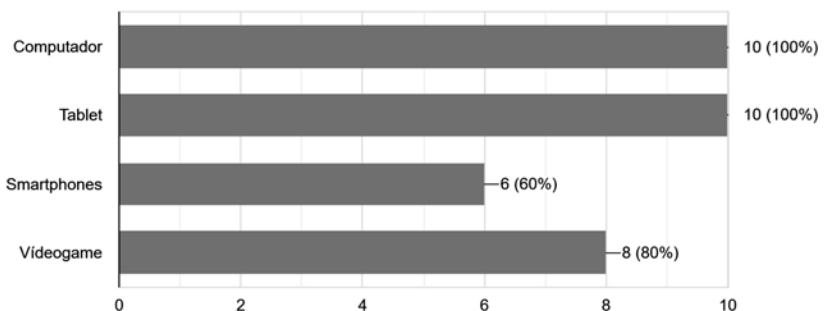


Figura 12 – Questão em relação ao uso de tecnologias

Fonte: autores

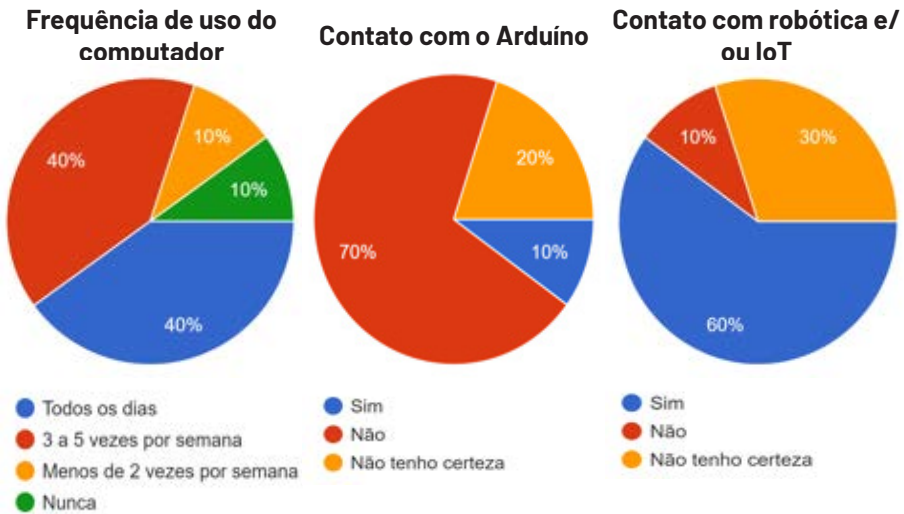


Figura 13 – Resultados das questões em relação à frequência de uso do computador, contato com o Arduino e contato com robótica e/ou IoT

Fonte: autores

Quantas vezes por semana você usa o computador?

10 responses

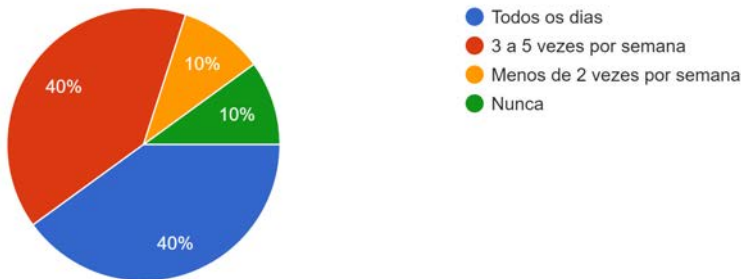


Figura 14 – Quantas vezes por semana você usa o computador?

Fonte: autores

Antes da aula, você sabia o que era Arduino?

10 responses

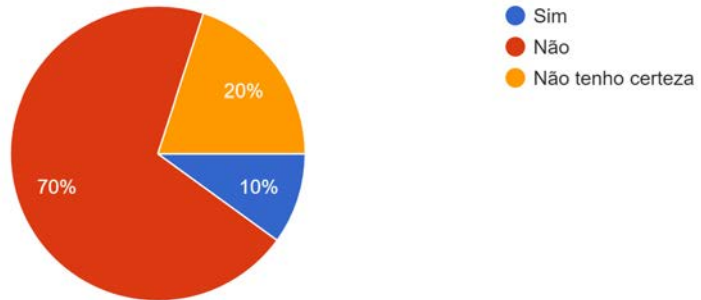


Figura 15 – Antes da aula, você sabia o que era Arduino?

Fonte: autores

Antes da aula, você já tinha ouvido falar de temas como robótica ou internet das coisas?

10 responses

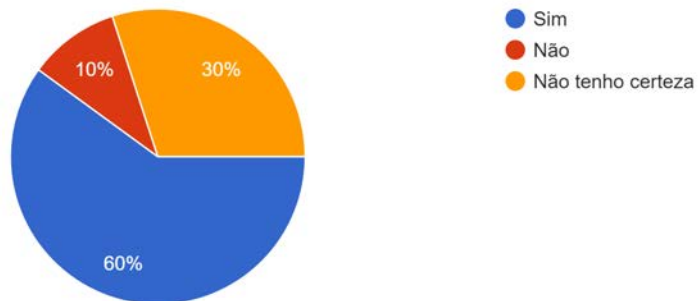


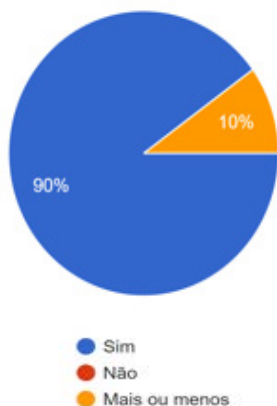
Figura 16 – Antes da aula, você já tinha ouvido falar de temas como robótica ou internet das coisas?

Fonte: autores

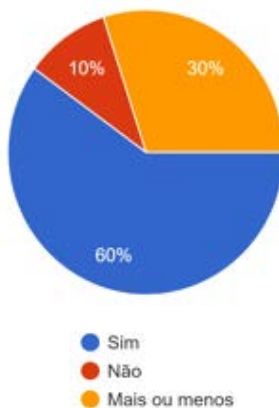
Satisfação dos Estudantes

Quando perguntados sobre a aula que fizeram com Scratch (Pyonkee) e Arduino, os estudantes apresentaram boa satisfação, como apresentado na Figura 14.

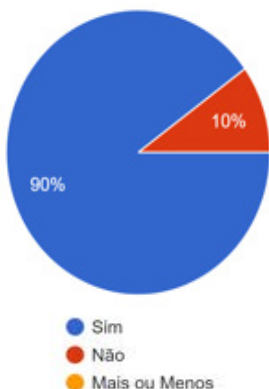
1. Gostaria de aprender mais Scratch (Pyonkee)



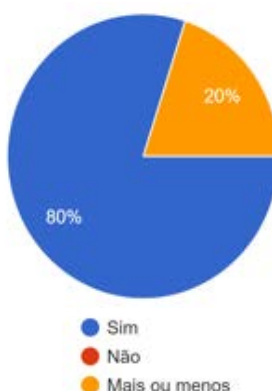
2. Foi fácil criar jogos com Scratch (Pyonkee)



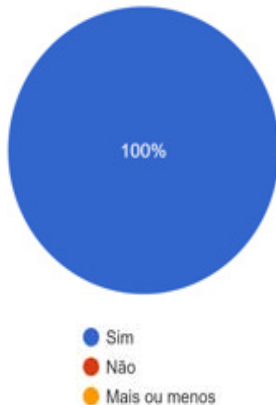
3. Prefiro usar o computador para trabalhar com Scratch (Pyonkee) do que iPad.



4. Gostaria de usar o Scratch (Pyonkee) para fazer outros jogos.



5. Gostei de usar o Arduino



6. Gostaria de continuar usando Arduino



Figura 17 – Resultados das questões em relação à satisfação os alunos como o uso do Scratch (Pyonkee) e Arduino

Fonte: autores

Além das questões fechadas, foram acrescentadas perguntas abertas para coletar opiniões dos estudantes acerca do curso (Figura 15)

1 - O que você achou mais fácil na ferramenta Scratch (Pyonkee)?

- Tudo
- Programação da música e personagens
- Mexer no Tablet e Computador
- Tudo é fácil
- Fazer o boneco andar
- Mover os personagens
- Achei mais fácil criar jogos
- Montar o Arduino
- Achei a parte mais fácil de montar

2 - O que você achou mais difícil na ferramenta Scratch (Pyonkee)?

- Nada
- Nada
- Encaixar as peças
- Fazer o space invaders
- Não achei nada difícil
- Furar o dedo nos fios
- Arrastar os códigos
- Não sei dizer

3 - O que achou mais difícil no Arduino?

- Conexões dos cabos
- Os fios de ligação
- Colocar as peças
- Nada é difícil
- Mais ou menos
- Nada
- Não achei nada difícil
- Não sei
- Nada

Figura 18 – Questões abertas apresentadas aos estudantes

Fonte: autores

Eu gostaria de aprender a mexer no Scratch um pouco mais.

10 responses

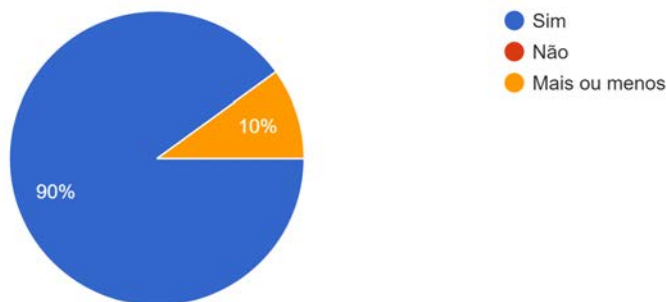


Figura 19 – Eu gostaria de aprender a mexer no Scratch um pouco mais

Fonte: autores

Foi fácil aprender a mexer no Scratch para fazer jogos?

10 responses

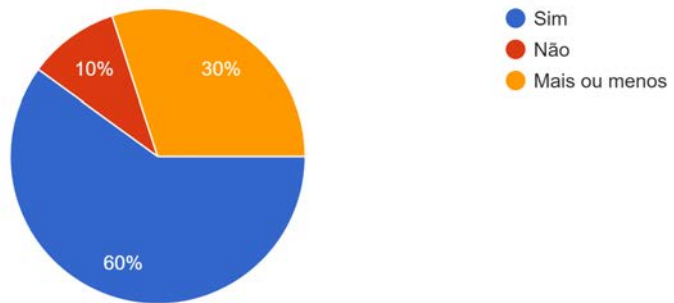


Figura 20 – Foi fácil aprender a mexer no Scratch para fazer jogos?

Fonte: autores

Eu acho melhor mexer com Scratch no computador do que no tablet.

10 responses

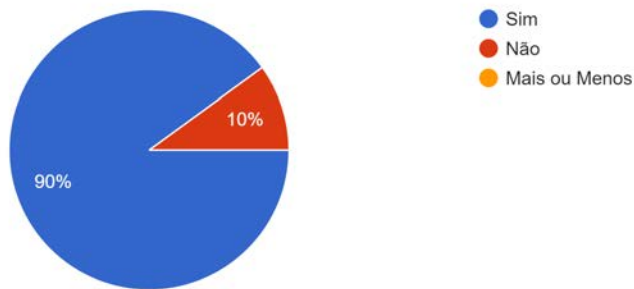


Figura 21 – Eu acho melhor mexer com Scratch no computador do que no tablet

Fonte: autores

Eu gostaria de continuar a mexer no Scratch para fazer mais jogos

10 responses

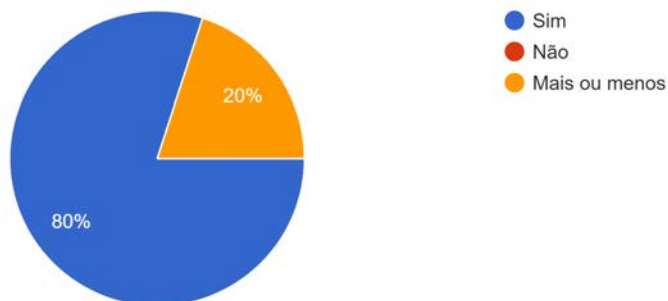


Figura 22 – Eu gostaria de continuar a mexer no Scratch para fazer mais jogos

Fonte: autores

O que você achou mais fácil no Scratch?

10 responses

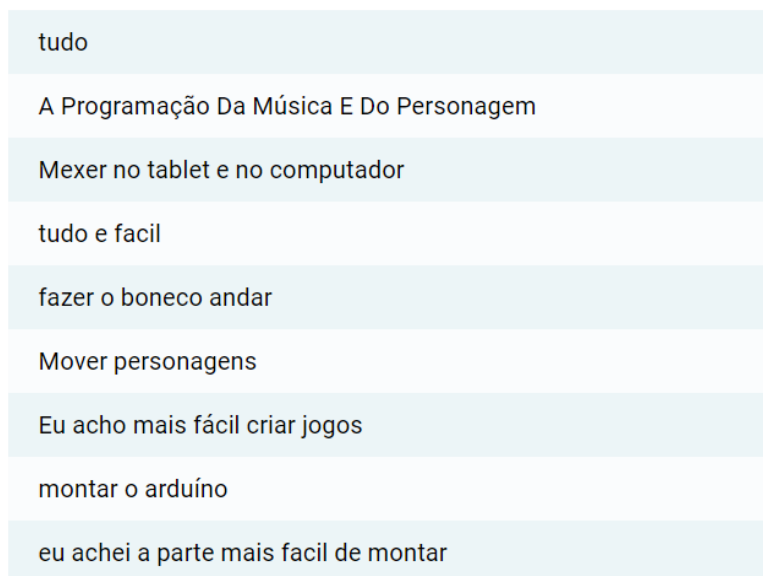


Figura 23 – O que você achou mais fácil no Scratch?

Fonte: autores

O que você achou mais difícil no Scratch?

10 responses



Figura 24 – O que você achou mais difícil no Scratch?

Fonte: autores

Eu gostei de aprender a mexer com Arduino.

10 responses



Figura 25 – Eu gostei de aprender a mexer com Arduino

Fonte: autores

Eu gostaria de continuar a mexer no Arduino

10 responses

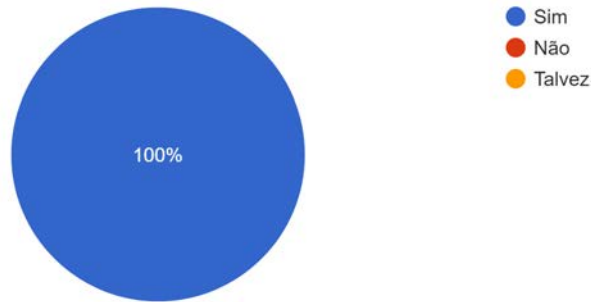


Figura 26 – Eu gostaria de continuar a mexer no Arduino

Fonte: autores

O que você achou mais difícil no Arduino?

10 responses

As Conexões Dos Cabos
os fios de ligação
Colocar as peças
nada e difícil
mais ou menos
Nada.
não acho nada difícil
Eu não sei
nada
as partes de conectar os fios

Figura 27 – O que você achou mais difícil no Arduino?

Fonte: autores

Por meio das respostas coletadas nos dois questionários, é possível apontar que:

1. Os estudantes já têm um bom contato com tecnologia, 100% deles já usam computadores e tablets, sendo que 80% deles já os utilizam todos os dias ou a maioria dos dias da semana.
2. Embora os estudantes usem *tablets* na escola e em casa com muita frequência, eles não gostaram de programar utilizando *tablets*. Como já tinham feito o primeiro ciclo do curso em computadores, conseguiram comparar as facilidades e perceberam que o computador era mais adequado para programação.
3. Os alunos sentiram-se motivados com a criação de jogos com Scratch (Pyonkee) e gostariam de construir mais jogos e estudar mais a linguagem.
4. Os alunos aprovaram com unanimidade a aula de robótica com Arduino e também gostariam de continuar a estudar o assunto, mostrando o quanto a aula foi motivadora para eles.
5. Muitos também acharam que utilizar Scratch (Pyonkee) é fácil ou não difícil. O mesmo para o Arduino, embora só tenham tido uma aula.

Avaliação Subjetiva dos Professores

Na opinião dos professores que têm preparado e ministrado o curso para os estudantes, percebe-se:

- Os estudantes são altamente motivados a fazer o curso.
- É necessário o envolvimento de mais professores e bolsistas para ministrar esses cursos, pois os estudantes demandam muita atenção, querendo aprender mais do que está sendo ensinado no momento.

- Os estudantes são motivados pelo caráter lúdico da ferramenta e do tipo de atividades, sendo importante conseguirem visualizar suas progressões assim que vão construindo seus exercícios. O mesmo se refere ao uso de Arduino.
- Os estudantes são altamente curiosos em relação às ferramentas, fazendo com que tenham um aprendizado também pela experimentação.

CONCLUSÕES

A inserção de Tecnologias de Informação e Comunicação nas escolas pode levar a resultados bastante promissores para o aumento da motivação dos estudantes, melhoria do raciocínio lógico e transformador para o processo de ensino-aprendizagem, se trabalhado adequadamente.

Este capítulo apresentou estudos sobre um curso de Pensamento Computacional utilizando Scratch (Pyonkee) para o desenvolvimento de jogos, assim como programação de Arduino, introduzindo o conceito de Internet das Coisas a alunos do ensino fundamental I, em que a principal contribuição deste estudo foi o desenvolvimento do protagonismo estudantil no desenvolvimento de projetos de jogos e internet das coisas.

Com base nos resultados apresentados neste capítulo, é possível concluir que, para esse grupo de 10 estudantes do ensino fundamental I do Colégio Presbiteriano Mackenzie, foi perfeitamente possível motivá-los com aulas de Pensamento Computacional, a partir do desenvolvimento de jogos e programação de microcontroladores (Arduino), ainda que o público-alvo fosse constituído por crianças ainda bastante novas.

Como trabalhos futuros, pretende-se desenvolver um curso inteiro de programação de Arduino focando robótica ou sensoramento de ambiente voltada para Internet das Coisas, sendo o ciclo 3 do projeto, respondendo a uma demanda de pais e estudantes.

AGRADECIMENTOS

O presente estudo foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de nível superior – Brasil (Capes) – Programa de Excelência (Proex 0653/2018). Agradecemos ao Prof. Dr. Pedro Cacique e aos alunos Matheus Gois, Kleber Yoshida, Wolfgang Wander e Renato Ribeiro pelo apoio no desenvolvimento das atividades.

REFERÊNCIAS

Ashton, K. (2009). *RFID Journal*. 22 de 06 de 2009. Recuperado em 2018 de <http://www.rfidjournal.com/articles/view?4986>.

Autodesk, Inc (s/d). Recuperado de <https://www.tinkercad.com/>.

Barcelos, T. S., & Silveira, I. F. (2012). Pensamento computacional e educação matemática: Relações para o ensino de computação na educação básica. *Anais do XXXII CSBC. XX Workshop sobre Educação em Computação*, Curitiba. v. 2, p. 23.

Barnaghi, P., Wang, W., Henson, C., & Taylor, K. (2012). Semantics for the Internet of Things: Early progress and back to the future. *Proceedings of the International Journal on Semantic Web and Information Systems (IJSWIS)*.

Bezerra, F. D. L., & Dias, K. (2014). Programação de computadores no ensino fundamental: experiências com logo e scratch em escola pública. *Proceedings of XXXIV Congresso da Sociedade Brasileira de Computação – CSBC 2014*.

BRASIL. Ministério da Educação. Governo Federal. *Base Nacional Curricular*. Recuperado em 29 jan. 2019, de http://basenacionalcomum.mec.gov.br/wp-content/uploads/2018/12/BNCC_19dez2018_site.pdf.

Fernandez, C. O., Biazon, L. C, Martinazzo, A. A. G., Ficheman, I. K., & Lopes, R. D. (2015). Uma proposta baseada em projetos para oficinas de Internet das Coisas com Arduino voltadas a estudantes do Ensino Médio. *RENOTE – Revista Novas Tecnologias na Educação*, 13, n. 2 .

Fleury, A., Nakano, D. N., Sakuda, L. O., & Cordeiro, J. H. (2015). Jogos Digitais: Desafios da competitividade nas indústrias do século XXI. *Revista da ESPM*, (5), 132-139.

IDC. International Data Corporation (2017). 7 dez. 2017. Recuperado em 2018 de <https://www.idc.com/getdoc.jsp?containerId=prUS43295217>.

Kalil, F., Hernandez, H., Antunez, M. F., Oliveira, K., Ferronato, N., & SANTOS, M. R. (2013). Promovendo a robótica educacional para estudantes do ensino médio público do Brasil. *Nuevas Ideas en Informática Educativa*, TISE, p. 739-742.

Kyriazis, D., Varvarigou, T., White, D., Rossi, A., & Cooper, J. (2013). Sustainable smart city IoT applications: Heat and electricity management & Eco-conscious cruise control for public transportation. *IEEE 14th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, p. 1 - 5

Maloney, J. H., Peppler, K., Kafai, Y., Resnick, M., & Rusk, N. (2008, March). Programming by choice: urban youth learning programming with scratch. *ACM SIGCSE Bulletin*, v. 40, n. 1, pp. 367-371. ACM.

Margolis, J. & Fisher, A. (2002). *Unlocking the Clubhouse: Women in Computing*. MIT Press.

Martins, V. F., Concilio, I. A. S., & Guimarães, M. P. (2018). Problem based learning associated to the development of games for programming teaching. *Computer Applications in Engineering Education*, 26(5), p. 1577-1589.

McCray, R. A., DeHann, R. L., & Schuck, J. A. (2003). *Editors, Improving Undergraduate Instruction in Science, Technology, Engineering, and Mathematics: Report of a Workshop*, Editors, The National Academies Press, Washington DC.

Oliveira, M. D., Souza, A. D., Ferreira, A., & Barbosa, E. F. S. B. (2014). Ensino de lógica de programação no ensino fundamental utilizando o Scratch: um relato de experiência. *XXXIV Congresso da SBC-XXII Workshop de Ensino de Computação*, Brasília.

Pereira, A. I., Pacheco, M. F., & Fernandes, F. P. (2016). Jogos matemáticos como ferramenta para motivar os estudantes para aprender matemática. *VII Congresso Mundial de Estilos de Aprendizagem*. p. 3029-3036.

Pyonkee. Recuperado em 24 nov. 2018, de <https://www.softumeya.com/pyonkee/en/>.

Rodrigues, B. S., Lima, B. L. S., & Oliveira, V. I. (2018). Desenvolvimento de Modelo Robótico Controlado Por Arduino. In Kawamoto, A. L. S., Correa, A. G. D., & Martins, V. F. (Org.). *I Jornada Latino-Americana de Atualização em Informática*. Porto Alegre, Sociedade Brasileira de Computação, p. 337-359.

Rodriguez, C., Zem-Lopes, A. M., Marques, L., & Isotani, S. (2015). Pensamento Computacional: transformando ideias em jogos digitais usando o Scratch. *Anais do Workshop de Informática na Escola*, 21(1), p. 62.

Rosa, C. T. W., & Trentin, M. A. S. (2017). Olimpíada de robótica educativa livre: potencialidades para a educação científica e tecnológica. *Enseñanza de las ciencias*. Revista de investigación y experiencias didácticas, p. 5543-5549.

Savi, R., Von Wangenheim, C. G., Ulbricht, V., & Vanzin, T. (2010). Proposta de um modelo de avaliação de jogos educacionais. *Renote*, 8(3).

Scaico, P. D., Lima, A. A., Azevedo, S., Silva, J. B. B., Raposo, E. H., Alencar, Y., & Scaico, A. (2013). Ensino de programação no ensino médio: Uma abordagem orientada ao design com a linguagem scratch. *Revista Brasileira de Informática na Educação*, 21(02), p. 92.

Scratch - Imagine, Program, Share. Recuperado em 29 jan. 2019, de <https://scratch.mit.edu/>.

Severance, C. (2014). Massimo Banzi: Building Arduino. *IEEE: Computer*, 47(1) p. 11-12.

Weiser, M. (1991). The Computer for the 21st Century. *Scientific American*, 265(3), p. 94-105.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, v. 49, n. 3, p. 33-35.

Wolz, U., Maloney, J., & Pulimood, S. M. (2008, March). 'scratch'your way to introductory cs. *ACM SIGCSE Bulletin*, v. 40, n. 1, pp. 298-299. ACM.

CAPÍTULO 4

DETECT – UM MODELO PARA A AVALIAÇÃO DE UNIDADES INSTRUCIONAIS PARA O ENSINO DE COMPUTAÇÃO NA EDUCAÇÃO BÁSICA

Christiane Gresse von Wangenheim

Giani Petri

Adriano F. Borgatto

INTRODUÇÃO

A inclusão do ensino de computação na educação básica é uma tendência mundial que vem se concretizando de diversas maneiras, inclusive informais, como acampamentos de verão, clubes ou em oficinas de pais e filhos (von Wangenheim & von Wangenheim, 2014). Atualmente, existem diversas iniciativas para ensinar o pensamento computacional por meio da programação, como, por exemplo, Code.org (<http://www.code.org>), Code.club (<https://www.codeclubworld.org>), Computação na Escola (<http://www.computacaonaescola.ufsc.br/>), entre outras. Com essas iniciativas, espera-se a popularização do pensamento computacional, e, ao mesmo tempo, a conscientização e o interesse dos alunos sobre a área de computação (Guzdial, 2014; Garneli *et al.*, 2015).

Nesse contexto, levando em consideração o número crescente de alternativas de ambientes de programação e unidades instrucionais (UIs), é importante que se tenha evidências dos benefícios esperados dessas iniciativas, para então selecioná-las e adotá-las de forma sistemática (Decker *et al.*, 2016). Segundo Guzdial (2004), as maiores contribuições a serem feitas nessa área não estão na construção de novos ambientes/linguagens de programação, mas em descobrir como estudar os ambientes existentes. Uma compreensão mais precisa a respeito dos resultados do uso dos ambientes de programação permitiria saber se eles contribuem, de fato, positivamente ao alcance dos seus objetivos e se compensam os custos e esforços envolvidos em adotá-los. No entanto, embora se tenha indícios de que as UIs existentes possam ser ferramentas capazes de aprimorar o processo de ensino e aprendizagem na educação básica, e que esse tipo de iniciativa começa a ser usada mais amplamente nas escolas mundialmente, ainda há poucas pesquisas voltadas à

análise da contribuição que essas Uls podem trazer para a educação (Decker *et al.*, 2016).

Atualmente, a avaliação da qualidade dessas Uls é limitada e, por vezes, inexistente (Decker *et al.*, 2016; Garneli *et al.*, 2015). Em muitos casos, a decisão de se utilizar uma dessas Uls é baseada em suposições de sua eficácia (Gross & Powers, 2005; Wilson *et al.*, 2010). Por outro lado, alguns estudos focam exclusivamente na avaliação do grau de aprendizagem alcançado (Gross & Powers, 2005; Kalelioglu & Gülbahar, 2014), inclusive a conceitos específicos de computação, como estruturas de controle (Mühling *et al.*, 2015). Outros estudos focam em perguntas específicas, como, por exemplo, a efetividade de linguagens de programação baseada em blocos (Weintrop & Wilensky, 2015; Grover *et al.*, 2014; Perdikuri, 2014). No entanto as percepções e intenções dos alunos também são fatores determinantes para o sucesso da aprendizagem (Giannakos *et al.*, 2013). Porém poucas avaliações levam também em consideração outros aspectos, como a motivação e a experiência do aluno durante a unidade instrucional (Craig & Horton, 2009; Giannakos *et al.*, 2014), ou ainda, as atitudes dos estudantes em relação à aceitação da tecnologia (Giannakos *et al.*, 2013). Nesse contexto, há trabalhos que apresentam escalas para mensurar a atitude dos alunos em relação à computação (Garland & Noyes, 2008). No entanto essas escalas são projetadas para a educação superior e parecem ser ultrapassadas no contexto atual de ensino de computação na educação básica (Garland & Noyes, 2008).

As medições utilizadas para avaliar a qualidade das Uls de ensino de pensamento computacional variam bastante, desde escalas genéricas para avaliar as atitudes dos alunos em relação à computação até instrumentos de medição desenvolvidos de maneira *ad-hoc*. Muitos são desenvolvidos sem a definição de um modelo para derivar os itens do instrumento de medição com base em constructos teóricos, o que pode deixar a validade dos resultados questionável. Desse modo, atualmente, há uma falta de modelos de avaliação e/ou instrumentos de medição sistematicamente desenvolvidos e avaliados que sejam amplamente aceitos para avaliar a qualidade de Uls para o ensino de pensamento computacional na educação básica.

Contudo, principalmente quando as UIs são realizadas de maneira mais informal, por exemplo, em oficinas de ensino de programação para pais e filhos fora do ambiente escolar, pode ser inviável a realização de experimentos que necessitem da realização de pré-testes e de inclusão de grupos de controle, pois pode causar uma maior interrupção e pode influenciar o grau de diversão propiciado pela oficina. Diante disso, no contexto de UIs mais informais, uma alternativa viável pode ser a condução de estudos de caso, em que a avaliação da UI é realizada somente ao final da oficina (pós-teste), tipicamente por meio de um questionário para obter a percepção dos alunos (Wohlin *et al.*, 2012). Uma vantagem desse tipo de estudo é que a avaliação pode ser executada com pouco esforço e de uma forma não intrusiva ao final da unidade instrucional. Estudos baseados na medição de percepções, utilizando instrumentos de medição, tipicamente questionários, são conduzidos em uma variedade de diferentes áreas de pesquisa (DeVellis, 2016; Takatalo *et al.*, 2010; Sweetser & Wyeth, 2005; Poels *et al.*, 2007). Assim, o objetivo deste capítulo é apresentar o desenvolvimento e avaliação do modelo *Evaluating TEaching CompuTing* (dTECT), um modelo para a avaliação da qualidade de unidades instrucionais focadas no ensino de computação na educação básica com base na percepção dos alunos.

METODOLOGIA

Com o objetivo de desenvolver um modelo de avaliação de unidades instrucionais para o ensino de computação, foi realizada uma pesquisa aplicada (Miller & Salkind, 2002), dividida em quatro etapas (Figura 1).

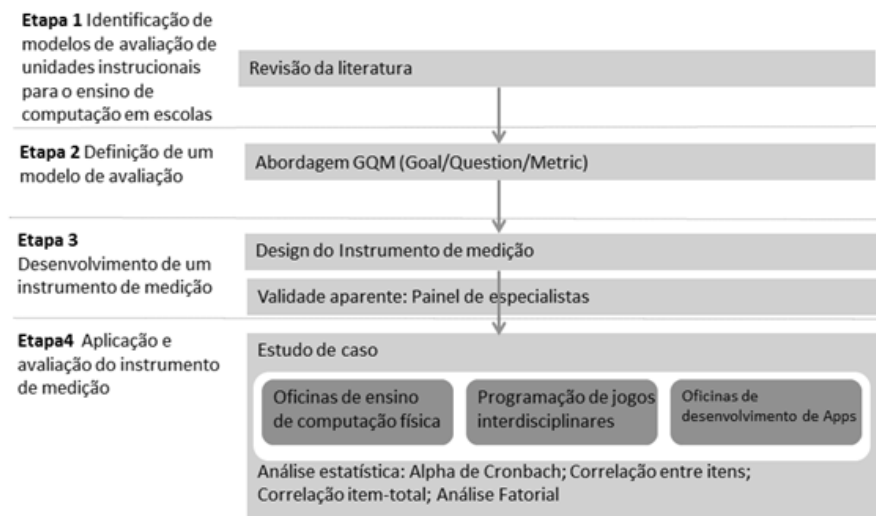


Figura 1 – Metodologia de pesquisa

Fonte: autores

Etapa 1. Levantamento do estado da arte. Foi realizada uma revisão da literatura sobre a bibliografia relacionada a modelos de avaliação de unidades instrucionais para o ensino de computação nas escolas.

Etapa 2. Definição de um modelo de avaliação. O modelo de avaliação dETECT foi desenvolvido para medir a qualidade das unidades instrucionais utilizando a abordagem *Goal/Question/Metric* (GQM) (Basili et al., 1994), sistematicamente definindo o objetivo da avaliação e decompondo-o em questões de análise e medidas.

Etapa 3. Definição do instrumento de medição. De modo a operacionalizar a medição, um questionário foi desenvolvido por uma equipe multidisciplinar, composta por pesquisadores sêniores nas áreas de computação e estatística (DeVellis, 2016; Krosnick & Presser, 2010; Malhotra, 2008; Kasunic, 2005). A definição dos itens do questionário é baseada em itens de estudos semelhantes que foram encontrados na literatura, considerados aderentes ao contexto do presente estudo e ao plano de medição definido.

Etapa 4. Aplicação e avaliação do instrumento de medição. Foi realizado um estudo de caso (Yin, 2009; Wohlin et al., 2012) para avaliar o

instrumento de medição em termos de confiabilidade e validade de constructo. Para a definição da avaliação, foi utilizada a abordagem GQM. O objetivo do estudo foi decomposto em aspectos de qualidade e questões de análise para serem analisados na avaliação do instrumento de medição (Carmines & Zeller, 1979; DeVellis, 2016; Trochim & Donnelly, 2008). Durante o estudo de caso, foi aplicado o instrumento de medição do modelo dETECT como parte da avaliação de 16 cursos/oficinas de computação realizados em diferentes instituições de ensino.

A pesquisa foi aprovada pelo Comitê de Ética em Pesquisa com Seres Humanos (CEPSH) da UFSC, conforme parecer n.º 1.021.541.

O MODELO DETECT (EVALUATING TEACHING COMPUTING)

O objetivo do modelo de avaliação dETECT é: analisar unidades instrucionais com o propósito de avaliar a percepção da qualidade em termos de qualidade da UI, experiência de computação e percepção da aprendizagem do ponto de vista de alunos no contexto de UIs focadas no ensino de computação na educação básica. A partir desse objetivo, são derivadas as perguntas de análise e as medidas com base na literatura (Figura 2).

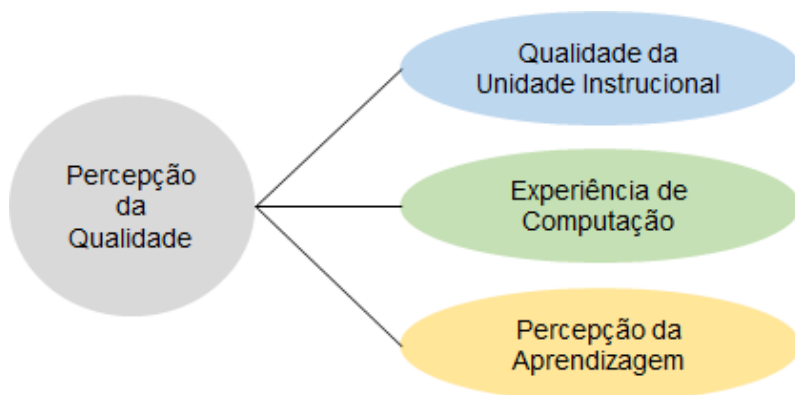


Figura 2 – Decomposição dos fatores de qualidade

Fonte: autores

Entende-se, neste estudo, que uma unidade instrucional (oficina, curso etc.) de qualidade é aquela que atinge os seus objetivos de aprendizagem, promove atividades prazerosas que facilitam a aprendizagem e que desperta uma percepção positiva e interesse referente à área de computação.

A medição é operacionalizada pelo desenvolvimento de um instrumento de medição (questionário) a ser respondido pelos alunos após o término da unidade instrucional, de modo a obter a sua percepção sobre qualidade da oficina/curso. Os itens que compõem o questionário (Figura 3) são derivados de estudos similares encontrados na literatura, considerados aderentes ao contexto desta pesquisa.

N.º	Descrição do Item	Formato das respostas
-----	-------------------	-----------------------

Qualidade da Unidade Instrucional

- | | | |
|---|---------------------------|--|
| 1 | A oficina foi: | (1) Muito Divertida
(2) Divertida
(3) Chata
(4) Muito chata |
| 2 | O tempo das aulas passou: | (1) Muito rápido
(2) Rápido
(3) Devagar
(4) Muito devagar |
| 3 | A oficina foi: | (1) Excelente
(2) Boa
(3) Regular
(4) Ruim |

Experiência de Computação

- | | | |
|---|---|--------------------|
| 4 | Vou mostrar meu programa de computador para outras pessoas: | (1) Sim
(2) Não |
| 5 | Quero aprender mais sobre como fazer programas de computador: | (1) Sim
(2) Não |

- | | | |
|---|---|---------------------|
| | | (1) Muito Divertido |
| | | (2) Divertido |
| 6 | Fazer um programa de computador é: | (3) Chato |
| | | (4) Muito chato |
| 7 | Eu gosto de fazer programas de computador: | (1) Sim |
| | | (2) Não |
| 8 | A computação é útil no dia a dia: | (1) Sim |
| | | (2) Não |
| 9 | Você já pensou em trabalhar com computação? | (1) Sim |
| | | (2) Não |

Percepção da Aprendizagem

- | | | |
|----|--|-------------------|
| | | (1) Muito fácil |
| | | (2) Fácil |
| 10 | A oficina foi: | (3) Difícil |
| | | (4) Muito Difícil |
| 11 | Eu consigo fazer programas de computador: | (1) Sim |
| | | (2) Não |
| 12 | Consigo explicar para um amigo/amiga como fazer um programa de computador: | (1) Sim |
| | | (2) Não |
| | | (1) Muito fácil |
| | | (2) Fácil |
| 13 | Fazer um programa de computador é: | (3) Difícil |
| | | (4) Muito Difícil |

Figura 3 – Instrumento de medição

Fonte: autores

O material completo do modelo dETECT está disponível em http://www.computacaonaescola.ufsc.br/?page_id=45.

DEFINIÇÃO E EXECUÇÃO DA AVALIAÇÃO DO MODELO DETECT

Ao desenvolver modelos de avaliação e questionários, é fundamental analisar se eles estão medindo o que se pretende medir (validade de constructo) e se o mesmo processo de medição produz os mesmos resultados (confiabilidade) (Carmines & Zeller, 1979). Por isso, foi avaliado o instrumento de medição do modelo dTECT em termos de confiabilidade e validade de constructo sob o ponto de vista de pesquisadores no contexto de unidades instrucionais para o ensino de computação na educação básica. São analisadas as seguintes questões de análise:

Confiabilidade

QA1: Há evidência de consistência interna no instrumento de medição do modelo dTECT?

Validade de Constructo

QA2: Há evidência de validade convergente e discriminante no instrumento de medição do modelo dTECT?

QA3: Quantos fatores subjacentes influenciam as respostas dos itens do instrumento de medição do modelo dTECT?

Para a avaliação do modelo dTECT, foram executados 16 estudos de caso aplicando três diferentes unidades instrucionais em 13 diferentes instituições de ensino entre 2015 e 2016, totalizando 477 alunos envolvidos com faixa etária entre 6 e 18 anos (Tabela 1).

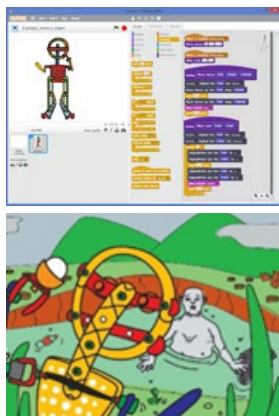
Unidade Instrucional	Local/Data	Número de alunos
Oficina de Computação Física	INE/UFSC - Florianópolis - 08/08/2015	8
	Escola Hamônia - Ibirama/SC - 29/08/2015	13
	INE/UFSC - Florianópolis - 17/10/2015	14
	IFSC - Gaspar/SC - 20 e 22/10/2015	32
	Escola Sabedoria Junior - Florianópolis/SC - 04/11/2015	22
	INE/UFSC - Florianópolis - 07/11/2015	16
	INE/UFSC - Florianópolis - 14/11/2015	15
Unidade Jogos com Scratch	Turmas 5Mat, 5 Vesp, 7A, 7B - Escola Autonomia - Florianópolis/SC	99
	Escola Básica Municipal Prefeito Reinaldo Weingartner - Palhoça/SC	25
	EEB Prof Vitorio Anacleto Cardoso - Gaspar/SC	43
	EEB Zenaide Schmitt Costa - Gaspar/SC	31
	EEB Luiz Franzoi - Gaspar/SC	15
	EEB Ferandino Dagnoni - Gaspar/SC	46
	EEB Prof Dolores dos Santos Krauss - Gaspar/SC	14
	EEB Norma Mônica Sabel - Gaspar/SC	49
Oficina de App Inventor	Turma 1 e 2 - Escola Básica Prof.ª Herondina Medeiros Zeferino 2016	35
Total		477

Tabela 1 - Visão geral das aplicações e dados coletados

Fonte: autores

O público-alvo são alunos no nível da educação básica levando em consideração diversos tipos de unidades instrucionais, tanto dentro de escolas quanto de forma extracurricular no Brasil. As unidades instrucionais são voltadas ao ensino de computação focando na programação e pensamento computacional (Figura 4).

Oficina de Computação Física



Integrando Scratch/ Snap! com Arduino e peças de hardware numa solução baixo custo, os alunos aprendem dar vida a um robô interativo.



Unidade Jogos com Scratch



De forma interdisciplinar os alunos aprendem conceitos básicos da computação programando jogos envolvendo diversos conteúdos (p.ex. história, língua portuguesa, geografia, etc.) utilizando Scratch.



Oficina de App Inventor



Os alunos aprendem programar um app de jogo “Caça Mosquito” utilizando o App Inventor.



Figura 4 – Visão geral das unidades instrucionais aplicadas

Fonte: autores

Análise dos Dados

De modo a obter uma maior precisão e poder estatístico, aumentando o tamanho da amostra, os dados coletados nos 16 estudos de caso foram analisados cumulativamente para responder às questões de análise definidas.

Confiabilidade

QA1: Há evidência de consistência interna no instrumento de medição do modelo dTECT?

A consistência interna foi medida por meio do Alfa de Cronbach (1951). O coeficiente alfa indica indiretamente o grau em que um conjunto de itens está medindo um único fator de qualidade (a percepção da qualidade da unidade instrucional). Tipicamente, valores de alfa de Cronbach maiores que 0,7 são considerados aceitáveis (DeVellis, 2016), assim indicando a consistência interna do instrumento de medição. Analisando os 13 itens que compõem o instrumento de medição do modelo dTECT (Tabela 1), o valor do coeficiente alfa de Cronbach é aceitável ($\alpha=.787$). Desse modo, pode-se concluir que as respostas entre os itens são consistentes e precisas, indicando a confiabilidade dos itens do instrumento de medição do modelo dTECT.

Validade de Constructo

QA2: Há evidência de validade convergente e discriminante no instrumento de medição do modelo dTECT?

Validade de constructo de um instrumento de medição refere-se à habilidade de medir o que ele se propõe a medir (Carmines & Zeller, 1979; Trochim & Donnelly, 2008). Validade convergente e discriminante são dois subtipos de validade que compõem a validade de constructo. A validade convergente refere-se ao grau em que dois itens de um fator de qualidade que teoricamente devem estar relacionados, de fato, estão relacionados. Em contrapartida, validade discriminante testa se conceitos ou medidas que supostamente não devem ter relação, de fato, não estão relacionados (Tro-

chim & Donnelly, 2008). De modo a analisar a validade convergente e discriminante do instrumento de medição do modelo dTECT, as correlações dos itens e correlação item-total são calculadas (DeVellis, 2016).

Correlação dos itens. Para analisar a correlação entre os itens, utilizou-se uma matriz de correlação não paramétrica de Spearman (Tabela 4). A matriz mostra o coeficiente de correlação de Spearman, indicando o grau de correlação entre dois itens (par de itens). Esse coeficiente foi utilizado por ser o coeficiente mais apropriado para análises de escalas Likert. Os coeficientes de correlação entre os itens de um mesmo fator de qualidade estão marcados em cores. De acordo com Cohen (1988), uma correlação entre itens é considerada satisfatória se o coeficiente de correlação é maior que 0,29, indicando que há uma média ou alta correlação entre os itens. Coeficientes de correlação satisfatórios estão destacados em negrito.

Item/ Fator de qualidade	1	2	3	4	5	6	7	8	9	10	11	12	13
	Qualidade da UI			Experiência de Computação						Percepção da Aprendizagem			
1	1,000												
2	,268	1,000											
3	,514	,294	1,000										
4	,243	,090	,230	1,000									
5	,291	,137	,295	,362	1,000								
6	,452	,174	,382	,266	,385	1,000							
7	,211	,160	,181	,325	,378	,395	1,000						
8	,046	,028	,035	,240	,327	,312	,345	1,000					
9	,221	,107	,262	,203	,324	,226	,263	,197	1,000				
10	,174	,203	,153	,067	,133	,144	,022	-,086	,125	1,000			
11	,213	,059	,087	,273	,320	,318	,287	,239	,202	,157	1,000		
12	,162	,094	,135	,312	,246	,194	,266	,192	,247	,143	,450	1,000	
13	,156	,082	,155	,168	,266	,325	,225	,226	,199	,391	,432	,286	1,000

Tabela 2 – Matriz de correlação de Spearman

Fonte: autores

Analisando as correlações entre os itens dos três fatores de qualidade (Tabela 2), pode-se observar que a maioria dos pares de

itens possui média ou alta correlação considerando cada fator de qualidade. Porém alguns pares de itens apresentaram uma baixa correlação (ex., 1-2, 6-9, 10-11). Mesmo assim, os resultados indicam uma evidência de validade convergente. Por outro lado, alguns pares de itens (ex., 1-6, 3-6, 5-11) apresentaram média ou alta correlação com itens de outro fator de qualidade. Desse modo, não há evidência de validade discriminante. No entanto a falta de existência de validade discriminante é aceitável, pois, embora o modelo seja dividido em três fatores de qualidade, todos os fatores também estão relacionados a um único fator, que é a percepção da qualidade da UI.

Correlação item-total. A correlação item-total é analisada de modo a verificar se qualquer item do instrumento é inconsistente com a média de correlação dos outros itens, e, assim, pode ser eliminado (Carmines & Zeller, 1979; DeVellis, 2016). A Tabela 5 mostra o coeficiente de correlação entre um único item e os outros itens (excluindo o item referido) do instrumento de medição. As correlações são consideradas satisfatórias se o coeficiente de correlação é maior que 0,29 (Cohen, 1988). Itens que apresentam uma baixa correlação são destacados em negrito. A Tabela 3 também apresenta o coeficiente alfa de Cronbach se o item for excluído, nesse caso, espera-se que nenhum item ao ser retirado aumente o alfa de Cronbach (DeVellis, 2016) do instrumento de medição.

Fator de Qualidade	N.º item	Correlação de item total corrigida	Alfa de Cronbach se o item for excluído
Qualidade da UI	1	,511	,764
	2	,269	,794
	3	,459	,769
Experiência de Computação	4	,431	,773
	5	,511	,769
	6	,594	,753
	7	,481	,771
	8	,338	,783
	9	,410	,774

Percepção da Aprendizagem	10	,280	,787
	11	,470	,770
	12	,415	,773
	13	,474	,769

Tabela 3 – Correlação item-total corrigida

Fonte: autores

Em geral, as correlações item-total são médias e altas. A maioria dos itens demonstra aceitável correlação item-total e valores satisfatórios de coeficiente alfa de Cronbach se o item for excluído, indicando, assim, a validade dos fatores de qualidade. Somente os itens 2 (“O tempo das aulas passou”) e 10 (“A oficina foi:”) apresentaram uma baixa correlação item-total, além disso, o item 2 apresenta um pequeno acréscimo no alfa de Cronbach caso seja excluído, desse modo, os resultados indicam que esses itens precisam ser revisados.

QA3: Quantos fatores subjacentes influenciam as respostas dos itens do instrumento de medição do modelo dTECT?

De modo a determinar quantos fatores de qualidade estão subjacentes ao conjunto de itens do instrumento de medição do modelo dTECT, foi realizada uma análise fatorial seguindo Brown (2006). Para analisar se os itens do instrumento de medição do modelo dTECT podem ser submetidos a uma análise fatorial, utilizou-se o índice de Kaiser-Meyer-Olkin (KMO). Como resultado, obteve-se um índice KMO de 0,827, indicando que a análise fatorial é apropriada nesse caso. Aplicando a análise fatorial, o número de fatores retidos na análise é decidido com base no critério de Kaiser-Guttman, que determina que o número de fatores é igual ao número de fatores com autovalor maior que 1 (Carmines & Zeller, 1979). Desse modo, justifica-se a retenção de três fatores que estão influenciando as respostas das avaliações das UIs. Em relação ao modelo dTECT, isso significa que as respostas do instrumento de medição estão representando três fatores subjacentes, indicando uma decomposição igual à definição original do modelo.

Uma vez identificado o número de fatores subjacentes, outra questão é determinar quais itens contribuem para qual fator. A fim de identificar as cargas fatoriais dos itens, o método de rotação

Promax com Normalização de Kaiser é usado, por ser o método de rotação mais amplamente aceito e utilizado (Carmines & Zeller, 1979). A Tabela 4 mostra as cargas fatoriais dos itens associado com os três fatores retidos. A maior carga fatorial de cada item, indicando a qual fator o item está mais relacionado, é marcada em negrito.

Fator de Qualidade	N.º item	Descrição	Fatores		
			1	2	3
Qualidade da UI	1	A oficina foi:	,146	,763	-,018
	2	O tempo das aulas passou:	-,096	,619	,043
	3	A oficina foi:	,110	,800	-,078
Experiência de Computação	4	Vou mostrar meu programa de computador para outras pessoas:	,591	,101	,008
	5	Quero aprender mais sobre como fazer programas de computador:	,571	,217	,035
	6	Fazer um programa de computador é:	,510	,400	,055
	7	Eu gosto de fazer programas de computador:	,683	,114	-,053
	8	A computação é útil no dia a dia:	,783	-,213	-,090
	9	Você já pensou em trabalhar com computação?	,401	,130	,165
Percepção da Aprendizagem	10	A oficina foi:	-,425	,239	,823
	11	Eu consigo fazer programas de computador:	,415	-,177	,546
	12	Consigo explicar para um amigo/amiga como fazer um programa de computador:	,432	-,139	,410
	13	Fazer um programa de computador é:	,230	-,102	,720

Tabela 4 – Cargas fatoriais

Fonte: autores

Analisando as cargas fatoriais dos itens, pode-se observar que o primeiro fator (fator 1) é representado por um conjunto de sete itens (4, 5, 6, 7, 8, 9 e 12). Assim, esse fator está diretamente relacionado ao

fator de qualidade sobre a experiência de computação proporcionada pela unidade instrucional (Tabela 4). Isso porque, com exceção do item 12, todos os itens correspondem ao referido fator de qualidade no modelo dTECT. Embora o item 12 apresente a maior carga fatorial no fator 1, ele também apresenta uma carga fatorial semelhante (,410) no fator 3, portanto, mostrando que esse item contribui para os dois fatores de qualidade (experiência de computação e percepção da aprendizagem). Em relação ao fator 2, é considerado um conjunto de três itens (1, 2 e 3). Esse resultado parece sugerir que esses itens estão relacionados ao fator sobre a qualidade da unidade instrucional do modelo dTECT. Isso porque, de fato, esses referidos itens correspondem a um mesmo fator de qualidade (qualidade da UI) na definição original do modelo dTECT (Tabela 1). Analisando os resultados do fator 3, ele inclui um conjunto de três itens (10, 11 e 13), indicando que esses itens estão relacionados a um único fator de qualidade.

DISCUSSÃO

Os resultados obtidos apresentam evidências suficientes para considerar o modelo dTECT confiável e válido para a avaliação de unidades instrucionais para o ensino de computação na educação básica.

Em termos de confiabilidade, os resultados da análise indicam um coeficiente alfa de Cronbach aceitável (Cronbach's alpha $\alpha = .787$), indicando a consistência interna do conjunto de itens do instrumento de medição do modelo dTECT. Assim, indica que os itens do instrumento de medição do modelo dTECT são consistentes e precisos para avaliar a qualidade de unidades instrucionais.

Em termos de validade de constructo, em relação à validade convergente, identificou-se que os itens tendem a ter uma média e alta correlação entre si, principalmente entre os itens de um mesmo fator de qualidade (ex., qualidade da UI, experiência de computação e percepção da aprendizagem). Desse modo, pode-se concluir que existe evidência de validade convergente considerando os fatores

de qualidade. Isso indica que os itens do instrumento de medição do modelo dTECT parecem estar medindo realmente o que se propõem a medir (ex., qualidade da UI, experiência de computação e percepção da aprendizagem). No entanto alguns itens apresentam baixa correlação, tanto em um único fator de qualidade quanto em relação aos outros fatores, isso pode indicar que alguns itens precisam ser revistos, principalmente quanto à sua descrição. Com relação à validade discriminante, em geral, a maioria dos itens apresentam uma correlação baixa com itens de outro fator de qualidade. Porém alguns pares de itens (ex., 1-6, 5-11) apresentaram uma média ou alta correlação com itens de outro fator de qualidade. Desse modo, os resultados não indicam evidências de validade discriminante. No entanto, nesse caso, a falta de existência de validade discriminante é aceitável, pois, embora o modelo seja dividido em três fatores de qualidade, todos os fatores também estão relacionados a um único fator, que é a percepção da qualidade da unidade instrucional, como proposto na composição original do modelo dTECT (Figura 2).

Analisando a correlação item-total, novamente, a grande maioria dos itens apresenta valores de correlação satisfatórios com os outros itens do instrumento de medição, indicando que o conjunto de itens do instrumento de medição do modelo dTECT está relacionado para medir o que se propõem a medir (percepção da qualidade de UI).

Com base nos resultados da análise fatorial, identificou-se que os dados coletados nos estudos de caso são explicados por três fatores. Confirma-se, assim, a estrutura inicial definida para o modelo dTECT, claramente agrupando os itens conforme seu fator de qualidade definido (qualidade da UI, experiência de computação e percepção da aprendizagem).

Ameaças à validade

Devido às características deste tipo de pesquisa, este estudo está sujeito a ameaças à validade. Assim, foram identificadas possíveis ameaças e aplicadas estratégias de mitigação de modo a minimizar o

seu impacto nesta pesquisa. Algumas ameaças estão relacionadas ao tipo de estudo. Desse modo, para minimizar essa ameaça, foi definida e documentada uma metodologia sistemática para este estudo. O modelo dTECT foi sistematicamente definido com base na abordagem GQM, decompondo sistematicamente o objetivo de avaliação em questões de análise e medidas. O instrumento de medição foi desenvolvido seguindo um método definido na literatura e envolvendo uma equipe multidisciplinar de pesquisadores. Além disso, para a avaliação do instrumento de medição do modelo dTECT, um estudo de caso também foi sistematicamente definido e documentado. Outro risco refere-se à qualidade dos dados agrupados em uma amostra, em termos de padronização dos dados e adequação ao modelo. Em relação a isso, como este estudo é exclusivo para avaliar UIs que utilizam o modelo dTECT, o risco é minimizado, pois todos os estudos de caso utilizam o mesmo instrumento de medição (com mesmo padrão de resposta) para a coleta de dados. Outra ameaça refere-se ao agrupamento de dados de diferentes contextos. Para minimizar esse risco, foram considerados estudos de caso somente de UIs para o ensino de computação exclusivamente em nível de educação básica.

Em termos de validade externa, uma ameaça para generalizar os resultados está relacionada ao tamanho da amostra e diversidade dos dados utilizados na avaliação. Em relação a isso, a avaliação do modelo dTECT usou dados coletados em 16 estudos de caso avaliando três diferentes unidades instrucionais em 13 instituições de educação básica, totalizando uma amostra de 477 estudantes. Em termos de significância estatística, o tamanho da amostra é considerado satisfatório, permitindo a geração de resultados significantes.

Em relação à confiabilidade do estudo, uma ameaça refere-se a quanto os dados e a análise são dependentes dos pesquisadores específicos. Para mitigar essa ameaça, foi documentada uma metodologia sistemática, tanto para o desenvolvimento do modelo dTECT quanto para o estudo de caso de sua avaliação, definindo claramente o objetivo do estudo, o processo de coleta de dados e os métodos estatísticos utilizados para a análise dos dados. Além disso, outra ameaça refere-se à correta escolha dos métodos estatísticos para a análise dos dados. Para minimizar essa ameaça, a análise estatística baseou-se no guia de desenvolvimento de escalas

proposto por Devellis (2016), que está alinhado aos procedimentos para a avaliação de consistência interna e validade de constructo de um instrumento de medição (Trochim & Donnelly, 2008).

CONCLUSÃO

Embora a avaliação de unidades instrucionais para o ensino de computação seja essencial para sua melhoria contínua, efetiva e eficiente aplicação, poucos esforços têm sido feitos para o desenvolvimento de modelos sistemáticos de avaliação. Nesse contexto, este estudo apresenta uma primeira proposta na direção de levar em consideração restrições práticas ao executar essas avaliações em programas de extensão mais informais. Com base na literatura e em experiências práticas, o modelo de avaliação dTECT, incluindo um instrumento de medição com 13 itens, foi desenvolvido sistematicamente e aplicado (pós-unidade) em 16 oficinas/cursos na educação básica no Brasil. Os resultados da análise das respostas de 477 alunos indicam que o instrumento de medição do modelo dTECT é aceitável em termos de confiabilidade e validade de constructo. Em termos de confiabilidade, o Alfa de Cronbach é de $\alpha=0,787$, indicando uma aceitável consistência interna, o que significa que as respostas entre os itens são consistentes e precisas. Os resultados da análise também indicam um aceitável grau de correlação entre os itens em relação aos fatores de qualidade. Assim, sugere-se que o instrumento de medição do modelo dTECT pode ser um instrumento confiável e válido para medir a percepção dos estudantes em relação à qualidade de unidades instrucionais para o ensino de computação. Os resultados da análise fatorial indicam que três fatores subjacentes influenciam as respostas dos itens do instrumento de medição do modelo dTECT. Esse resultado confirma a estrutura original do modelo dTECT (Figura 2), que define três fatores de qualidade (qualidade da UI, experiência de computação e percepção da qualidade) para a avaliação de unidades instrucionais. E, embora o foco inicial do modelo dTECT seja a Educação Básica, para a qual demonstra ser válido e confiável, o modelo

pode também ser aplicado para avaliar o ensino de computação em ambientes de Educação Integral.

AGRADECIMENTOS

Este estudo foi apoiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e o *Google Rise Award*.

REFERÊNCIAS

- Basili, V. R., Caldiera, G., Rombach, H. D. (1994). Goal, Question Metric Paradigm. In Marciniak, J. J. (Ed.). *Encyclopedia of Software Engineering*. John Wiley & Sons, p. 528–532.
- Brown, T. A. (2006). *Confirmatory factor analysis for applied research*. 1. ed. New York: The Guilford Press, 475 p.
- Carmines, E. G., & Zeller, R. A. (1979). *Reliability and validity assessment*. 1. ed. Beverly Hills: Sage Publications Inc., 75 p.
- Cohen, J. (1988). *Statistical Power Analysis for the Behavioral Sciences*. Routledge Academic.
- Craig, M., & Horton, D. (2009). Gr8 designs for Gr8 girls: a middle-school program and its evaluation. *40th ACM Technical Symposium on Computer Science Education*. Chattanooga, TN, USA, p. 221–225.
- Cronbach, L. J. (1951). Coefficient alpha and the internal structure of tests. *Psychometrika*, 16(3), 297–334.
- Decker, A., McGill, M. M., & Settle, A. (2016). Towards a Common Framework for Evaluating Computing Outreach Activities. *ACM Technical Symposium on Computer Science Education*. Memphis, TN, USA, p. 627–632.

Devellis, R. F. (2016). *Scale development: theory and applications*. 4. ed. SAGE Publications, 280 p.

Garland, K. J., & Noyes, J. M. (2008). Computer attitude scales: How relevant today? *Computers in Human Behavior*, 24(2), 563-575.

Garneli, V., Giannakos, M. N., & Chorianopoulos, K. (2015). Computing Education in K-12 Schools: A Review of the Literature. *IEEE Global Engineering Education Conference*. Tallinn, Estonia, p. 543-551.

Giannakos, M. N., Hubwieser, P., & Chrisochoides, N. (2013). How Students Estimate the Effects of ICT and Programming Courses. *ACM Technical Symposium on Computer Science Education*. Denver, CO, USA, p. 717-722.

Giannakos, M. N., Jaccheri, L., & Leftheriotis, I. (2014). Happy Girls Engaging with Technology: Assessing Emotions and Engagement Related to Programming Activities. *International Conference on Learning and Collaboration Technologies*. Heraklion, Greece, p. 398-409.

Gresse von Wangenheim, C. & von Wangenheim, A. (2014). Teaching Game Programming in Family Workshops. *IEEE Computer Magazine*, v. 47, n. 8, p. 84-87.

Gross, P., & Powers, K. (2005). Evaluating Assessments of Novice Programming Environments. *International Workshop on Computing Education Research*. Seattle, WA, USA, p. 99-110.

Guzdial, M. (2004). Programming environments for novices. In Fincher, S., & Petre, M. (Eds.). *Computer Science Education Research*. Swets and Zeitlinger. Chapter 3.

Guzdial, M., Ericson, B., Mcklin, T., & Engelman, S. (2014). Georgia Computes! An intervention in a US state, with formal and informal education in a policy context. *ACM Transactions on Computing Education*, 14(2), article 13.

Kalelioğlu, F., & Gülbahar, Y. (2014). The Effects of Teaching Programming via Scratch on Problem Solving Skills: A Discussion from Learners' Perspective. *Informatics in Education*, 13(1), 33-50.

Kasunic, M. (2005). *Designing an effective survey*. Carnegie-Mellon University Pittsburgh Pa Software Engineering Inst.

Krosnick, J. A., & Presser, S. (2010). Questionnaire design. In Wright, J. D., & Marsden, P. V. (Eds.). *Handbook of Survey Research*. 2. ed. West Yorkshire, England: Emerald Group.

Malhotra, N. K., Birks, D. F. (2008). *Marketing Research: An Applied Approach*. 3. ed. Trans-Atlantic Publications, 816 p.

Miller, D. C., Salkind, N. J. (2002). *Handbook of Research Design and Social Measurement*. 6. ed. SAGE Publications, 808 p.

Mühling, A., Ruf, A., & Hubwieser, P. Design and First Results of a Psychometric Test for Measuring Basic Programming Abilities. *Workshop in Primary and Secondary Computing Education*, London, United Kingdom, 2015.

Perdikuri, K. (2014). Students' Experiences from the use of MIT App Inventor in classroom. *Panhellenic Conference on Informatics*. Athens, Greece, p.1-6.

Poels, K., Kort, Y. D., & Ijsselstein, W. (2007). It is always a lot of fun!: exploring imensions of digital game experience using focus group methodology. *7th Conference on Future Play*. Toronto, p. 83-89.

Sweetser, P., & Wyeth, P. (2005). GameFlow: a model for evaluating player enjoyment in games. *Computers in Entertainment*, 3(3), 1-24.

Takatalo, J., Häkkinen, J., Kaistinen, J., & Nyman, G. (2010). Presence, Involvement, and Flow in Digital Games. In Bernhaupt, R. (Ed.). *Evaluating User Experience in Games: Concepts and Methods*, Springer, p. 23-46.

Trochim, W. M., & Donnelly, J. P. (2008). *Research methods knowledge base*. 3. ed. Mason, OH: Atomic Dog Publishing, 361 p.

Weintrop, D., & Wilensky, U. (2015). To Block or Not to Block, That is the Question: Students' Perceptions of Blocks-based Programming. *Conference on Interaction Design and Children*. New York, NY, USA, p. 199-208.

Wilson, A., & Moffat, D. C. (2010). Evaluating Scratch to introduce younger schoolchildren to Programming. *Annual Workshop of the Psychology of Programming Interest Group*. Leganés, Spain

Wohlin, C., Runeson, P., Host, M., Ohlsson, M. C., Regnell, B., & Wesslen, A. (2012). *Experimentation in Software Engineering*. Springer Verlag.

Yin, R. K. (2009). *Case study research: design and methods*. 4. ed. Beverly Hills: Sage Publications, 312p.

CAPÍTULO 5

A MODELAGEM MATEMÁTICA COMO METODOLOGIA PARA O DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL POR MEIO DA CRIAÇÃO DE OBJETOS DE APRENDIZAGEM

*Márcia Regina Kaminski
Clodis Boscaroli*

INTRODUÇÃO

A formação da cidadania crítica é um dos principais aspectos que permeiam as discussões sobre o papel da escola. A maioria dos Projetos Políticos Pedagógicos das instituições de ensino destaca a formação cidadã crítica como principal objetivo do ensino escolar, de modo que as escolas têm, há algum tempo, de acordo com Santos (2006), reconhecido os conteúdos escolares como necessários para que os estudantes compreendam sua realidade e sejam capazes de discutir, debater e opinar de forma a intervir sobre ela para modificá-la. Há, também, a preocupação em estruturar os Currículos de forma a propiciar aos estudantes essa formação cidadã crítica.

O acesso ao conhecimento científico e técnico construído historicamente é fundamental, conforme apontado por Gouvêa (2015), para que os indivíduos tenham condições e se tornem capazes de discutir e opinar criticamente como cidadãos atuantes e participativos, e a escola tem papel fundamental em ajudar os estudantes a obter e sistematizar esse conhecimento. Porém atingir o objetivo de propiciar essa formação envolve aspectos que ultrapassam a mera transmissão de informações e abrangem a formação integral em aspectos científicos, cognitivos, sociais e emocionais, sendo necessário considerar os interesses dos alunos, contextualizar os conteúdos com a realidade, estimular a iniciativa e valorizar as suas atividades, em consonância com a sistematização dos conhecimentos, conforme apontado por Saviani (1999).

De acordo com a Base Nacional Comum Curricular (Brasil, 2018), é importante estruturar o ensino de modo a propiciar a formação integral que, muito além da transmissão de informações, engloba o desenvolvimento da capacidade de comunicação, da criatividade, da análise crítica, da resiliência, da colaboração, da responsabili-

dade, da cooperação, da tomada de decisões, da proatividade, da autonomia, entre outros aspectos.

No novo cenário mundial, reconhecer-se em seu contexto histórico e cultural, comunicar-se, ser criativo, analítico-crítico, participativo, aberto ao novo, colaborativo, resiliente, produtivo e responsável requer muito mais do que o acúmulo de informações. Requer o desenvolvimento de competências para aprender a aprender, saber lidar com a informação cada vez mais disponível, atuar com discernimento e responsabilidade nos contextos das culturas digitais, aplicar conhecimentos para resolver problemas, ter autonomia para tomar decisões, ser proativo para identificar os dados de uma situação e buscar soluções, conviver e aprender com as diferenças e as diversidades (Brasil, 2018, p. 14).

Em harmonia com o documento supracitado, é necessário pensar o desenvolvimento dos sujeitos em todas as dimensões, assumindo uma visão plural, que inclui aspectos cognitivos e afetivos e que consideram a singularidade e a diversidade dos sujeitos, respeitando as diferenças e comprometendo-se com um processo educativo que intencionalmente contribua para o desenvolvimento dos estudantes em todos esses aspectos, junto com o conhecimento científico, considerando também as possibilidades, as necessidades e os desafios da sociedade contemporânea.

Nota-se no contexto escolar a necessidade de um trabalho voltado para esses objetivos, visto que não raro os conteúdos são abordados em sala de aula como se não estivessem conectados à realidade e como se pertencessem a cada uma das disciplinas do currículo separadamente, sem haver relação entre as diversas áreas do conhecimento e por meio de abordagens metodológicas centradas na transmissão de informações pelo professor, que não propiciam a participação ativa dos estudantes, dificultando o desenvolvimento da criatividade, da colaboração, da cooperação, da autonomia, do

comprometimento, da capacidade de diálogo e argumentação, do respeito às diferenças, entre outros aspectos relevantes.

Nesse sentido, as práticas pedagógicas devem levar em consideração o contexto da realidade *cibercultural*, considerando as Tecnologias Digitais da Informação e Comunicação (TDIC), como parte do conhecimento ao qual os estudantes devem ter domínio, e adotando abordagens e estratégias metodológicas que possibilitem o ensino dos conteúdos científicos de forma articulada com a realidade e que coloquem os alunos em atividades investigativas, de análise, de avaliação, de levantamento e verificação de hipóteses para que possam desenvolver aspectos importantes à formação cidadã.

O Pensamento Computacional (PC) é um processo de resolução de problemas, das diferentes áreas do conhecimento, baseado nos fundamentos da Ciência da Computação, que segue um conjunto de passos estruturados, a fim de encontrar a solução de forma mais simples possível, e que favorece a contextualização do ensino e o desenvolvimento integral e crítico dos estudantes, podendo ser utilizado como elo entre diferentes conteúdos, e como forma de dinamizar o processo de ensino, inclusive como prática de formação cidadã.

A Modelagem Matemática (MM) é uma tendência da Educação Matemática, que tem por objetivo a contextualização dos conteúdos por meio da resolução de problemas, baseados em situações da realidade, seguindo um conjunto de etapas sistemáticas das quais os estudantes são participantes ativos na busca da solução, e visa à interdisciplinaridade. Ambos, PC e MM, têm recebido destaque na educação em função das contribuições que oferecem para o desenvolvimento da capacidade crítica, analítica, reflexiva, sintética, da tomada de decisões, da autonomia.

Considerando a relevância dos aspectos supracitados e as contribuições do PC e da MM para o ensino, o presente estudo tem por finalidade apresentar, de forma a permitir a reprodutibilidade em outros contextos educacionais, o que vem sendo desenvolvido com estudantes do 5º ano do ensino fundamental I, em uma escola pública do município de Cascavel/PR, onde são explorados os conceitos do PC por meio da metodologia da MM, contribuindo não apenas para a apropriação do conhecimento científico, mas para a formação da cidadania crítica dos estudantes.

O documento segue assim organizado. Na Seção 2, discute-se sobre a relevância do PC e possibilidades de abordagem sob a perspectiva da MM. A Seção 3 apresenta a forma como a escola tem desenvolvido esse trabalho juntamente com alguns dos resultados obtidos, e na Seção 4 estão as conclusões e as perspectivas da pesquisa.

PENSAMENTO COMPUTACIONAL E MODELAGEM MATEMÁTICA

Atingir o objetivo de propiciar a formação da cidadania crítica exige que sejam adotadas concepções teóricas e estratégias metodológicas que possibilitem que os estudantes exercitem, durante os processos de ensino e aprendizagem, aspectos cognitivos (resolução de problemas, memória, raciocínio, atenção, elaboração de estratégias, análise, reformulação de ideias, reflexão, observação); sociais (colaboração, cooperação, relacionamento, comunicação, argumentação); emocionais (autonomia, responsabilidade, protagonismo, autoconfiança, criatividade, motivação, curiosidade, iniciativa), além da abordagem do conhecimento científico não fragmentado, mas de forma interdisciplinar e transdisciplinar contextualizada com a realidade em todas as disciplinas e em todos os níveis de ensino.

O PC é um método de resolução de problemas definido por Wing (2006) como um processo no qual se aplicam os fundamentos da Ciência da Computação para solucionar problemas de diversas áreas, que inclui ordenar e analisar os dados logicamente, por meio de diversas etapas que envolvem abstração, elaboração de modelos, algoritmos e simulações. Para essa autora, os humanos podem utilizar a lógica presente nos sistemas computacionais para resolver problemas e é esse tipo de pensamento e lógica que deve ser ensinado a todas as pessoas, e não apenas aos estudantes de Computação, isso porque, conforme ela destaca, esse tipo de pensamento estimula a capacidade crítica e analítica tanto da escrita quanto da linguagem matemática, desenvolvendo a capacidade de planejamento, elaboração de estratégias, sequência e síntese.

Blikstein (2008) já salientava que saber pensar computacionalmente aumenta o poder cognitivo e é essencial à formação do cidadão crítico do século XXI, para o qual não basta saber navegar na *internet* e utilizar os recursos básicos que as tecnologias podem oferecer. Para o autor, o cidadão precisa deixar de ser apenas consumidor passivo da tecnologia, mas utilizá-la para produzir conhecimento, pois somente assim, embora seja cada vez mais dependente da máquina para a realização de muitas tarefas e demandas, o ser humano pode se tornar cada vez mais produtivo e criativo na medida em que pode concentrar seu potencial em desenvolver o que a máquina não pode fazer.

O cidadão contemporâneo precisa dominar a tecnologia no sentido de valer-se dela para a realização de tarefas que sejam mecânicas, rotineiras, que exijam velocidade, para passar a se preocupar com a produção de conhecimento que a máquina não é capaz de produzir, de modo que possa utilizar o conhecimento sistematizado para atuar de forma crítica transformando a realidade. É necessário, portanto, que a escola propicie e tenha o compromisso com o desenvolvimento da capacidade de “compreender e interpretar o mundo (natural, social e tecnológico), mas também de transformá-lo com base nos aportes teóricos e processuais das ciências” (Brasil, 2018, p. 319).

Para isso, torna-se necessário compreender os mecanismos envolvidos nos recursos tecnológicos, o que inclui pensar computacionalmente. França *et al.* (2014) consideram que o PC na Educação Básica desenvolve a capacidade de resolução de problemas em todas as áreas do conhecimento, contribuindo de forma relevante à formação crítica.

Segundo Brackmann (2017), o PC é constituído de quatro pilares: decomposição (capacidade de decompor um problema em partes menores mais fáceis de ser resolvidas); reconhecimento de padrões (capacidade de reconhecer padrões similares em problemas); abstração (envolve a filtragem e classificação de dados), e algoritmos (elaboração de um plano, estratégia composta de um conjunto de instruções para solucionar o problema). Como benefícios do desenvolvimento do PC, o autor indica que pode auxiliar muito na aprendizagem, especialmente no desenvolvimento do raciocínio lógico matemático, além da preparação para o trabalho, compreensão

do mundo, transversalidade, alfabetização digital, produtividade, inclusão, trabalho em equipe e superação das dificuldades pessoais.

Com base nessas concepções, o PC vem conquistando espaço nos ambientes escolares, podendo ser trabalhado de diversas formas, seja com atividades de computação desplugada, seja incluindo a utilização de tecnologias como jogos para desenvolvimento do raciocínio lógico ou ambientes para programação visual.

Brackmann (2017) aponta que a programação em *Scratch* (MIT, 2007) é uma das formas de trabalhar o desenvolvimento do PC, e para Ferri e Rosa (2016), *Scratch* é uma das ferramentas mais utilizadas para o trabalho com programação visual, devido ao encaixe sequencial de blocos de comando já prontos¹⁰. Desenvolvido para ensinar conceitos básicos de programação para crianças, é intuitivo e simples de ser utilizado, não exigindo conhecimento profundo de Linguagem de Programação dos seus usuários. As cores e formas dos blocos facilitam a criação de animações, histórias ou jogos, orientados pelos objetivos e criatividade do usuário. Por ser disponibilizado gratuitamente, tanto de forma *online* como *offline*, pode ser explorado mesmo em realidades nas quais o acesso à *internet* é inexistente.

O trabalho com o desenvolvimento do PC por meio do *Scratch* oferece possibilidades de abordagem dos conteúdos curriculares que propiciam aos estudantes compreender a relevância do conhecimento científico no desenvolvimento da humanidade e para o seu próprio desenvolvimento como cidadão crítico, participativo e atuante na sociedade. Porém, assim como ocorre com qualquer atividade de ensino, exige objetivos bem definidos, acompanhados de mudanças metodológicas compatíveis com esses objetivos para que as contribuições para os processos de ensino e aprendizagem possam ser relevantes e que as atividades não sejam apenas um momento de ludicidade ou exploração livre.

Embora a criação de animações ou jogos de forma livre no *Scratch* contribua para o desenvolvimento da criatividade e exploração do *software*, é importante, especialmente em contextos escolares, não limitar as atividades apenas nesses aspectos, considerando que o

¹⁰ Outras plataformas podem ser exploradas para o trabalho com Programação Visual, como CODE.org, Programaê, *Lightbot*, sendo que o principal diferencial entre elas e o *Scratch* é a possibilidade de criação por parte do usuário que o *Scratch* oferece.

trabalho realizado na escola requer uma intencionalidade que conduza à formação da cidadania crítica. Portanto, é importante pensar em metodologias que favoreçam a integração entre a abordagem dos conteúdos curriculares e o desenvolvimento do PC, além de aspectos sociais e emocionais. Nesse sentido, a MM pode ser uma opção metodológica. Burak (1992, p. 62) traz a seguinte definição para MM:

A Modelagem Matemática constitui-se em um conjunto de procedimentos cujo objetivo é construir um paralelo para tentar explicar, matematicamente, os fenômenos presentes no cotidiano do ser humano, ajudando-o a fazer predições e a tomar decisões.

Na Perspectiva da Educação Matemática, cujo enfoque principal é o ensino e a aprendizagem da Matemática, a MM pode ser definida, de acordo com Borssoi e Almeida (2004), como uma forma de tratar de uma situação-problema da realidade, por meio da Matemática, com a participação ativa dos estudantes durante todo o processo (desde a escolha do problema até a identificação da solução, a elaboração e verificação das próprias hipóteses), seguindo um conjunto determinado de procedimentos, abordando os conteúdos de maneira interdisciplinar e contextualizada.

Para Almeida e Silva (2010), essa abordagem contextualizada dos conteúdos por meio da MM propicia o desenvolvimento de habilidades importantes à formação da cidadania crítica, como coleta de dados, formulação, verificação e validação de hipóteses, obtenção e interpretação da resposta ao problema proposto, tornando o ensino mais contextualizado com a realidade e significativo.

Percebem-se, assim, similaridades importantes entre atividades com PC e MM e que ambos podem oferecer contribuições significativas à formação da cidadania crítica, de modo que é possível pensar em uma abordagem do PC por meio da MM, haja vista que, segundo Almeida e Silva (2010), os problemas escolhidos para o trabalho com MM não precisam ser necessariamente problemas Matemáticos, mas podem surgir de qualquer contexto da realidade. Nesse sentido,

Vecchia (2012) apregoa que a Ciência como base do desenvolvimento tecnológico permite compreender o universo tecnológico como uma das dimensões da realidade, expandindo as possibilidades de trabalho com MM, a partir do reconhecimento de que os problemas tratados com essa abordagem podem partir dos contextos cibernéticos, que, por sua vez, também fazem parte da realidade e do cotidiano.

Assim, considerando os avanços tecnológicos e suas consequentes transformações nos diversos âmbitos da sociedade, como econômico, político, social, que por consequência afetam e modificam o cotidiano das pessoas em muitos aspectos relacionados às suas necessidades, como saúde, trabalho, e, de forma inevitável, também trazem impactos à educação, para a forma como os estudantes aprendem e encaram o ensino escolar, as TDIC podem ser encaradas como uma das dimensões da realidade nas quais os problemas da MM podem se estruturar.

Reconhecendo as TDIC como parte desse saber científico e técnico construído historicamente, compreende-se que fazem parte do arsenal de conhecimentos aos quais os estudantes devem ter acesso, sendo necessário que conheçam não apenas os seus princípios básicos, mas que compreendam os conceitos científicos envolvidos nos recursos tecnológicos por eles utilizados: “[...] é preciso garantir não apenas o domínio técnico-operativo dessas tecnologias, mas a compreensão dos princípios científicos e dos processos que as tornaram possíveis” (Saviani, 2010, p. 32).

Dentre as possibilidades de abordagem do PC por meio da MM, Vecchia e Maltempi (2014) apontam que a linguagem de programação visual do *Scratch* permite discutir, analisar, simular, avaliar, interpretar resultados da mesma maneira como ocorre com um Modelo Matemático, embora se utilize de uma linguagem diferenciada que envolva a linguagem dos símbolos matemáticos junto com a linguagem falada.

A Escola Municipal Aloys João Mann, situada na cidade de Cascavel/PR, tem procurado estratégias para o desenvolvimento do PC alinhado aos conteúdos curriculares, visando à formação da cidadania crítica dos estudantes desde 2013. Desde o início do trabalho, vários encaminhamentos foram adotados para essa finalidade, passando da exploração livre do *Scratch* à reprodução de animações ou jogos disponíveis como exemplos no próprio site do *Scratch* e desta à pro-

dução de Objetos de Aprendizagem (OA) pelos alunos para os alunos, até uma abordagem que utiliza a MM como metodologia, sendo que essa última opção tem se mostrado adequada e que passaremos a relatar na próxima seção, que descreve brevemente a trajetória do trabalho desenvolvido com o PC, destacando a forma como hoje é realizada a sua abordagem por meio da MM.

DESENVOLVENDO O PC NA PERSPECTIVA DA MM

A proposta de abordagem do PC utilizando a MM como estratégia metodológica é baseada na experiência da escola Aloys João Mann, de Cascavel/PR, onde, desde 2013, há a preocupação de desenvolver práticas pedagógicas com PC com os estudantes, durante as aulas de Informática Educacional que fazem parte das atividades semanais dos alunos, desde a educação infantil até o 5º ano. As aulas têm atualmente a duração de 40 minutos e são ministradas por uma instrutora de Informática.

O trabalho com PC é desenvolvido utilizando estratégias diferenciadas para cada turma, de acordo com a faixa etária dos estudantes, e variam entre atividades com computação desplugada e programação visual em diferentes ambientes. Com os estudantes de 5º ano, o foco do trabalho está na programação visual com o *Scratch*. Visando à articulação entre o desenvolvimento do PC, os conteúdos curriculares e os aspectos sociais e emocionais, a proposta de trabalho para o 5º ano, desde 2014, é a produção de Objetos de Aprendizagem (OA) pelos estudantes para uso por estudantes de anos anteriores, conforme descrito em Kaminski e Boscarioli (2018a).

Neste estudo, compreendemos como OA conteúdos digitais desenvolvidos para fins educacionais, conforme definido por Filatro (2008). Segundo Silva et al. (2014), os OA são importantes recursos para os processos de ensino e aprendizagem, uma vez que alinham conteúdos pedagógicos com a ludicidade, tornando as atividades mais

atrativas. Dessa forma, se utilizar um OA em um contexto de ensino pode ser algo atrativo para os estudantes, produzir seu próprio OA sabendo que será utilizado por outros estudantes torna-se ainda mais desafiador e motivador para os alunos, que conseguem atribuir um objetivo claro à sua produção. Além disso, a produção de um OA vincula diretamente os conteúdos curriculares ao desenvolvimento do PC, de modo que é possível desenvolver os dois aspectos simultaneamente.

Considerando esses fatos, a escola tem utilizado a produção de OA pelos estudantes no *Scratch* como uma das estratégias para o desenvolvimento do PC. Porém, buscando melhor embasamento teórico e metodológico para essas atividades, começou-se a experimentar a criação desses objetos de aprendizagem seguindo os passos da MM, conforme já relatado em Kaminski e Boscaroli (2018b; 2018c). Com base nessas experiências, propõe-se aqui o trabalho com PC utilizando a MM como estratégia metodológica, sob uma perspectiva de formação integral, onde são desenvolvidos aspectos cognitivos, científicos, sociais e emocionais. Segundo Burak e Klüber (2016), o trabalho com MM é caracterizado por etapas metodológicas que podem ser ou não seguidas linearmente. Na sequência, são descritas na Figura 1 cada uma dessas etapas e como estas são desenvolvidas durante o trabalho com PC na produção de OA com o *Scratch* pelos alunos.

Etapa	Desenvolvimento
Escolha do tema: Fase em que os problemas a serem estudados são propostos, discutidos e definidos. Partem de uma situação da realidade, preferencialmente escolhida pelos estudantes, podendo ser definidos ou sugeridos pelo professor em casos nos quais os alunos são iniciantes nas atividades com MM, como é o caso dos anos iniciais (Borssoi & Almeida, 2004).	(1h/aula): Considerando ser a primeira experiência com atividades de MM dos estudantes que têm em média 10 anos de idade, o tema foi proposto pela Instrutora de Informática, que, a exemplo de Vecchia (2012), partiu de contextos digitais, sendo definida a criação de jogos, em especial destinados à educação (OA), como tema de estudo. Para introduzir a atividade, conduzir a discussão em torno do tema analisando aspectos como: o que são jogos? Que tipos de jogos existem? São apenas digitais? Quais as características de um jogo? Que elementos fazem parte de um jogo?

Pesquisa Exploratória: Etapa em que os estudantes estudam e analisam o tema levantado e os aspectos nele envolvidos, buscando compreendê-lo.	Fase 1(1h/aula): Discutir o que são OA e suas características(ter um conteúdo pedagógico, dar um <i>feedback</i> ao jogador, forma de interação com o jogador, entre outros elementos). Apresentar alguns OA aos alunos e discutir essas características por meio de questionamentos como: sobre o que é esse jogo? Qual o conteúdo envolvido nele? Como acontece a interação entre o jogador e o objeto (por mouse, teclado)? Como sabemos se acertamos ou erramos? Há pontuação? Há algum som, personagem, cenário ou outros elementos?
	Fase 2(1h/aula): Apresentar o <i>Scratch</i> aos estudantes explicando seus objetivos. Criar uma animação simples para aguçar a curiosidade e explicar o uso dos principais comandos. Propor um tempo de exploração livre dos recursos principais.
	Fase 3 (1h/aula): Exploração dos principais recursos: por meio dos projetos compartilhados do <i>site do Scratch</i>, apresentar aos alunos algumas possibilidades de jogos como fonte de ideias. Propor que os alunos escolham um projeto e analisem o código avaliando qual a função de cada um dos blocos utilizados.
	Fase 4 (1h/aula): Criação de um jogo ou animação livre. A fim de reduzir a ansiedade para utilizar os diversos recursos, propor que os alunos criem um jogo ou animação livre. No decorrer do trabalho, esclarecer dúvidas. Durante as mediações pedagógicas da aula, auxiliar a partir de questionamentos os estudantes a avaliarem e analisarem se o código criado por eles cumpriu o objetivo, se houve problemas, quais foram, por que ocorreram e como podem ser solucionados.
	Fase 5(1h/aula): Propor que os alunos elaborem, em um editor de texto, um roteiro simples do OA que desejam criar. Devem ser definidos a faixa etária de alunos que utilizarão o OA, o conteúdo que será abordado, a sua natureza (<i>quiz</i>, labirinto, aventura ou outro – ideias podem ser pesquisadas, assim como o código fonte básico que será necessário para criação do OA escolhido). Iniciar a pesquisa do conteúdo definindo enredo do OA, questões, respostas corretas, alternativas erradas, personagens, seleção de imagens e outros recursos necessários.

Levantamento dos problemas: Etapa na qual, a partir dos dados obtidos na pesquisa exploratória, os problemas são elaborados e analisados.	Fase 6 (4h/aula, em média): Estas duas etapas ocorrem paralelamente, durante o processo de criação do OA no <i>Scratch</i>, com base no roteiro previamente elaborado na aula anterior. Durante a criação, surgem os problemas que consistem basicamente na elaboração do código adequado aos objetivos. Na medida em que os problemas surgem, vão sendo discutidos e resolvidos um a um. A busca da solução envolve a análise dos blocos de comando, a experimentação, a análise dos resultados, dos erros, a formulação e verificação de hipóteses e a pesquisa do tema central do jogo, até que a solução seja encontrada. Importante durante esta etapa auxiliar os alunos nas dificuldades com os códigos, na inserção de imagens e outros recursos, orientando quanto à importância da escrita correta. Essa etapa pode abranger mais aulas dependendo do OA que está sendo desenvolvido e das dificuldades dos alunos.
Resolução dos Problemas e Desenvolvimento do conteúdo matemático envolvido: A partir dos problemas levantados, os conteúdos matemáticos e outros interdisciplinares necessários à solução são discutidos e abordados, até que a solução seja encontrada.	
Análise crítica das soluções: Etapa em que as diferentes possíveis soluções encontradas são discutidas e analisadas. Abre a possibilidade da análise crítica, da argumentação lógica e do compartilhamento de ideias.	Fase 7 (2h/aula): Avaliação, teste e compartilhamento dos OA. Assim que os projetos estiverem concluídos, realizar os últimos testes e avaliações. Uma dupla pode testar o projeto da outra para detectar possíveis erros e ajudar na correção. Finalizar com o compartilhamento dos projetos no <i>site</i> do <i>Scratch</i> em contas individuais que os alunos podem criar, ou em uma conta da escola.

Quadro 1 – Etapas da Criação de OA no *Scratch* na perspectiva da MM

Fonte: elaborado pelos autores a partir do referencial teórico e das experiências desenvolvidas na escola

Desde que essa metodologia vem sendo aplicada na escola, foi possível produzir OA sobre diversos conteúdos que são utilizados por outros estudantes nas aulas de Informática. Os estudantes autores são organizados em duplas e definem o tipo de objeto que desenvolverão, para qual faixa etária, qual o conteúdo que será abordado, suas características, entre outros aspectos. As questões que compõem os OA são também elaboradas pelos próprios estudantes com base em pesquisas sobre os conteúdos.

Durante a criação do objeto, os problemas surgem e são discutidos até que a solução seja encontrada. Como exemplo, podemos citar o trabalho de criação de *quiz*. Neste estudo, os alunos definiram durante a Fase 5 da pesquisa exploratória que seriam criados *quizzes* sobre um tema curricular que seria definido por cada uma das duplas de trabalho. Além das perguntas que comporiam o *quiz* e da definição das respostas corretas pensadas durante a elaboração do roteiro do jogo, no momento da criação no *Scratch*, surgiram problemas do tipo: como elaborar o código para que o personagem interaja com o jogador fazendo as perguntas e fornecendo o *feedback*?

A busca das soluções envolveu a experimentação dos blocos. Inicialmente, os alunos experimentaram o bloco DIGA para inserir a pergunta. Ao realizar os testes, perceberam que o uso desse bloco não possibilita a inserção de uma resposta pelo jogador. Foi necessário, portanto, buscar outro bloco que permitisse esse tipo de interação. A instrutora auxiliou os estudantes na análise dos blocos por meio de questionamentos como: há algum outro bloco que transmita a ideia de dizer ou falar alguma coisa? Qual o bloco que mais se aproxima da ideia de fazer uma pergunta? Com esses questionamentos em mente, os estudantes analisaram os blocos disponíveis e levantaram a hipótese de que o bloco PERGUNTE seria o mais adequado. Os testes validaram a hipótese de que esse bloco permitiria a inserção de uma resposta do jogador em uma caixa de diálogo.



(a)



(b)

Figura 1 – Primeira parte do código para criação do *quiz* Jogo da Matemática

Fonte: Portfólio de atividades criadas pelos alunos no Scratch (MIT, 2007) como dados da pesquisa

A Figura 1 ilustra como ficou a primeira parte do código elaborado em um dos trabalhos. Na Figura 1(a), é possível ver o resultado do código utilizado e ilustrado pela Figura 1(b).

Outro problema levantado durante a criação do jogo foi: o que acontecerá se o usuário responder corretamente? A busca da solução envolveu pensar que a resposta correta geraria uma mensagem de PARABÉNS, uma pontuação e o avanço no jogo para a pergunta seguinte. Assim, o código da pergunta necessitou de complemento.

As condicionais simples e as variáveis passaram a ser exploradas. A instrutora auxiliou os alunos a criar um placar por meio de questionamentos como: a pontuação em um placar é algo fixo ou variável? O que significa dizer que algo é variável? O que pode fazer um placar variar? Assim, os alunos foram ajudados a criar a variável PLACAR, desenvolvendo uma noção simples de variáveis. Os blocos referentes a VARIÁVEIS no *Scratch* foram lidos e experimentados a fim de verificar seus efeitos na programação. O código da pergunta foi complementado utilizando a condicional simples, que também foi apresentada aos alunos por meio de experimentações e análise dos resultados. Após as discussões que permitiram a abordagem de conteúdos matemáticos como variáveis e condicionais, em uma linguagem adequada aos alunos, os estudantes chegaram ao código ilustrado na Figura 2.

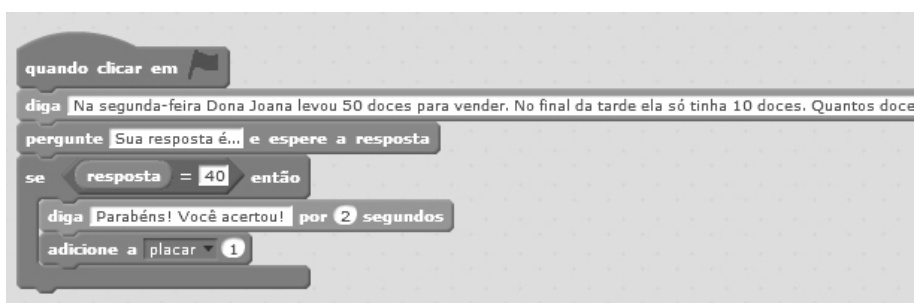


Figura 2 – Segunda parte do código para criação do quiz Jogo da Matemática
Fonte: Portfólio de atividades criadas pelos alunos no Scratch (MIT, 2007) como dados da pesquisa

Na sequência, o problema foi: o que acontecerá caso o jogador insira uma resposta equivocada? Nesse caso, foi preciso analisar que nessa situação a pergunta deveria ser repetida até que a resposta fornecida fosse a correta. O bloco REPITA ATÉ QUE foi acrescentado ao código, conforme ilustra a Figura 3.



Figura 3 – Terceira parte do código para criação do quiz Jogo da Matemática
Fonte: Portfólio de atividades criadas pelos alunos no Scratch (MIT, 2007) como dados da pesquisa

O que fazer para que a próxima pergunta fosse apresentada caso o usuário acertasse a resposta foi o próximo problema. A instrutora explicou que seria necessário “informar” a todos os elementos do jogo que a resposta inserida foi a correta, e que a próxima pergunta deveria ser apresentada. O bloco ENVIE A TODOS foi discutido com os alunos. Também foi necessário trabalhar com os operadores envolvendo igualdade. A Figura 5 mostra como ficou a programação básica do jogo de quiz.

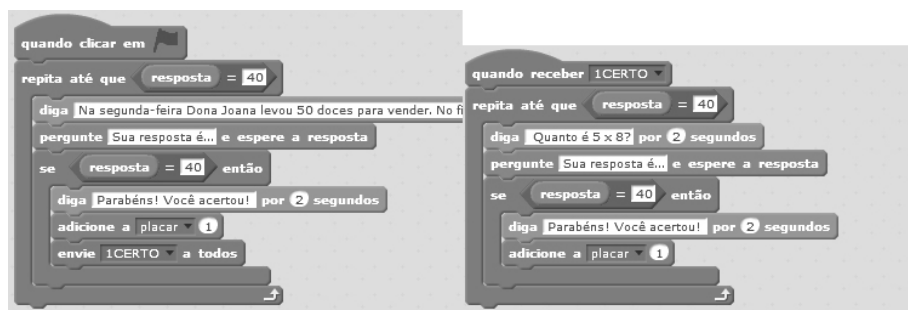


Figura 4 – Quarta parte do código para criação do quiz Jogo da Matemática

Fonte: Portfólio de atividades criadas pelos alunos no Scratch (MIT, 2007) como dados da pesquisa

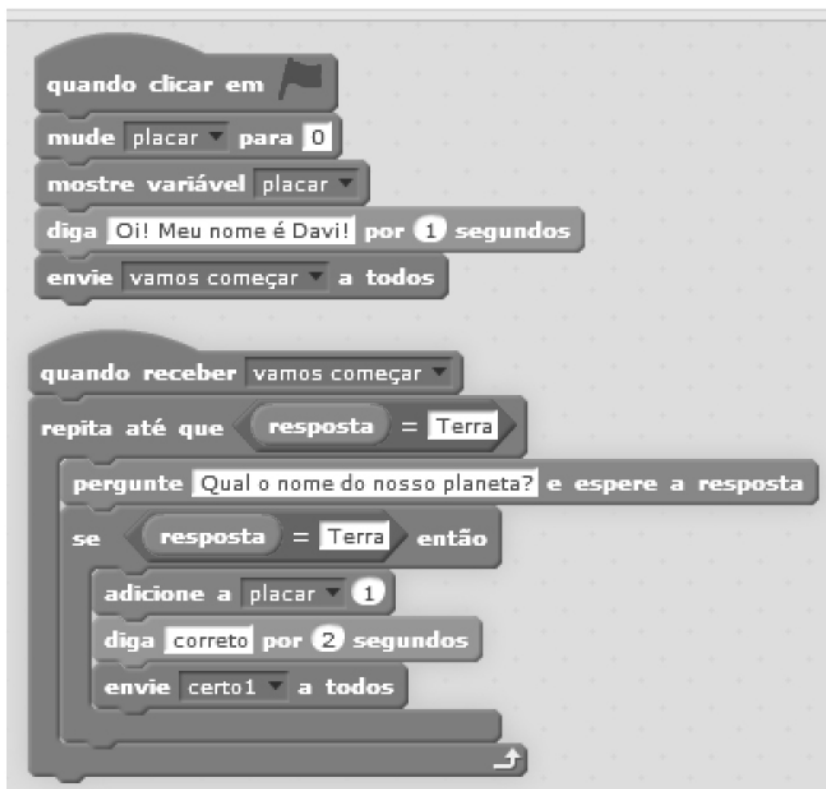
Elaborado o código básico, cada dupla passou a personalizar seu *quiz* conforme desejado. Cada jogo foi composto por no mínimo 10 questões. Os temas foram variados. Algumas duplas trabalharam com conteúdos da Matemática, como foi o caso da dupla responsável por elaborar o Jogo da Matemática ilustrado pelas Figuras 1, 2, 3 e 4, que criaram um *quiz* composto por 12 problemas matemáticos de autoria própria. Outros alunos criaram *quiz* com questões sobre Sistema Solar, história da cidade, preservação ambiental, entre outros temas, de modo que houve pesquisa e interdisciplinaridade durante todo o trabalho que também explorou com intensidade a Língua Portuguesa, pois se exigiu a escrita adequada.

Em termos do conteúdo Matemático envolvido, foram exploradas de forma lúdica, e por meio de uma linguagem adequada às crianças, as estruturas condicionais simples, igualdade, noções de variáveis e pensamento algébrico. O raciocínio lógico, o levantamento e verificação de hipóteses, a experimentação, a análise dos resultados estiveram presentes durante todo o trabalho, na medida em que os estudantes pensavam em possíveis códigos que resultariam nos seus objetivos. Além do código básico, cada dupla necessitou aprimorar seus códigos conforme a necessidade de mudanças de planos de fundo, inserção de sons, movimentos dos personagens, entre outros recursos. Essa personalização dos jogos envolveu outros problemas que foram discutidos e analisados com os alunos, até que a solução fosse encontrada.

Na etapa de Análise crítica das soluções, os projetos foram testados pelos autores e também pelos colegas da turma, sendo que as duplas testaram os jogos de outros autores e ajudaram a identificar problemas, tanto no código quanto em relação à elaboração das perguntas que, algumas vezes, não estavam claras ou necessitavam de correções em relação às respostas que estavam definidas como corretas equivocadamente. Dessa forma, e também nas etapas anteriores durante a criação dos jogos, os alunos desenvolveram aspectos sociais, como a cooperação, a colaboração, a exposição de ideias, a argumentação, e também emocionais, como autonomia, criatividade e protagonismo. Além desse momento de compartilhamento e avaliação dos projetos, houve também a análise de como ficaram os códigos dos grupos e a discussão das soluções encontradas. A exemplo disso, uma das duplas elaborou o código base para o *quiz*, conforme apresentado na Figura 5. Na Figura 5(a), notamos o resultado da programação elaborada, e na Figura 5(b), observamos algumas diferenças em relação ao código utilizado no Jogo da Matemática. Nesse *quiz*, os autores dispensaram o uso do bloco DIGA, que no exemplo anterior foi utilizado antes do bloco PERGUNTE. Nesse caso, a pergunta foi inserida diretamente no bloco PERGUNTE. Essa situação gerou discussões como: é necessário utilizar o bloco DIGA antes de PERGUNTE? Que diferença faz no jogo o uso desse bloco? Há algum caso especial em que seria interessante utilizar esse bloco antes do PERGUNTE?



(a)



(b)

Figura 5 – Parte do código elaborado para criação do quiz Sistema Solar

Fonte: Portfólio de atividades criadas pelos alunos no Scratch (MIT, 2007) como dados da pesquisa

Por meio desses questionamentos, uma das duplas comentou que utilizou o bloco DIGA para criar um *quiz* de múltipla escolha, como ilustrado na Figura 6.



(a)



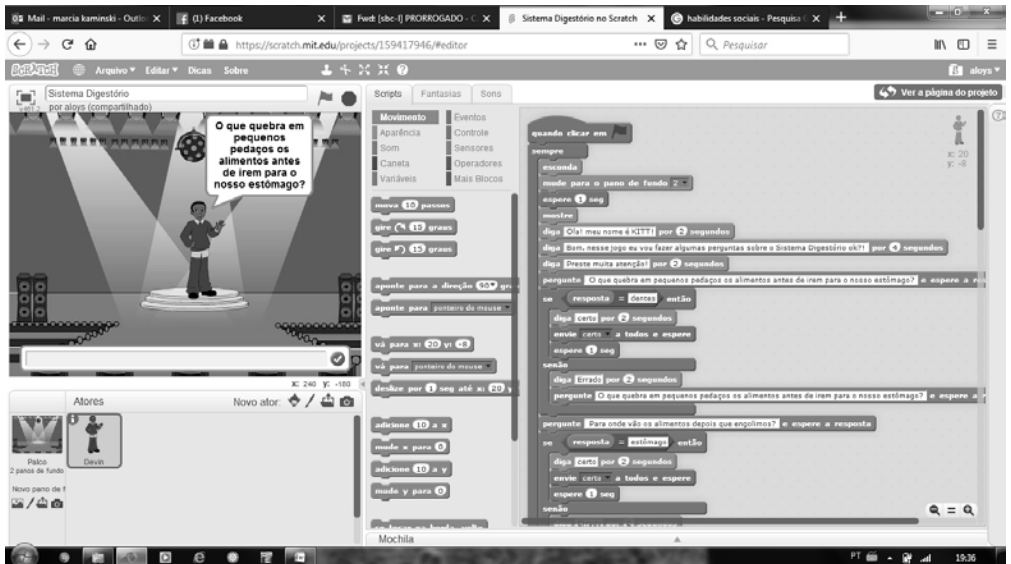
(b)

Figura 6 – Parte do código elaborado para criação do quiz #Meio Ambiente

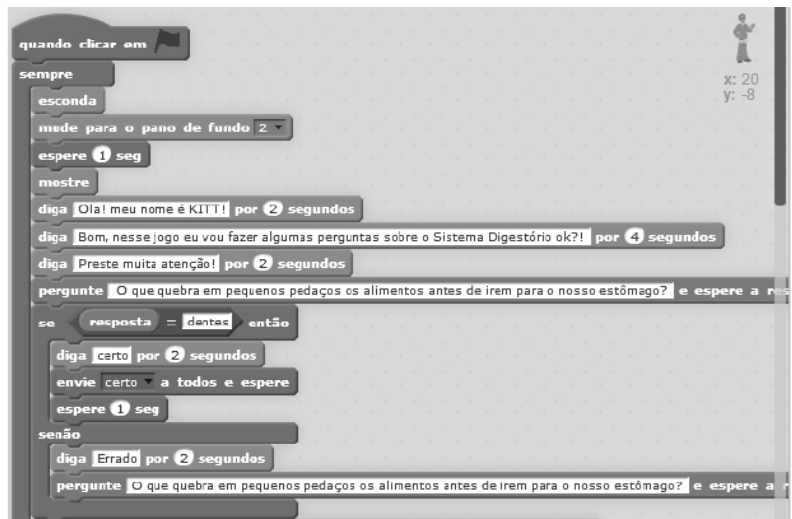
Fonte: Portfólio de atividades criadas pelos alunos no Scratch (MIT, 2007) como dados da pesquisa

Na Figura 6(a), é ilustrada uma das telas do jogo #Meio Ambiente, e na Figura 6(b), parte do código que o origina. Nesse caso, o bloco DIGA foi utilizado para fornecer as alternativas de forma mais lenta, de modo a permitir a melhor leitura e compreensão da pergunta e das alternativas.

Outra equipe utilizou o bloco SE e SE NÃO, para que, além da mensagem de acerto, fosse fornecida ao jogador também uma mensagem de erro. Nesse caso, o bloco REPITA ATÉ não foi necessário, uma vez que a repetição da pergunta no caso da resposta errada estava já inserida como uma condição se a resposta não fosse a correta. A Figura 7(a) ilustra uma das telas do jogo Sistema Digestório e parte do código elaborado na Figura 8(b). Essa análise e comparação dos códigos ajudou os estudantes a perceber que há mais de uma forma de elaborar um *quiz*, e que o código mais adequado dependerá, em muito, dos objetivos e do estilo escolhido pelo(s) autor(es).



(a)



(b)

Figura 7 – Parte do código elaborado para criação do quiz Sistema Digestório
Fonte: Portfólio de atividades criadas pelos alunos no Scratch (MIT, 2007) como dados da pesquisa

No caso da escola, as atividades acontecem sempre nas aulas de Informática e, portanto, no espaço do laboratório. Porém, caso a escola não disponha desse espaço e tenham recursos físicos

limitados em termos de TDIC, é possível realizar o trabalho utilizando a Rotação por Estações de Aprendizagem ou o Laboratório Rotacional. Na primeira opção, segundo Moran e Bacich (2015), o professor prepara a sala de aula com estações de trabalho, em que os alunos, em grupo, trabalham em sistema de rodízio passando por todas as estações, sendo que uma delas deve incluir atividades *online*. Já na segunda opção, uma das estações é o próprio laboratório ou outro espaço que disponha de equipamentos eletrônicos, ainda que em quantidade reduzida. Nesses casos, o professor pode dividir o trabalho de forma que, enquanto alguns alunos elaboram seus roteiros e pesquisam o tema, outros trabalham na produção do OA no *Scratch*. Equipes com maior número de integrantes também podem ser formadas, de acordo com a realidade.

CONCLUSÕES

Desde que iniciado o trabalho com o desenvolvimento do PC, foram experimentadas diferentes formas de abordagem. A produção de OA pelos estudantes tem se mostrado uma estratégia interessante na medida em que coloca os alunos como protagonistas, desenvolvendo aspectos importantes à formação da cidadania crítica.

Quando vinculados a uma abordagem metodológica consistente como a MM, essa atividade torna-se ainda mais produtiva, considerando que, seguindo passos bem definidos, é possível auxiliar os alunos em um processo de ensino e aprendizagem contextualizado, crítico e reflexivo.

A abordagem de PC por meio da MM aplicada neste estudo foi descrita de forma a permitir reprodutibilidade e adaptações para diferentes contextos educacionais. Os exemplos aqui demonstrados são apenas para ilustrar o trabalho que vem sendo desenvolvido, de modo que, em outros momentos, os alunos já produziram outros tipos de OA que não *quiz*, e outros tipos de jogos que não os educacionais. Cada um deles traz diferentes formas de contribuições para o desenvolvimento dos estudantes, sendo que os OA têm um

potencial de integrar de forma mais intensa os conteúdos curriculares, enquanto que a produção de outros tipos de jogos favorece, em muito, o desenvolvimento da criatividade. Cabe ao professor definir seus objetivos para cada atividade, sendo que a variedade delas é um fator importante a ser considerado em seu planejamento.

Em qualquer caso, a MM é um encaminhamento metodológico que tem se mostrado adequado para o trabalho com PC e que potencializa essas atividades. Como trabalhos futuros, o estudo mais aprofundado da MM e o desenvolvimento de atividades cujos problemas sejam propostos pelos alunos de 5º ano estão sendo considerados, bem como a aplicação dessa abordagem também com estudantes de 4º ano.

REFERÊNCIAS

Almeida, L. M. W., & Silva, A. (2010). Por uma Educação Matemática Crítica: a Modelagem Matemática como alternativa. *Educação Matemática Pesquisa*, São Paulo, v. 12, n. 2, p. 221-241. Recuperado em 30 jan. 2019, de <https://is.gd/dosoBI>.

Blikstein, P. (2008) O pensamento computacional e a reinvenção do computador na educação. Recuperado em 8 jan. 2019, de goo.gl/57eqxG.

Borssoi, A. H., & Almeida, L. M. W. (2004). Modelagem matemática e aprendizagem significativa: uma proposta para o estudo de equações diferenciais ordinárias. *Educação Matemática Pesquisa*, São Paulo, v. 6, n. 2, p. 91-121, mar. Recuperado em 24 jan. 2019, de <https://revistas.pucsp.br/index.php/emp/article/view/4689>.

Brackmann, C. P. (2017). *Desenvolvimento do Pensamento Computacional através de Atividades Desplugadas na Educação Básica*. 226 f. Tese de Doutorado em Informática na Educação, Programa de Pós-Graduação em Informática na Educação (PPGIE) do Centro Interdisciplinar de Novas Tecnologias na Educação (CINTED) da Universidade Federal do Rio Grande do Sul (UFRGS), Porto Alegre, RS. Recuperado em 16 jan. 2019, de <http://www.lume.ufrgs.br/handle/10183/172208>.

Brasil. Ministério da Educação (2018). Secretaria da Educação Básica. *Base nacional comum curricular*. Brasília, DF. Recuperado em 30 jan. 2019, de http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518_ver-saofinal_site.pdf.

Burak, D. (1992). *Modelagem Matemática: Ações e interações no processo de ensino-aprendizagem*. 1992. 460 f. Tese de doutorado, Curso de Educação, Universidade Estadual de Campinas, Campinas. Recuperado em 5 jan. 2019, de <https://bit.ly/2yp8h7Z>.

Burak, D., & Klüber, T. E. (2016). Considerações sobre a modelagem matemática em uma perspectiva de educação matemática. *Revista Margens Interdisciplinar*, v. 7, n. 8, p. 33-50, maio 2016. Recuperado em 29 jan. 2019, de <https://is.gd/Aah3tC>.

Ferri, J., & Rosa, S. D. S. (2016). Como o Ensino de Programação de Computadores Pode Contribuir Com a Construção de Conhecimento na Educação Básica Uma Revisão Sistemática da Literatura. *Revista Novas Tecnologias na Educação*, v. 14, n. 2, dez., 2016. Recuperado em 20 jan. 2019, de <http://goo.gl/Q1PriW>.

Filatro, A. (2008). *Design Instrucional na prática*. São Paulo: Pearson Education do Brasil, 173 p.

França, R. S. de, Ferreira, V. A. D. S., Almeida, L. C. F. D, & Amaral, H. J. C. D. (2014). A disseminação do pensamento computacional na educação básica: lições aprendidas com experiências de licenciandos em computação. *Anais do XXXIV Congresso da Sociedade Brasileira de Computação*. (CSBC) p. 1505-1514. Recuperado em 8 jan. 2019, de goo.gl/CQzPQP.

Gouvêa, G. (2015). A Divulgação da Ciência, da técnica e Cidadania e a sala de aula. In Giordan, M. & Cunha, M. B. da (Org). *Divulgação Científica na Sala de Aula: perspectivas e possibilidades*. Ijuí: Unijuí, p. 14-36.

Kaminski, M. R; & Boscaroli, C. (2018a). Production of Scratch Learning Objects by Elementary School Students. *Anais do LACLO - XIII Conferência Latino-americana de Tecnologias de aprendizagem*, p. 1-8. Recuperado em 14 jan. 2019, de <http://cleilaclo2018.mackenzie.br/docs/LACLO/FULL/183877.pdf>.

Kaminski, M. R; & Boscaroli, C. (2018b). Criação de jogos digitais na perspectiva de introdução à Modelagem Matemática nos anos iniciais. *Revista Thema*, v. 15, n. 4, p. 1538-1548. Recuperado em 14 jan. 2019, de <https://bit.ly/2EjJ5Gj>.

Kaminski, M. R., & Boscaroli, C. (2018c). Modelagem Matemática e Pensamento Computacional no 5º Ano do Ensino Fundamental. *Anais do VIII Encontro Paranaense de Modelagem na Educação Matemática*, p. 1-6. Recuperado em 14 jan. 2019, de <https://bit.ly/2TR1BZw>.

Massachusetts Institute of Technology (MIT)(Org.)(2007). *Scratch*. Recuperado em 5 jan. 2019, de <https://scratch.mit.edu/>.

Moran, J., & Bacich, L. (2015). Aprender e ensinar com foco na educação híbrida. *Revista Pátio*, n. 25, p. 45-47, jun. Recuperado em 5 jan. 2019, de <http://goo.gl/1oBHmQ>.

Santos, P. R. D. (2006). O Ensino de Ciências e a Ideia de Cidadania. *Mirandum*, ano X, n. 17, p. 25-34 – IJI – Universidade do Porto (Portugal): Ed. Mandruvá. Recuperado em 8 jan. 2019, de <http://www.hottopos.com/mirand17/index.htm>.

Saviani, D. (1999). *Escola e democracia: teorias da educação, curvatura da vara, onze teses sobre educação e política!* 32ª ed. Campinas, SP: Autores Associados, 99 p.

Saviani, D. (2010). Ciência e educação na sociedade contemporânea: desafios a partir da pedagogia histórico-crítica. *Faz Ciência* (UNIOESTE. Impresso), v. 1, n. 16, p. 13-36, jul/dez. Recuperado em 19 jan. 2019, de <https://bit.ly/20J4zwG>.

Silva, A. M. D. S., Moraes, D. A. S. D., & Souza, B. S. C. F. (2014). Meio Ambiente e Sustentabilidade: ações pedagógicas no Ensino Fundamental com uso do Scratch. *Tecnologias, Sociedade e Conhecimento*, Campinas – SP, v. 2, n. 1, p. 63-84, 2014. Recuperado em 5 jan. 2019, de <https://bit.ly/2zNdNTj>.

Vecchia, R. D. (2012). *A Modelagem Matemática e a Realidade do Mundo Cibernético*. 2012. 275 f. Tese de doutorado, Curso de Educação Matemática, Instituto de Geociências e Ciências Exatas, Universidade Estadual Paulista, Rio Claro – SP, 2012. Recuperado em 17 jan. 2019, de <https://is.gd/BYGcMj>.

Vecchia, R. D., & Maltempi, M. V. (2014). O Modelo na Modelagem Matemática na Realidade do Mundo Cibernético. *Acta Scientiae*, Canoas, v. 16, n. 4, p. 199-213, 2014. Recuperado em 4 jan. 2019, de <https://bit.ly/2IXPl5>.

Wing, J. (2006). Computational Thinking. *Communications of the ACM*, v. 49, n. 3, p. 33-35, mar. Recuperado em 29 jan. 2019, de <https://bit.ly/1r4smq0>.

CAPÍTULO 6

DESENVOLVIMENTO E IMPACTO DA INTEGRAÇÃO CURRICULAR DE PROGRAMAÇÃO E PENSAMENTO COMPUTACIONAL EM TODAS AS DISCIPLINAS E SÉRIES DO ENSINO FUNDAMENTAL DA FUNDAÇÃO TORINO ESCOLA INTERNACIONAL

*Ana O. Rodrigues
Ana Márcia A. M. Paiva
Sabrina M. de Oliveira
Márcia L. A. A. Naves*

INTRODUÇÃO

O mundo hoje vive um momento de transição. Acreditamos que o cenário atual impõe o desenvolvimento de competências para tomar decisões, lidar com um volume e disponibilidade de informações nunca antes visto. Se faz necessária uma discussão contínua para a garantia de uma Educação Integral, que vise “à formação e ao desenvolvimento humano global do indivíduo” (BNCC, 2018).

A Educação Integral pressupõe uma “construção intencional de processos educativos que promovam aprendizagens sintonizadas com as necessidades, possibilidades e interesses dos alunos” (BNCC, 2018), respeitando a cultura, contextos e ambientes individuais e criando um ambiente propício à troca de experiências, experimentos e colaboração.

Ademais, estamos inseridos em uma era digital em que o uso de ferramentas tecnológicas faz parte do cotidiano de todos. Nossas crianças, nativos digitais, nascem nesse mundo de forma intensa e permanente. Sob outra perspectiva estão os educadores, que foram imersos nesse ambiente, mas não tiveram o mesmo contato com tais ferramentas tecnológicas desde a infância e as adaptações são necessárias para que possam sintonizar de forma efetiva com esse universo para benefício de uma educação mais integral.

Dessa forma, faz-se necessária a construção de habilidades de inteligência tecnológica de forma consciente e construtiva, de forma que as crianças sejam responsáveis não somente pelo domínio em relação ao uso das novas ferramentas disponíveis, mas também em relação às novas formas de raciocínio e metodologias de resolução de problemas os quais a sociedade impõe todos os dias. Os professores das diversas disciplinas devem estar inseridos em

todo o processo, para que haja consonância entre necessidades, interesses e possibilidades.

Esse é o objetivo do trabalho desenvolvido na Fundação Torino Escola Internacional, que se pauta por uma educação humanista e oferece uma educação multicultural, que se encontra em total consonância com o compromisso com a Educação Integral (BNCC, 2018) e com as referências em Tecnologia e Computação (CIEB, 2018), e que será detalhado nas próximas seções deste artigo:

- Histórico e Motivação do Ensino de Programação e Pensamento Computacional.
- Metodologia, apresentando os estágios da implantação do projeto:

Fase 1: Formação continuada dos professores;

Fase 2: Estudo das ementas e definição dos projetos das disciplinas;

Fase 3: Desenvolvimento dos tutoriais, com exemplos de projetos desenvolvidos pelos alunos e professores, em diversas áreas do conhecimento, como Artes, Inglês, História e Geografia, Italiano, Matemática, Música, Educação Física, Português, Religião, Ciências e Espanhol;

Fase 4: Acompanhamento da implantação em sala de aula.

- Avaliação do projeto:
 - Avaliação qualitativa pelos professores e sugestões de melhorias;
 - Avaliação quantitativa do impacto da implantação desse projeto nas notas dos alunos e suas possíveis causas, e
- Conclusão: Propostas de melhorias e próximos passos baseadas no impacto do projeto, aprendizados durante a implantação e sugestões e comentários da avaliação anônima qualitativa feita pelos professores.

HISTÓRICO E MOTIVAÇÃO DO ENSINO DE PROGRAMAÇÃO E PENSAMENTO COMPUTACIONAL

A importância da introdução de inteligência tecnológica nos currículos escolares tem sido a motivação para diversos movimentos internacionais e nacionais desde a década de 80, com o início da popularização do computador pessoal.

Em 2005, a Organização das Nações Unidas para a Educação, a Ciência e a Cultura (*United Nations Education, Scientific, and Cultural Organization* – Unesco), durante o “*High-Level Colloquium on Information Literacy and Lifelong Learning*” em Alexandria, no Egito, explorou a questão da internacionalização e institucionalização do Letramento Digital (“*Information Literacy*”) e do aprendizado ao longo da vida (“*Lifelong Learning*”) como fatores essenciais ao desenvolvimento das nações, chamados de “Faróis da Sociedade da Informação”, e declarou uma série de recomendações às nações e às sociedades, governamentais ou não, no sentido de fomentar o desenvolvimento das Competências Informacionais e o aprendizado ao longo da vida como premissas para o desenvolvimento educacional e social na sociedade da informação.

A então chamada “Proclamação de Alexandria” declarou o Letramento Digital como a base da aprendizagem ao longo da vida, em todas as disciplinas, em todos os ambientes de aprendizagem e em todos os níveis de ensino, mesmo considerando a natureza e desenvolvimento de países diferentes, pois se preocupa com o ensinar e aprender sobre toda a gama de fontes e formatos de informação. Declara que a competência informacional capacita as pessoas para todos os caminhos que possam buscar, sejam pessoais, sociais, ocupacionais ou educacionais, além de considerar essa habilidade um direito humano básico no mundo digital.

O Letramento Digital, nessa concepção, abrange as competências para reconhecer as necessidades informacionais e localizar, avaliar, aplicar e gerar informação dentro de contextos culturais e sociais, além de promover o pensamento crítico e as habilidades interpretativas dentro de uma sociedade complexa.

Segundo Paiva (2010, p. 2325), o

Letramento digital é entendido como capacidade de buscar informações por meio das TDIC, desenvolver a consciência crítica e a habilidade de análise das reais possibilidades que elas trazem para o cotidiano, promover o autodesenvolvimento e a aprendizagem [...] propiciando uma maior participação social e trazendo soluções para si próprios e para seu entorno.

Além disso, o letramento digital contribui, principalmente, para a educação integral do indivíduo, tornando-o mais autônomo e capaz de construir seu próprio caminho em busca do seu projeto de vida, seja profissional, pessoal ou social.

As competências informacionais, dentro das novas perspectivas mediadas pelas TDIC, constituem níveis de Letramento Digital que vão desde o acesso e recuperação de informações, passando pelas fontes de informação, às estratégias escolhidas e usadas para o acesso e uso da informação, ao controle da informação, à construção do próprio conhecimento, à capacidade de ampliação do conhecimento (saber aprender) e, por fim, à geração de conhecimento (Bruce, 1997; Unesco, 2006, 2007).

Como podemos observar na Figura 1, quanto maior o nível de avanço nas competências, da categoria 1 até a categoria 7, maior a consolidação do Letramento Digital, menor é o foco na Tecnologia em si, e ressalta a construção da base de conhecimento e o uso da informação para gerar novos conhecimentos, gerar novas aprendizagens autônomas e gerar soluções e inovações.

Observa-se que a Categoria 1 está ligada basicamente ao **acesso** às tecnologias e à informação e está muito focada na tecnologia em si e na exploração de suas funcionalidades e da gama de informações que estão disponíveis; na medida em que o acesso foi bastante explorado, o indivíduo já inicia um processo de busca e seleção de fontes de informação, pois observa diferentes formas de acesso e diferentes níveis de informação; a partir daí, o indivíduo é capaz de

criar as próprias estratégias para a busca, e caminhos para aquisição das informações, de forma mais personalizada e eficaz; outra competência necessária diante do grande volume de possibilidades é o controle da informação que diz respeito ao resgate, armazenamento e recuperação da informação quando for necessária, de forma segura e eficaz; a partir das categorias 5, 6 e 7, o foco do indivíduo é o uso da informação para a construção de um conhecimento com olhar e análise críticos, tornando-o mais capaz de fazer inferências, comparações, ampliações e uso criativo do conhecimento, fazendo um bom uso para a resolução de problemas e geração de ideias e inovações que sejam desejáveis, factíveis e inovadoras (Paiva, 2010).

	ACESSO Categoria 1	FONTES Categoria 2	PROCESSO Categoria 3	CONTROLE Categoria 4	CONSTRUÇÃO Categoria 5	EXTENSÃO Categoria 6	SABER Categoria 7
Foco Principal	Tecnologia da Informação	Fontes de Informação (uso de diversas de fontes)	Informação como processo (estratégias)	Controle da Informação (resgate, seleção)	Uso da Informação (análise crítica)	Uso da Informação (intuição)	Uso da Informação valores)
Segundo Plano	Exploração da Informação	Tecnologia da Informação	Uso da Informação	Uso da Informação	Base de Conhecimento	Base de Conhecimento	Base de Conhecimento
Terceiro Plano	Uso da Informação	Uso da Informação	Tecnologia da Informação	Tecnologia da Informação	Tecnologia da Informação	Tecnologia da Informação	Tecnologia da Informação

Figura 1 – Sete categorias da competência informacional e seus níveis de concentração

Fonte: Paiva, 2010

Essa mudança gradual de foco acontece de forma natural se as crianças e jovens forem estimulados de forma consistente e consciente a desenvolverem tais habilidades e competências, com um currículo de referência que estimule e propicie um desenvolvimento integral do aluno, respeitando-se as diferenças, experiências individuais, capaz de torná-los mais colaborativos entre si na construção de um conhecimento com mais significado para cada um e para a coletividade.

O ensino e aprendizagem de programação e pensamento computacional no âmbito escolar, integrado ao currículo, traz oportunidades para o desenvolvimento de todos os níveis de competência informacional e produz um arcabouço, uma estrutura, para que

nossas crianças e jovens construam suas próprias soluções a partir dos problemas encontrados e busquem soluções pessoais e para o seu entorno, independentemente do que a sociedade impõe como consumo, para além do tecnológico.

A Sociedade Brasileira de Computação (SBC) corrobora esse pensamento e considera a aprendizagem da computação como essencial na sociedade contemporânea. As estratégias usadas e escolhidas pelos indivíduos para a solução de problemas se dão em boa medida por meio da aquisição dessas competências e conhecimentos básicos de computação. Dessa forma, os indivíduos poderão fazer escolhas, desenvolver soluções próprias para novos problemas e outros que não foram ou não eram possíveis de serem solucionados anteriormente (SBC, 2017).

A publicação de *Mindstorms* de Seymour Papert, em 1980 (Papert, 1980), com os princípios da linguagem de programação LOGO, apresenta pela primeira vez a ideia de que o ensino de programação é uma ferramenta poderosa de educação, principalmente para compreensão da matemática e para o desenvolvimento do pensamento crítico, da experimentação e da solução de problemas. Papert defende que esse aprendizado, principalmente quando associado a máquinas físicas como o robô LOGO, permite um aprendizado mais profundo, relacionado ao senso de autoimportância da criança. Suas teorias ficariam limitadas à pesquisa e algumas poucas escolas privilegiadas que têm acesso à computadores na década de 80.

Em 15 de maio de 2007, o Laboratório *Lifelong Kindergarten* do MIT lança oficialmente a plataforma Scratch (MIT, 2007). Essa ferramenta rapidamente ganha projeção internacional, sendo usada por milhões de crianças e jovens no mundo todo para aprender pensamento computacional e programação de computadores. Escolas do mundo inteiro adotam programas parciais ou extracurriculares que incluem o Scratch em seus programas. Novamente, a realidade da maioria dos países em desenvolvimento é limitadora para o desenvolvimento consistente de integração curricular desse tipo de aprendizado.

Em 2013, a plataforma Code.org (2013) é lançada com cursos, tutoriais e diversos níveis de profundidade e faixas etárias. A iniciativa da “Hora do Código” (Code.org, 2018) introduz o ensino de

programação em mais de 180 países, com mais de 651 milhões de participantes. Eventos como esse são apoiados por políticos como o ex-Presidente dos EUA Barack Obama, que cria o programa “*Computer Science for All*” em janeiro de 2016 nos EUA. Segundo Barack Obama, “Na nova economia, conhecimento computacional não é mais uma habilidade opcional, é uma habilidade básica...” (White House USA, 2016). O maior acesso a computadores e laboratórios de informática, além do desenvolvimento dessa plataforma gratuita, faz com que iniciativas como essa comecem a ser comuns em escolas particulares, embora ainda de forma individual, implementadas por professores interessados.

Em 2014, no Brasil, é criado o Programaê!, uma iniciativa da Fundação Lemann e da Fundação Telefônica Vivo que facilita a introdução da linguagem de programação e pensamento computacional nas práticas pedagógicas, garantindo subsídios para que os alunos sejam protagonistas desse processo. O Programaê! disponibiliza um Guia para Construção do Pensamento Computacional para orientar educadores sobre como integrar a programação dentro do ensino fundamental I e II (Fundação Telefônica Vivo e Fundação Lemann, 2014).

Em 2014, o Ministério da Educação da Itália (*Ministero dell’Istruzione, dell’Università e della Ricerca* – Miur), em colaboração com o Consórcio Nacional Interuniversitário de Tecnologia da Informação (CINI), lança a iniciativa “*Programma il futuro*” como parte do programa #labuonascuola, dotando as escolas com um conjunto de ferramentas para oferecer aos alunos os conceitos básicos da ciência da computação. Com a circular Miur 9759 de outubro de 2015, a Itália se torna um dos três primeiros países da Europa, junto ao Reino Unido e à Finlândia, a oferecer programação de forma integrada ao currículo (Miur, 2014).

A Fundação Torino, por ser uma escola internacional, deve cumprir as exigências não apenas do currículo brasileiro, mas também do currículo italiano. Observando essas tendências e a importância para o futuro dos alunos, em 2015, a Fundação Torino Escola Internacional contrata a Just CODING para criar o projeto de três anos de implantação curricular de programação e pensamento computacional detalhado neste artigo.

A perspectiva interdisciplinar é defendida por Vera Cabral, consultora em educação e curadora da Bett Brasil Educar 2016:

A cada novidade contrata-se um novo professor. O de robótica, o de programação... Enquanto isso, eu, professora de Física ou de matemática, continuo dando minhas aulas como sempre dei, e a robótica, a programação ou o teatro acabam colocados como disciplinas optativas. Isso não muda nada. Para transformar efetivamente a atuação da escola, é necessário que a segmentação diminua, não que seja acentuada. (Geek, 2016, s/p).

Em 2016, o Fórum Econômico Mundial (Fórum Econômico Mundial, 2016), destaca as 8 habilidades de inteligência digital que as crianças precisam aprender, de forma que estejam melhor preparados para os impactos e desafios econômicos e sociais das tecnologias. Essa publicação ganha atenção mundial e empresas, governos e escolas começam a apoiar iniciativas que promovam o desenvolvimento destas habilidades:

- Identidade Digital: criar e gerenciar identidade online e reputação.
- Uso Digital: usar dispositivos e suportes digitais, com equilíbrio entre *online* e *offline*.
- Proteção Digital: gerenciar riscos *online*, como *cyberbullying*, conteúdos obscenos e violentos.
- Segurança Digital: detectar ameaças virtuais, como pirataria, fraudes, *malware* e usar ferramentas de segurança para a proteção de dados.
- Inteligência Emocional Digital: ser compreensivo e construir boas relações *online*.
- Comunicação Digital: comunicar e colaborar com outros usando tecnologia.
- Habilidade Digital: encontrar, avaliar, utilizar, compartilhar e criar conteúdo, e ter capacidade em pensamento computacional.

- Direitos Digitais: compreender e defender direitos pessoais e coletivos à privacidade, à propriedade intelectual, à liberdade de expressão e proteção contra discursos de ódio.

Em 2016, no Brasil, diversas escolas de programação particulares surgem em modelos diferentes. A Just CODING tem como objetivo implantar metodologias de integração curricular de programação de computadores e pensamento computacional por meio do seu programa SMART, que inclui experiências de Ciências, Matemática, Artes, Robótica e Tecnologia, sempre em uma perspectiva do Aprendizado Centrado no Aluno, como defendido por Carl Rogers (1975).

Em 2017, Mitchel Resnick, diretor do Lifelong Kindergarten Lab do MIT, onde o Scratch foi criado, lança o livro *Lifelong Kindergarten – Cultivating Creativity through Projects, Passion, Peers, and Play* (Resnick, 2017). Nele, Mitchel advoga o Aprendizado Criativo como a nova ferramenta para desenvolvimento das habilidades necessárias para ser um cidadão do mundo digital e das tecnologias. Além da utilização das ferramentas tecnológica, é fundamental que o aprendizado seja baseado em quatro “Ps” (em inglês): Projetos (*projects*), Paixão (*passion*), Companheiros (*peers*) e Brincadeiras (*play*). Também em 2017, a Prof.^a Marina Umashi Bers, coordenadora do Scratch Jr., versão do Scratch para crianças de até 7 anos, lança o livro *Coding as a playground* (Bers, 2017), defendendo a tese de que o ambiente de aprendizado deve ser mais parecido com um parquinho (*playground*) do que com um cercadinho (*playpen*).

No final de 2017, é aprovada a Base Nacional Comum Curricular (BNCC), que se compromete com a Educação Integral e preconiza que a Educação assuma uma visão “plural, singular e integral” das crianças e jovens, considerando-os “sujeitos da aprendizagem” e que promova o “acolhimento, reconhecimento e desenvolvimento pleno, em suas singularidades e diversidades”. A BNCC propõe o estímulo à aplicação do conhecimento disciplinar na vida cotidiana e à importância de dar sentido ao que se aprende. Define também que a abordagem sobre tecnologia e ensino de computação deve ocorrer de forma transversal em todas as áreas do conhecimento e componentes curriculares.

Notadamente, a competência geral número 5 explicita a necessidade de se trabalhar com o tema de tecnologias digitais de infor-

mação e comunicação (TDICs), colocando os alunos no papel de produtores de conteúdo, e não apenas consumidores passivos de tecnologia (BNCC, 2018; Cieb, 2018).

Utilizar tecnologias digitais de comunicação e informação de forma crítica, significativa, reflexiva e ética nas diversas práticas do cotidiano (incluindo as escolares) ao se comunicar, acessar e disseminar informações, produzir conhecimentos e resolver problemas. (BNCC, 2018, p. 18).

Essa definição chama a atenção de escolas e instituições governamentais para essa importante questão, fundamental para a inserção dos alunos no mundo digital de forma produtiva, crítica e consciente.

METODOLOGIA

A integração de programação de computadores e pensamento computacional ao currículo do 1º ao 8º ano do ensino fundamental da Fundação Torino ocorreu em quatro fases, cujo detalhamento encontra-se a seguir:

Fase 1: Formação continuada dos professores;

Fase 2: Estudo das ementas em conjunto com os professores;

Fase 3: Desenvolvimento dos tutoriais, e

Fase 4: Acompanhamento da implantação em sala de aula.

Fase 1: Formação continuada dos Professores

A primeira fase constitui-se em uma apresentação do projeto de integração para todos os professores de ensino fundamental I e II da Fundação Torino, e de Formação continuada específica em pensamento computacional para os professores do fundamental I e do fundamental II, detalhadas a seguir:

- Palestra Geral: Foi realizada uma palestra para todo o corpo docente e administração. Nessa palestra, foram apresentados os objetivos do projeto de integração, as metodologias utilizadas e foram descritas as vantagens de aprendizado de programação.
- Formação continuada em Pensamento Computacional: Foram feitos dois workshops em Pensamento Computacional, com o objetivo de esclarecer e engajar os professores das disciplinas envolvidas na integração, fornecendo aos professores os conhecimentos e ferramentas tecnológicas para identificação e desenvolvimento de desafios que poderiam ser resolvidos a partir de técnicas de programação de computadores, como mostra a Figura 2.



Figura 2 – Formação continuada dos professores em pensamento computacional
Fonte: autores

O workshop para formação continuada dos professores do fundamental I, de quatro horas, no Laboratório de Informática, cobriu os seguintes tópicos: introdução à Ciência da Computação, sistema

binário, apresentação da plataforma da CODE.ORG, estruturas de repetição, estrutura condicional, pensamento computacional e conceito de algoritmo. O workshop dos professores do fundamental II, de nove horas, no Laboratório de Informática, cobriu os seguintes tópicos: introdução à Ciência da Computação, binário, plataforma da Code.org, estruturas de repetição, estrutura condicional, pensamento computacional, conceito de algoritmo, introdução à JavaScript, funções, variáveis, loops com incremento, desenvolvimento de desafios, conclusões e referências adicionais.

Fase 2: Estudo das ementas e Definição de Projetos

A Fundação Torino forneceu todas as ementas trabalhadas no ensino fundamental I e II. As professoras Ana Rodrigues e Daniela Lacerda da Just CODING estudaram essas ementas com o objetivo de compreender os pontos curriculares específicos que poderiam ser trabalhados por meio da programação.

Foram então feitas duas reuniões com os professores de forma individual. Na primeira, os professores trouxeram propostas de projeto, com temas escolhidos por eles. Por exemplo, a professora de Artes propôs uma galeria de arte virtual em 3D.

Cada uma dessas propostas foi transformada em um projeto rascunho, no Scratch ou na plataforma de desenvolvimento de aplicativos (App Lab) do CODE.ORG. No caso da professora de Artes, o projeto desenvolvido foi a construção de uma galeria, em 2D, no Scratch, com “salas” diferentes para cada estilo de arte.

O passo seguinte foi uma reunião de retorno, para esclarecimento de dúvidas e aprovação dos projetos em sua forma final. No exemplo do projeto de Artes, foi explicado que um projeto 3D iria além da capacidade de programação esperada para os alunos daquela série, e que o projeto deveria ser desenvolvido em uma aula de 50 minutos. A professora concordou com as modificações, e aprovou o projeto.

Por fim, foi desenvolvido um cronograma anual, contendo quais aulas e projetos seriam desenvolvidos, com definição dos pontos correlacionados ao ensino de programação de computadores, pensamento computacional e os temas específicos dentro do currículo. Os projetos foram ordenados por complexidade e pelo momento em que o tópico estaria sendo trabalhado com o professor da matéria.

Fase 3: Desenvolvimento dos Tutoriais

O material de suporte para as aulas de programação de computadores e pensamento computacional e tutoriais para desenvolvimento dos projetos foram desenvolvidos em conjunto pela Just CODING e pela professora Ana Márcia Paiva, responsável interna pela implantação do projeto. A Figura 3 mostra os alunos desenvolvendo os projetos no laboratório de informática da Fundação Torino. As Figuras 4 e 5 mostram exemplos de projetos desenvolvidos.



Figura 3 – Alunos desenvolvendo projetos de diversas disciplinas. (1) Projeto de Artes: Construindo um quadro. (2) Professora Ana Márcia Paiva acompanhando alunos. (3) Projeto de Educação Física: Quiz do Futebol

Fonte: autores

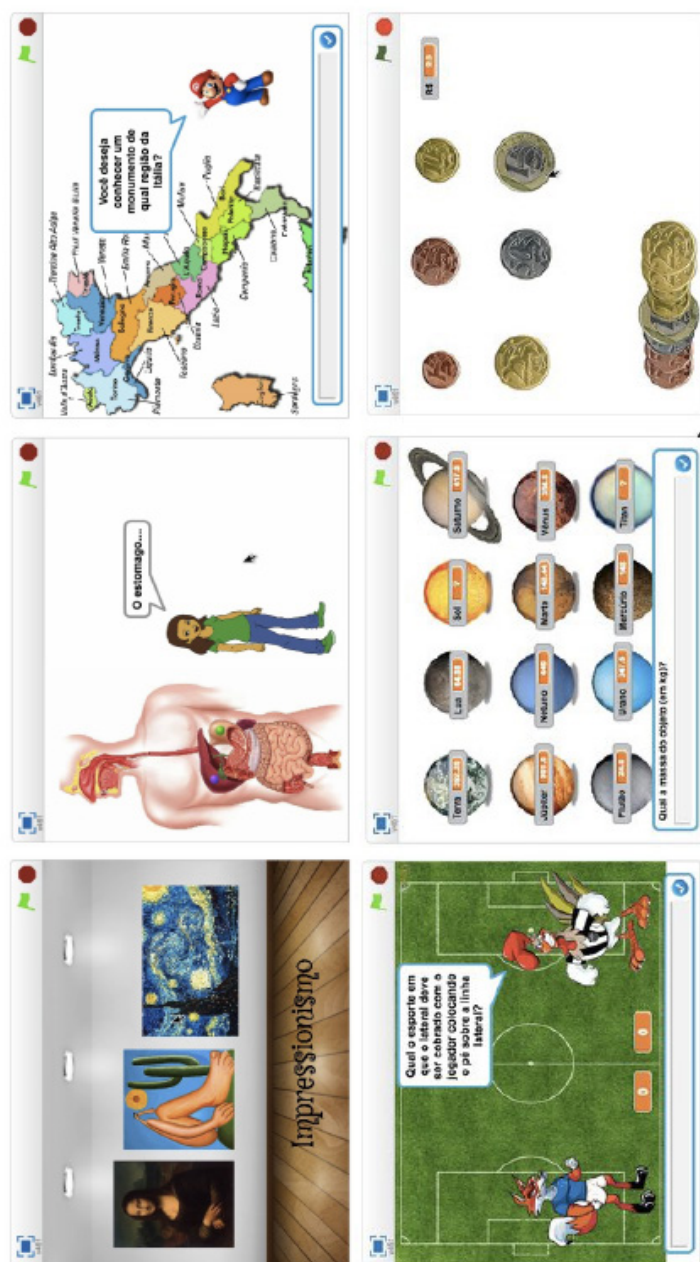


Figura 4 – Exemplos de projetos no Scratch. (1) Museu de Arte – Artes. (2) Partes do Sistema Digestivo – Ciências. (3) Monumentos da Itália – História Italiana. (4) Jogo de Futebol – Educação Física. (5) Peso nos Planetas – Física. (6) Sistema Monetário – Matemática

Fonte: autores

A cada aula, foram apresentadas as propostas dos projetos e, a partir daí, os alunos são convidados a buscarem uma solução e desenvolvimento para a ideia.

Os alunos então desenvolveram e completaram o projeto, consultando seus livros e cadernos, a internet e o colega parceiro. Ao final da aula, eles compartilharam o link do projeto em um documento *online* (Google Docs), que foi encaminhado de volta para o professor que propôs o projeto. O professor da disciplina tinha as seguintes opções: a) pedir que os alunos apresentem esse projeto; b) pedir que continuem esse projeto como tarefa de casa; c) avaliar o que foi feito; d) usar o projeto como base para perguntas em avaliações, ou outras iniciativas que achar interessante.

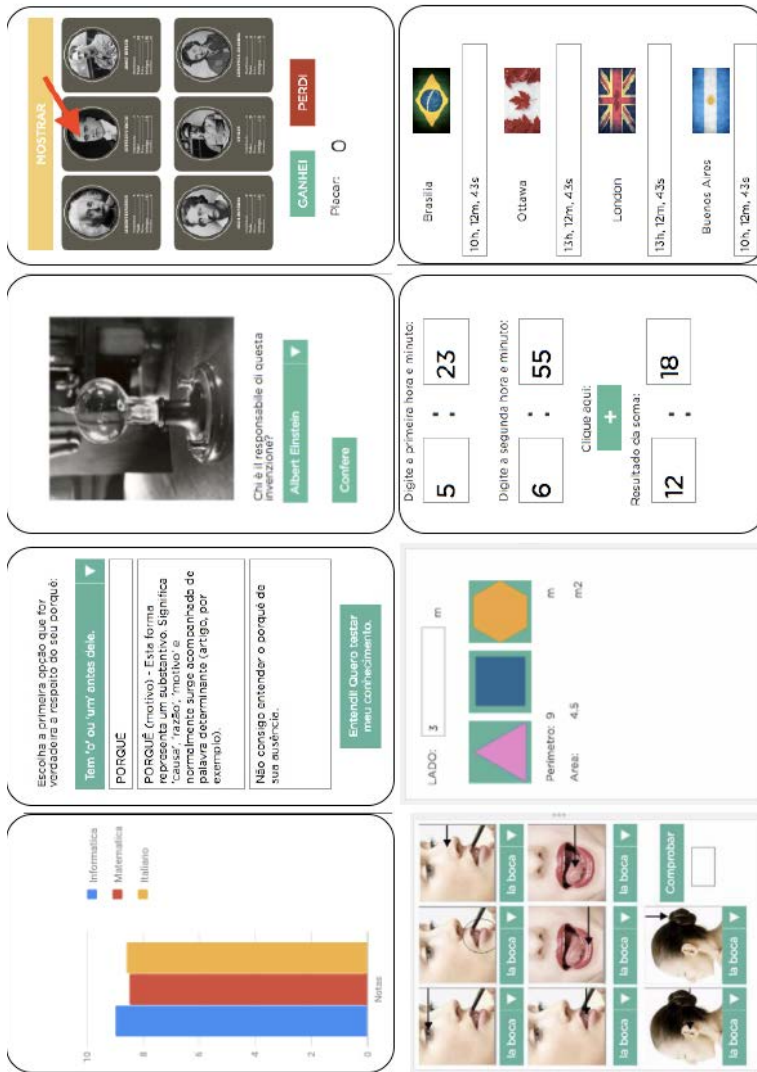


Figura 4 – Exemplos de projetos de Aplicativos propostos pelos professores, com material de apoio, implementados pelos alunos. (1) Organizador de Notas – Matemática. (2) Significado dos Porquês – Língua Portuguesa. (3) Invenções – Organização Industrial. (4) Trunfo Histórico – História. (5) Partes da Cabeça – Língua Portuguesa. (6) Área e Perímetro – Matemática. (7) Calculadora de hora e minuto – Matemática. (8) Fusos Horários – Inglês

Fonte: autores

No total, foram 103 projetos desenvolvidos, sendo 57 projetos em Scratch Jr. e Scratch para o fundamental I, e 46 projetos no Scratch e App Lab para o fundamental II. No fundamental I, foram oito de Geografia, sete de História, sete de História e Geografia Brasileira, seis de Inglês, cinco de Português, cinco de Música, cinco de Língua Italiana, cinco de Matemática, quatro de Artes, quatro de Educação Física e um de Ciências. No fundamental II, foram nove projetos de Matemática, sete de Tecnologia, cinco de Inglês, três de Geografia, três de Música, três de Ciências, três de Educação Física, três de Espanhol, três de História e Geografia Brasileira, dois de Artes, dois de Italiano, dois de Religião e um de Português. Para uma lista completa dos projetos desenvolvidos, visite: www.justcoding.com.br/projetosprontos.

Fase 4: Acompanhamento de Implantação em sala de aula

Durante os três primeiros anos do projeto, um acompanhamento constante foi necessário, de forma a dar maior consistência aos projetos, corrigir caminhos e ajustar os projetos para torná-los mais significativos e mais efetivos no processo de aprendizagem e interdisciplinaridade.

Os tutoriais inicialmente foram desenvolvidos na forma de documentação detalhada e, posteriormente, foram transformados em “Code Cards”, inspirados nos “Code Cards” disponíveis para o Scratch, como o exemplo mostrado na Figura 5.

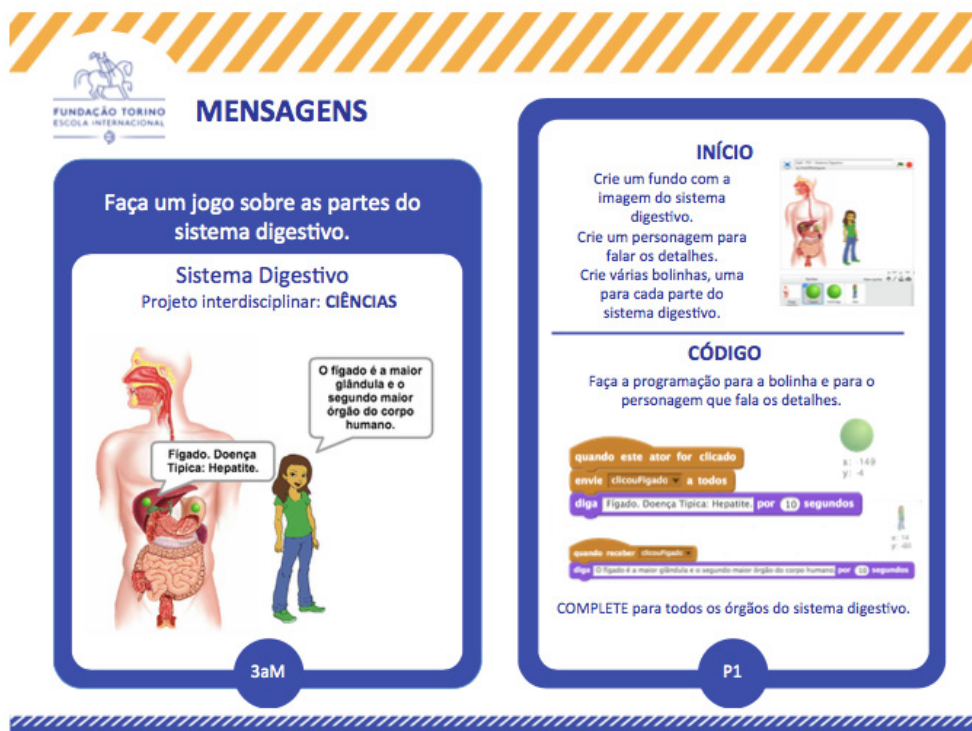


Figura 5 – Exemplo de *Code Card*: Projeto Partes do Sistema Digestivo

Fonte: autores

Esses “*Code Cards*” funcionam como referência no desenvolvimento do projeto, ao mesmo tempo em que permitem que o aluno seja responsável pelo próprio processo de aprendizagem, uma vez que não oferecem as respostas, apenas um caminho para a descoberta das respostas, de acordo com os princípios de aprendizagem defendidos pelo *Creative Learning* e pelo *Coding as a Playground*.

Os “*Code Cards*” são organizados por complexidade, sendo os primeiros relativos a comandos simples, como adicionar um personagem, ou mover um personagem, passando por conceitos mais complexos, como estrutura de repetição, condicional e mensagens, e terminando em eventos e menus para aplicativos.

AVALIAÇÃO DO PROJETO

Avaliação qualitativa do impacto pelos professores

Durante todo o período do projeto, os professores trouxeram *feedback* positivo para o projeto. Apenas um professor sugeriu modificações na forma como o projeto estava sendo implantado, propondo que o professor da disciplina responsável pelo projeto fosse ao laboratório com os alunos durante o desenvolvimento dos projetos.

Ao final do primeiro ano de implantação, os professores foram convidados a responder um questionário *online* que apresentava questões abertas, questões fechadas e um espaço para comentários livres. O preenchimento era voluntário e anônimo, até o final do semestre, e a participação era feita clicando em um *link* enviado por e-mail. Em conversas informais posteriores, professores relataram ter esquecido de preencher, não se lembrar de ter recebido o e-mail, ou não ter tido tempo. Por isso, dos 35 professores envolvidos no projeto, 18 responderam ao questionário (51%).

A Figura 6 mostra as respostas para duas perguntas: (1) Como você considera o nível de engajamento dos alunos no projeto? e (2) Qual a sua avaliação geral do projeto?

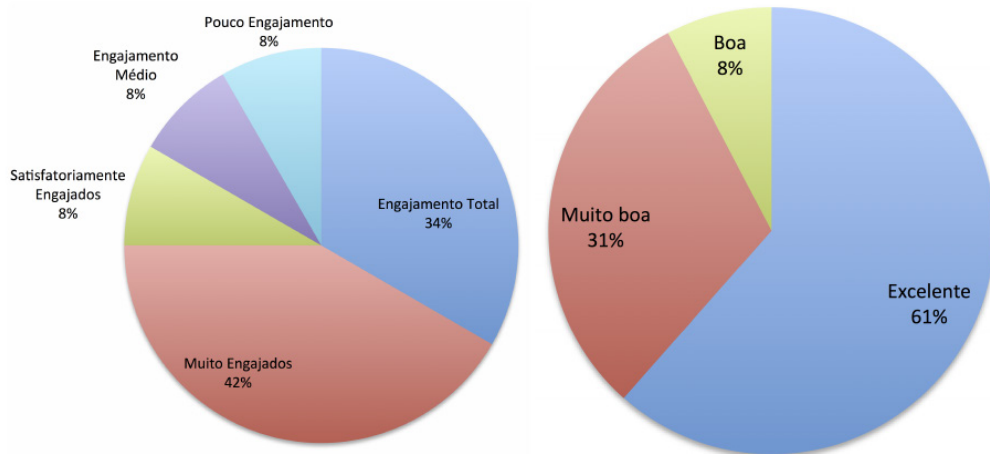


Figura 6 – Avaliação voluntária qualitativa do projeto pelos professores que responderam ao questionário. (1) Como você considera o nível de engajamento dos alunos no projeto? (2) Qual a sua avaliação geral do projeto?

Fonte: autores

A grande maioria dos professores (84%) considerou o engajamento dos alunos total (34%), muito engajados (42%) ou satisfatoriamente engajados (8%). Apenas 16% consideraram engajamento médio (8%) ou pouco engajamento (8%), o que é um resultado excelente, considerando a abrangência do projeto e a grande quantidade de alunos envolvidos.

A totalidade dos professores aprovou o projeto, sendo que 61% consideraram o projeto como excelente, 31% como muito bom e 8% como bom (não houve nenhuma resposta regular ou ruim). Esse resultado mostra que a implantação foi extremamente bem-sucedida, e sua importância e relevância foi aprovada pela comunidade envolvida.

Com relação ao espaço para comentários livres, foram feitos os seguintes comentários e sugestões dos professores:

1. “Talvez uma maior compreensão pelos professores acerca do que está ocorrendo e o que os alunos estão aprendendo.”

2. “Talvez uma apresentação por parte dos alunos, no estilo da escola superior.”
3. “Acho que as Terceiras Médias (8o Ano) deste ano foram muito “fracas”. Talvez, para o próximo ano, com turmas mais preparadas, o mesmo projeto possa dar mais certo.”
4. “Aulas melhores distribuídas, pois quase sempre eram no meus horários(sic). Essa situação prejudicou-me no desenvolvimento da minha disciplina.”
5. “Mais tempo para execução do projeto e mais momentos para que os professores das diferentes áreas os possam se encontrar com os alunos para acompanhar o desenvolvimento dos trabalhos.”
6. “A possibilidade dos alunos usarem tablets em sala de aula.”
7. “Gostei muito da forma como foram conduzidas as aulas.”
8. “Aumento da carga horária, em especial na 1ª Elementare.”

A análise desses comentários livres resultou nas seguintes modificações, realizadas a partir do segundo ano de implantação do projeto:

- Os professores estavam se sentindo desconectados do projeto após a fase inicial de definição. Foi marcada uma reunião para esclarecimento das próximas etapas. Ao final do segundo ano de implantação, foi realizado um Workshop de reciclagem, em que os projetos foram novamente apresentados para os professores e eles foram convidados a contribuir, e participarem. Vinte professores compareceram e desenvolveram novos projetos, além de discutir técnicas de como incluir a programação de forma mais integrada em suas disciplinas.
- Foi criada uma página *online* para registro e exposição dos projetos dos alunos, que é enviada anualmente para os pais.
- No ano de 2017, a professora de Inglês baseou todo o projeto da apresentação da Feira da Cultura em torno de Programação e Coding. Os alunos desenvolveram projetos originais, que foram apresentados para os pais em tablets.

- No ano de 2018, a apresentação de dança da Fundação Torino incluiu um QR code com o link dessa página, e as roupas dos alunos foram temáticas ligadas à programação de computadores.
- Foi feita uma reunião com a coordenação para traçar estratégias para evitar problemas no cronograma com relação aos professores de disciplinas com menos horas semanais.
- Foram comprados tablets e computadores em um suporte móvel, que agora pode ser usado por todos os professores da escola em sala de aula, para continuação dos projetos desenvolvidos ou criação de projetos novos.

Avaliação quantitativa do impacto nas notas

Para avaliar o impacto do projeto nas notas dos alunos, foram analisadas as notas finais dos alunos do 1º ao 8º ano da Fundação Torino, durante três anos: um ano antes e dois anos durante a implantação, resultando em um total de 37.042 notas de mais de 700 alunos.

Os resultados foram compilados utilizando-se a plataforma R, analisando os dados segundo diversas dimensões, como diferentes disciplinas e diferentes séries, e são apresentados a seguir.

Impacto nas notas por disciplina

Na Fundação Torino Escola Internacional, as séries são divididas de forma diferente do que nas demais escolas brasileiras. Do 1º ao 5º ano, é chamado de Elementare. Do 6º ao 8º ano, é chamado de Media. E o 9º Ano é considerado parte de um ciclo seguinte, chamado Superiore.

As Figuras 7 e 8 mostram o efeito na nota média dos alunos por disciplina, separados em educação fundamental I e educação fundamental II. A área em amarelo indica o período de implantação do projeto.

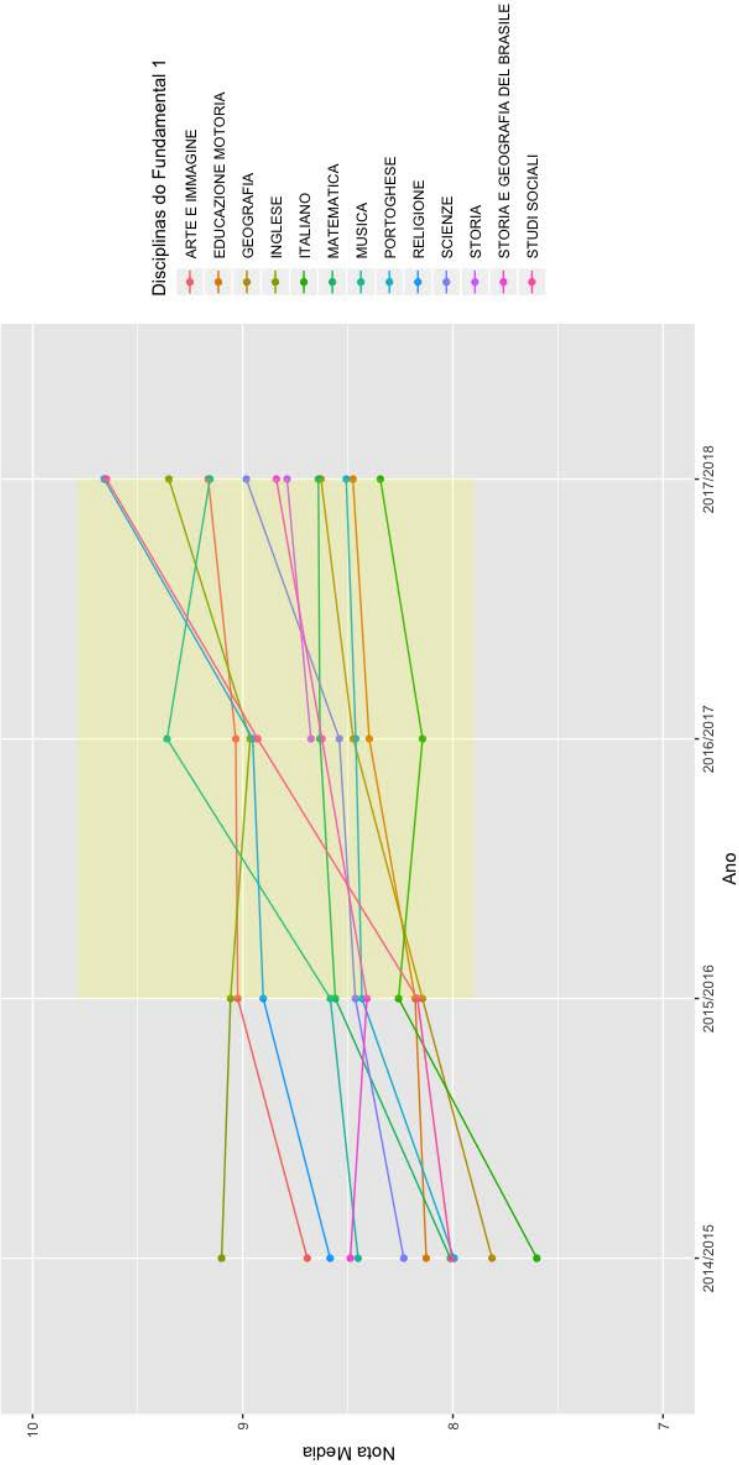


Figura 7 – Nota média dos alunos por disciplina do ensino fundamental I

Fonte: autores

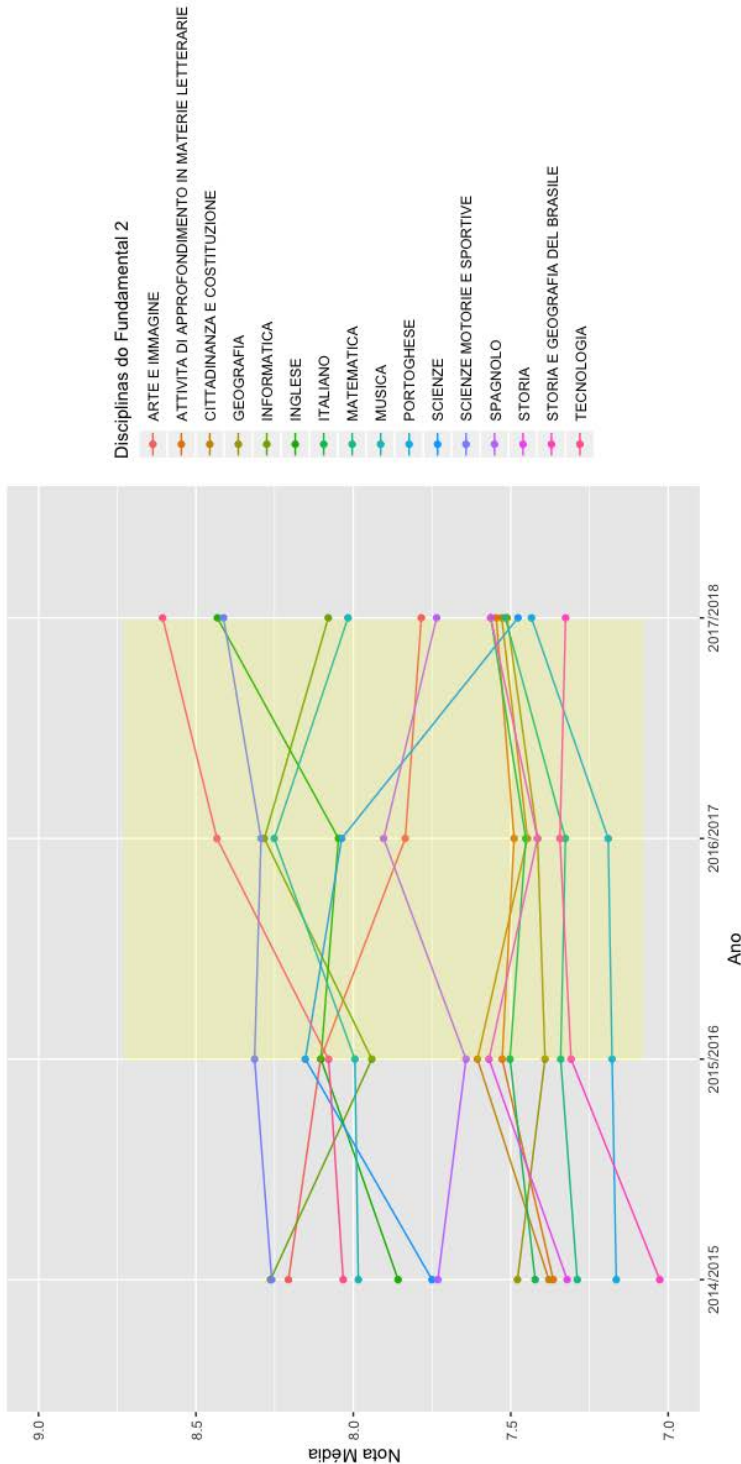


Figura 8 – Nota média dos alunos por disciplina do ensino fundamental II
Fonte: autores

Os resultados indicam que há uma grande diferença no efeito entre as diversas disciplinas, tendo um efeito positivo em algumas, enquanto outras permanecem estacionárias ou mesmo observam queda. Alguns fatores podem explicar esse impacto diferente em cada disciplina, como:

1. Formas de aprendizado diferente, em áreas diferentes, resultam em impactos diferentes em cada disciplina.
2. Alguns professores deixaram a instituição após desenvolverem o projeto, e seus projetos não foram acompanhados. Ao final do segundo ano, estes foram reformulados para atender aos novos professores.
3. Alguns professores incluíram a programação em outras iniciativas, adicionando ao projeto proposto. Professores mais engajados, como de Inglês e Educação Física, resultaram em notas com tendência de aumento.
4. Alguns professores mudaram seus critérios de avaliação durante o período avaliado, tendo um impacto negativo no desenvolvimento geral do projeto.

Por esses motivos, não foi possível estabelecer uma correlação direta entre o número de projetos implementados e a nota da disciplina. Esses dados serão estudados com mais detalhes em agosto de 2019, na conclusão do projeto.

Impacto nas notas por turma

O efeito da implantação com relação às turmas encontra-se apresentado nas figuras 9 e 10, separados em educação fundamental I e fundamental II. A área em amarelo indica o período de implantação do projeto.

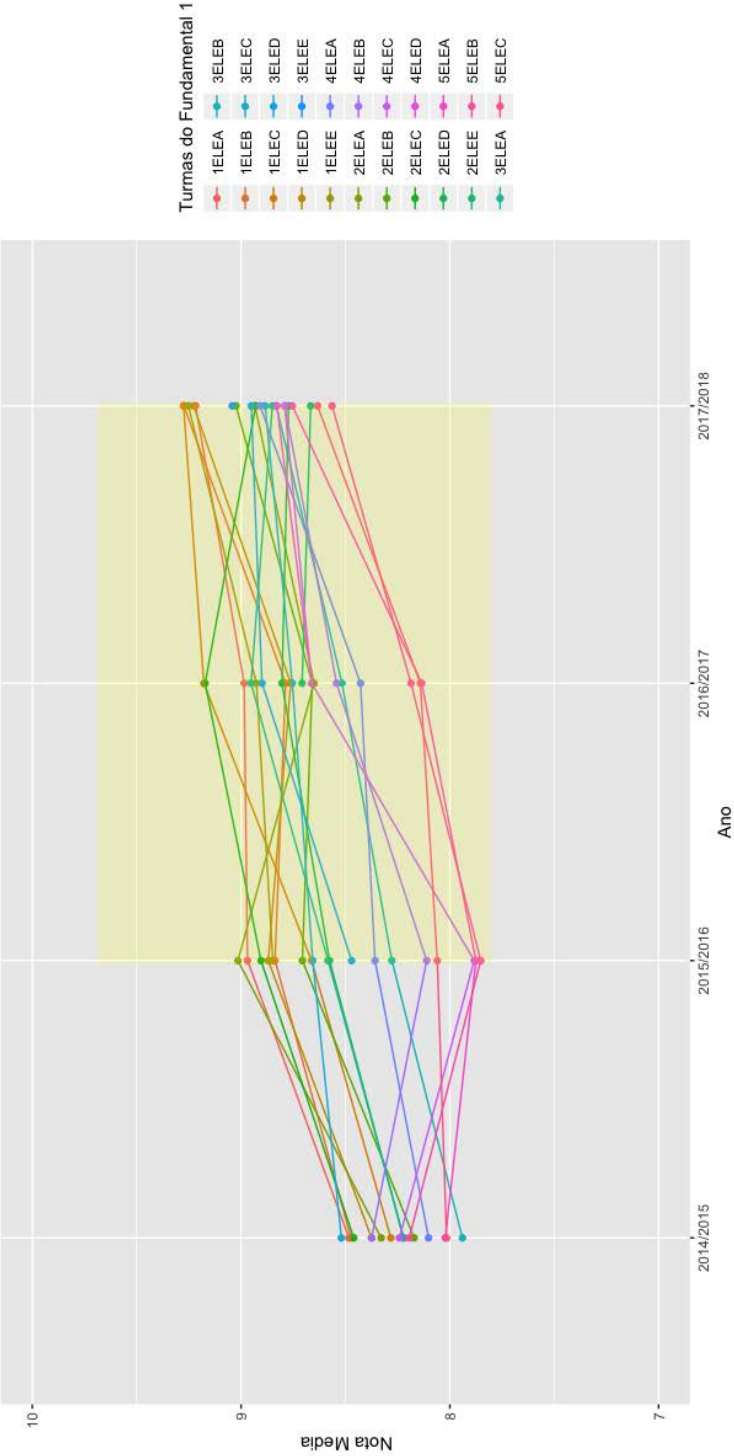


Figura 9 - Nota média dos alunos por ano escolar do ensino fundamental I
Fonte: autores

Esses resultados mostram uma tendência de impacto positivo geral nas notas, quando comparadas ao ano anterior. Observa-se também que esse aumento não está distribuído de forma igual entre todas as séries, sendo que o maior impacto é observado no último ano do fundamental I e no penúltimo ano do fundamental II.

A hipótese que está sendo avaliada para esse efeito maior nas séries finais é de que esses alunos foram apresentados com projetos mais desafiadores do que as séries iniciais. Além disso, geralmente as notas nas séries finais tendem a ser mais baixas do que nas séries iniciais, oferecendo uma maior oportunidade de melhoria. Esses efeitos continuarão sendo observados até o final do projeto, previsto para junho de 2019.

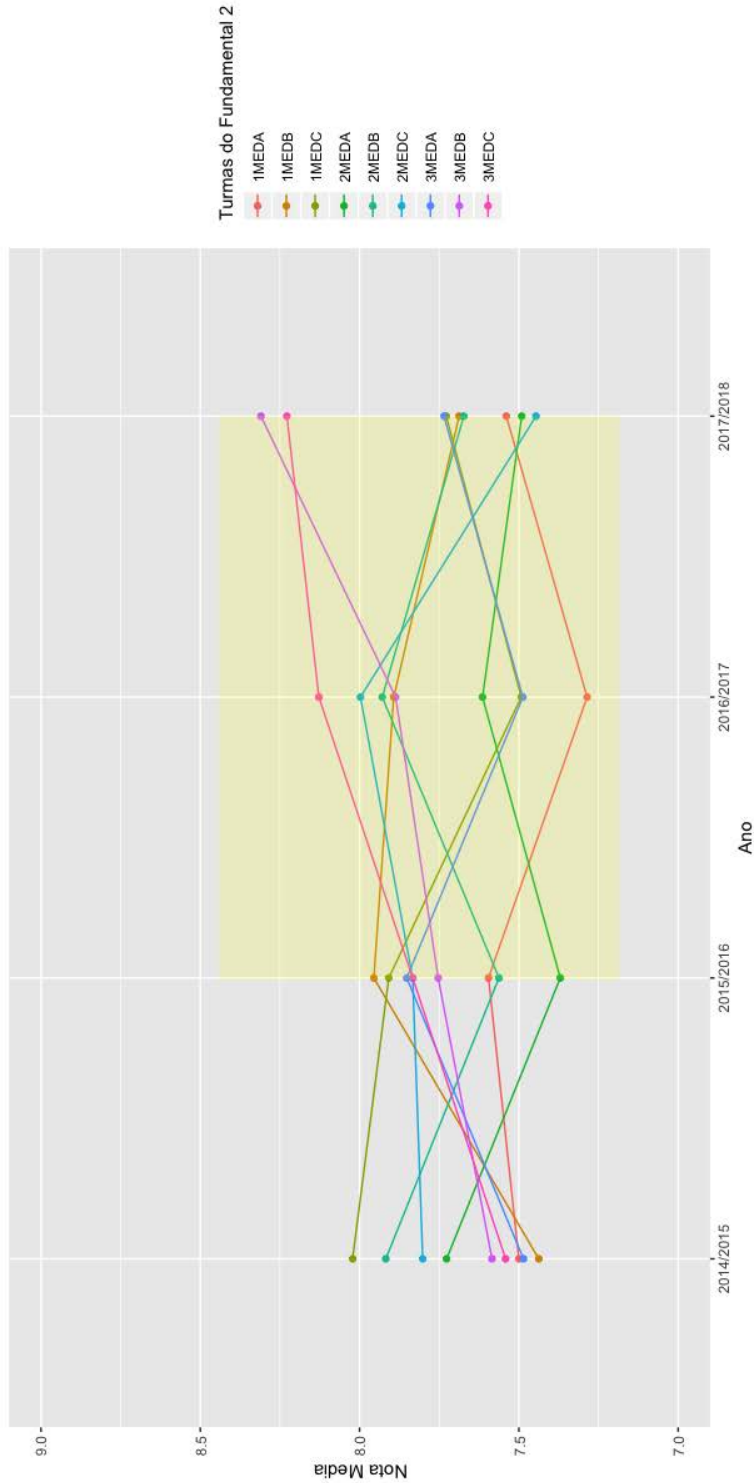


Figura 10 Nota média dos alunos por ano escolar do ensino fundamental II

Fonte: autores

CONCLUSÃO

Este artigo apresentou o processo de implantação e o impacto de um projeto de três anos para integrar programação em todas as séries e disciplinas do ensino fundamental I e II da Fundação Torino.

A implantação resultou no desenvolvimento de mais de 100 projetos, envolveu mais de 20 professores e impactou positivamente as notas de mais de 700 alunos em um período de três anos. Foi criado material didático na forma de tutoriais para os professores e “Code Cards” para os alunos, de forma que todo o processo de aprendizado foi baseado na exploração pelo aluno do conhecimento de forma autônoma e engajada, seguindo os princípios do *Creative Learning*.

No geral, a avaliação do projeto pelos professores foi positiva, sendo que os professores foram envolvidos em todas as fases do projeto, começando com o treinamento destes em pensamento computacional, durante a escolha dos temas a serem desenvolvidos e na avaliação do projeto de forma anônima.

Foi observado que o sistema de “Code Cards” foi muito mais eficiente do que tutoriais detalhados, que prendiam o aluno a um modelo único de projeto. Os resultados finais foram mais heterogêneos e o engajamento e nível de aprendizado obtidos eram muito superiores em um modelo no qual o aluno tinha liberdade de escolha.

Ao mesmo tempo, o engajamento de alguns professores de disciplinas específicas, como o Inglês e Educação Física, tornou o projeto muito mais rico em termos de experiências para os alunos, como relata Gabriella Fiuza, professora de Educação Física:

“Quando eu estou empolgada com o projeto, eu passo isso para as crianças, e isso faz toda a diferença.”

Outro aprendizado de grande valia foi o modelo de duplas no uso dos computadores, que criou um ambiente de cooperação, trabalho em grupo, colaboração para solução de problemas e comemoração das vitórias alcançadas.

Estratégias para maior engajamento dos professores incluíram a formação continuada, a reciclagem após um ano de projeto, e a

disponibilização de um e-mail para envio de sugestões e ideias de projetos. Considerando-se que a participação no projeto não era obrigatória, concluímos que houve um grande engajamento.

Por fim, foi observada uma tendência de impacto positiva nas notas dos alunos, especialmente nas séries finais do fundamental I e do fundamental II. O projeto está em constante evolução, e os impactos serão novamente avaliados ao final da implantação, prevista para o segundo semestre de 2019.

Conclui-se que o projeto foi desenvolvido com sucesso, podendo ser adotado como referência na implantação curricular de programação de computadores e de pensamento computacional em escolas, tanto na rede particular quanto na rede pública, por não precisar de compra de material específico e nem resultar em custos recorrentes. O principal fator para o sucesso da implantação é a formação adequada de todos os profissionais envolvidos, assim como engajar a participação dos professores da instituição.

REFERÊNCIAS

Bers, M. U. (2017). *Coding as a Playground*. Routledge, 1st ed.

BNCC(2018). *Base Nacional Comum Curricular(BNCC)*. Versão final homologada da Educação Infantil ao Ensino Fundamental em 20/12/2017. Ministério da Educação. Recuperado em nov. 2018, de <http://basenacionalcomum.mec.gov.br>.

Bruce, C. S. (1997). *Seven faces of information literacy in higher education*. Brisbane: Queensland University of Technology.

Cieb - Centro de Inovação para a Educação Brasileira (2018). *Currículo de Referência em Tecnologia e Computação*. Recuperado em nov. 2018, de <http://curriculo.cieb.net.br>.

Code.org (2013). *Code.org*. Recuperado em nov. 2018, de <http://code.org>.

Code.org (2018). *Hour of code*. Recuperado em nov. 2018, de <https://hourofcode.com/pt/pt>.

Geekie (2016) *Escola terá mais tecnologia e menos disciplinas, defende curadora da Bett*. Recuperado em jul. 2019, de <https://www.geekie.com.br/blog/escola-tecnologia-disciplinas-bett/>.

MIT - *Lifelong Kindergarten Lab* (2007). *Scratch*. Recuperado em nov. 2018, de <http://scratch.mit.edu>.

MIUR - Ministero dell'istruzione, dell'università e della ricerca (2014). *Programma il futuro*. Recuperado em 2018, de <https://pt.slideshare.net/miursocial/programma-il-futuro>.

Fórum Econômico Mundial (2016). *Digital skills we must teach our children*. Recuperado em nov. 2018, de <https://www.weforum.org/agenda/2016/06/8-digital-skills-we-must-teach-our-children/>.

Fundação Telefônica Vivo e Fundação Lemann (2014). *Um Guia para Construção do Pensamento Computacional*. Fundação Telefônica Vivo, Fundação Lemann. Livro eletrônico, recuperado em nov. 2018, de <http://fundacaotelefonica.org.br/acervo/um-guia-para-construcao-do-pensamento-computacional>.

Paiva, A. M. (2010). Letramento digital como prerrogativa social. *II Congresso Internacional Tic e Educação*. Lisboa. Portugal. Recuperado em nov. 2018, de <http://ticeduca.ie.ul.pt/atas/pdf/98.pdf>.

Papert, S. (1980). *Mindstorms: Children, Computers and Powerful Ideas*. Basic Books. 1st ed.

Resnick, M., & Robinson, K. (2017). *Lifelong Kindergarten - Cultivating Creativity through Projects, Passion, Peers, and Play*. MIT Press, 1st ed.

Rogers, C. (1975). *Teoria da Personalidade: Aprendizagem centrada no aluno*. Editora Santo Antônio.

SBC. Referenciais de Formação para Computação na Educação Básica. Recuperado em jan. 2019, de <http://sbc.org.br/files/ComputacaoEducacaoBasica-versaofinal-julho2017.pdf> 2017.

Unesco (2006). *HIGH-LEVEL Colloquium on Information Literacy and Lifelong Learning Final Report*. Alexandria: Unesco/NFIL/IFLA. Recuperado em nov. 2018, de <http://www.ifla.org/III/wsis/High-Level-Colloquium.pdf>.

Unesco (2007). *Understanding Information Literacy: A Primer Information for All Programme*. Paris: Information Society Division, Communication and Information - Unesco.

White House USA (2016). *Computer science for all*. Recuperado em nov. 2018, de <https://www.whitehouse.gov/%E2%80%A6/2016/01/30/computer-science-all>.

CAPÍTULO 7

PENSAMENTO COMPUTACIONAL E PROGRAMAÇÃO: IMPACTOS NA FORMAÇÃO DE PROFESSORES E CONTRIBUIÇÕES PARA PRÁTICAS PEDAGÓGICAS INTERDISCIPLINARES

Eliei Constantino da Silva

Maria Teresa Zampieri

Sueli Liberatti Javaroni

INTRODUÇÃO

Neste artigo, apresentamos e discutimos um espaço de formação em serviço, constituído em uma escola pública pertencente ao Programa Ensino Integral (PEI), localizada no município de Rio Claro/SP. Esse espaço de formação ocorreu por meio de encontros quinzenais na própria escola, nos horários de Aula de Trabalho Pedagógico Coletivo (ATPC). Os participantes dessa formação são professores que atuam nos anos finais do ensino fundamental e atuam nas áreas de Ciências Humanas; Linguagens e Códigos e Ciências da Natureza e Matemática, e os proponentes desse espaço de formação foram: a coordenadora do projeto de extensão (terceira autora deste capítulo), dois discentes de mestrado (sendo um deles o primeiro autor), uma pesquisadora de pós-doutorado (segunda autora) e duas discentes de Iniciação Científica da Unesp (campus Rio Claro e Bauru).

A ideologia subjacente a essa formação levou em consideração que os saberes profissionais dos professores são construídos e transformados ao longo da temporalidade de suas carreiras, e são indissociáveis de suas vivências, sejam estes pessoais ou profissionais. Esse argumento é defendido por Tardif (2000), e está em sinergia com a formação desenvolvida, cujo objetivo é construir um espaço colaborativo com os professores das diversas áreas, de modo que possam produzir conhecimento sobre a integração do Pensamento Computacional em suas abordagens pedagógicas, por meio do uso de softwares de programação e uso de robótica nos processos de ensino e aprendizagem, além de promover reflexões acerca da interdisciplinaridade em suas práticas pedagógicas. Foi realizado ainda o acompanhamento desses professores na elaboração e aplicação de atividades em sala de aula de modo que seus

estudantes pudessem vivenciar experiências que envolvessem tais recursos e o Pensamento Computacional.

Esse espaço de formação está vinculado ao Projeto de Extensão “Pensamento Computacional e a Interdisciplinaridade em sala de aula”, aprovado sob o Edital n.º 2/2018 da Pró-reitoria de Extensão Universitária da Universidade Estadual Paulista (Unesp), e a sua constituição e realização na escola mencionada se deu pela parceria com a escola para o desenvolvimento de duas pesquisas de Mestrado: a primeira (Silva, 2018) teve o objetivo de analisar o processo de formação de conceitos matemáticos dos estudantes do nono ano do ensino fundamental ao desenvolverem o Pensamento Computacional por meio da realização de atividades com robótica (Silva & Javaroni, 2018a, 2018b), e a segunda, iniciada em 2018, tem o objetivo de investigar a relação entre a Modelagem Matemática e o Pensamento Computacional com estudantes do nono ano do ensino fundamental, ao realizarem atividades com kits de robótica (Souza, 2018).

A parceria com a escola, desde o desenvolvimento da primeira pesquisa de Mestrado mencionada, estreitou-se na medida em que os estudantes foram demonstrando à gestão escolar o seu desenvolvimento e melhoria no rendimento escolar na disciplina de Matemática e em outras disciplinas regulares (aquelas que constam na matriz curricular dos anos finais do ensino fundamental, que são obrigatórias para o estudante) ou eletivas (aquelas que constam na matriz curricular com temáticas diferentes das disciplinas regulares, para opção do estudante) da escola, indicando as contribuições do desenvolvimento do Pensamento Computacional para as demais componentes curriculares (Silva, 2018).

Em consequência dessas pesquisas e refletindo sobre os dados de uma terceira pesquisa de Mestrado, que está sendo desenvolvida desde 2017, nessa referida escola e em outras duas escolas pertencentes ao Programa Ensino Integral localizadas nesse mesmo município – cujo objetivo é compreender as concepções dos professores acerca do uso das tecnologias dentro desse contexto (Silva & Javaroni, 2017) –, revelou-se então a necessidade de formação para que os professores de todas as disciplinas que compõem a grade escolar pudessem utilizar recursos que a escola possui, como kits de

robótica e software de programação instalados nos computadores e *netbooks*, em suas práticas pedagógicas.

Assim, firmamos essa parceria com a escola no intuito de dar essa devolutiva de nossas pesquisas, de modo a contribuir para a formação de recursos humanos para o trabalho pedagógico com robótica, linguagem de programação e demais recursos educacionais a fim de propiciar o desenvolvimento do Pensamento Computacional e contribuir para o processo de ensino e aprendizagem da disciplina ministrada por eles, em uma perspectiva interdisciplinar que, em particular, nas escolas da Educação Básica pertencentes ao PEI, têm esse aspecto privilegiado por conta de sua proposta de trabalho e grade curricular (São Paulo, 2017).

Ou seja, no PEI, além de suas atividades cotidianas do magistério, os professores têm responsabilidade

[...] na orientação aos alunos em seu desenvolvimento pessoal, acadêmico e profissional. Com a dedicação integral à unidade escolar, dentro e fora da sala de aula, espera-se do professor iniciativas que operacionalizem seu apoio social, material e simbólico à elaboração e realização do projeto pessoal e profissional do aluno, ações que o ajudem a superar suas dificuldades e atividades que o energizem para buscar o caminho de seus ideais. (São Paulo, 2012, p. 12).

Assim, as escolas pertencentes ao PEI apresentam características que favorecem o trabalho interdisciplinar, permitindo a criação de

novos conhecimentos que se agregam a cada uma das disciplinas ou se situam na zona de interseção entre elas, partindo das interações dos sujeitos no ambiente e de elementos de uma prática comunicativa que eles desenvolvem (Tomaz & David, 2007, p. 27).

Esse argumento está em sinergia com o que preconiza a Base Nacional Comum Curricular (BNCC) para as instituições escolares, ou seja, o documento sugere que eles tenham a autonomia para “decidir sobre formas de organização interdisciplinar dos componentes curriculares [...] para adotar estratégias mais dinâmicas, interativas e colaborativas em relação à gestão do ensino e da aprendizagem” (Brasil, 2018).

Desse modo, durante o desenvolvimento dessa formação, nossos olhares de pesquisadores visaram a focalizar aspectos subjetivos acerca de como os professores desenvolvem o Pensamento Computacional ao longo da realização das atividades propostas e como estes promovem o seu desenvolvimento em suas práticas pedagógicas em sala de aula com seus estudantes. Assim, a abordagem metodológica adotada neste projeto de extensão é a qualitativa, pois ela é

focalizada no indivíduo, com toda a sua complexidade, e na sua inserção e interação com o ambiente sociocultural e natural. O referencial teórico, que resulta de uma filosofia do pesquisador, é intrínseco ao processo (D'Ambrosio, 2008, p. 103).

Como parte de nossos procedimentos metodológicos, adotamos o diário de campo digital, videogravação dos encontros e a observação participante. Ou seja, cada um dos pesquisadores que compuseram a formação elaborou, posteriormente, a cada encontro, um áudio, narrando detalhadamente e criticamente todos os acontecimentos que observou. Isso possibilitou que as particularidades e nuances de cada discussão, as quais são intrínsecas ao desenvolvimento de cada atividade, fossem resgatadas, debatidas e analisadas sob a luz de um referencial teórico que está se constituindo, sendo algumas dessas obras literárias principais apresentadas aqui.

Ademais, foram utilizados os diversos materiais didáticos presentes na escola para o estudo sobre a integração do Pensamento Computacional em abordagens pedagógicas. Um objetivo específico desse espaço de formação foi desenvolver com os professores

possibilidades de uso de linguagem de programação no processo de ensino e aprendizagem de suas áreas específicas, de maneira que tais kits pudessem ser utilizados no ensino de Língua Portuguesa, Artes, Geografia, Ciências etc. Isso foi feito com o intuito de que as diferentes áreas do conhecimento se unissem promovendo a interdisciplinaridade e uma formação mais completa ao estudante, propiciando a aquisição de conhecimentos de diferentes áreas ao se engajar em atividades que fomentassem o desenvolvimento dessas temáticas mencionadas.

A dinâmica adotada, portanto, incluiu uma ambientação com a linguagem de programação presente no software *Scratch*, de modo que eles pudessem conhecer um ambiente de programação ao mesmo tempo em que pudessem, na prática, perceber o Pensamento Computacional sendo desenvolvido, reflexões sobre o uso de ambientes de programação em suas áreas específicas de acordo com os conteúdos constantes no currículo do estado de São Paulo, de maneira que essa reflexão ocorresse com base na realidade vivenciada por eles em suas práticas pedagógicas.

Além dessa ambientação, fez parte da dinâmica desse espaço de formação o estudo, discussão e reflexão de artigos e trabalhos sobre a temática, de modo que os professores pudessem conhecer iniciativas que acontecem em outras realidades escolares, aprofundando-se também teoricamente.

Desse modo, a dinâmica de trabalho e os procedimentos metodológicos adotados tiveram o objetivo de centralizar as ações na formação completa do professor para que ele tivesse condições de integrar programação computacional e outros recursos educacionais presentes na escola, em suas práticas pedagógicas, a partir de estudos teóricos e reflexão sobre essa integração, cujos referenciais serão discutidos a seguir.

PENSAMENTO COMPUTACIONAL

Entendemos por Pensamento Computacional o conjunto de processos mentais, como a decomposição do problema, reconhecimento de padrões, pensamento algorítmico e abstração, que provoca o surgimento de novas ideias à medida que conexões entre as soluções são realizadas (Silva, 2018). Trata-se de processos mentais envolvidos na formulação de problemas e suas soluções para que estas sejam representadas de uma forma que possa ser realizada por um agente de processamento de informações (Wing, 2008). Portanto, o Pensamento Computacional está relacionado com o raciocínio.

Ele é desenvolvido pelo humano e acontece quando este reorganiza o seu pensamento incorporando aos seus processos mentais as ferramentas abstratas utilizadas para construir os equipamentos computacionais e desenvolver os grandes sistemas de software. Portanto, é a união do pensamento humano com as capacidades computacionais, ou seja, com a habilidade de pensar algorítmicamente (Wing, 2016; Philips, 2009)

Esse Pensamento é analítico, pois compartilha com o raciocínio matemático as formas gerais para solucionar um problema, tendo como essência a abstração (Wing, 2006), permitindo a conexão entre várias áreas; é o conceituar, pensar de maneira abstrata, ao invés de simplesmente utilizar técnicas de programação para solucionar um problema; é a ideia, o conceito computacional que utilizamos para resolver problemas, gerenciar nossas vidas, comunicar e interagir com outras pessoas (Wing, 2006).

Ao desenvolvê-lo, estamos desenvolvendo o pensamento abstrato para perceber os problemas e solucioná-los, o pensamento algorítmico para encontrar a forma mais eficaz e eficiente de resolver um problema, o pensamento lógico para a formulação e verificação de conjecturas, e o pensamento dimensionável para decompor problemas grandes em pequenas partes, visando a formular uma solução mais complexa (Sousa & Lencastre, 2014).

Esses pensamentos permitem “reconhecer padrões e desenvolver métodos de representação e generalização [que] são ações fundamentais do fazer e aprender” (Moraes, Basso, & Fagundes, 2017,

p. 470). Assim, o Pensamento Computacional pode ser desenvolvido nos processos de ensino e aprendizagem de todas as áreas do conhecimento e possibilita o fazer e o aprender.

Isso permite que o estudante seja o protagonista de todo o processo de aprendizagem, que passa a ser mediado pelo professor. Desse modo, as ações colaborativas desse espaço de formação se pautaram no estudo, compreensão e aplicação do Pensamento Computacional em diferentes campos, nos processos de ensino e aprendizagem, a partir de diferentes enfoques pedagógicos.

Diante disso, considerando a importância do trabalho que ocorre nessa escola do PEI e as possibilidades que o desenvolvimento do Pensamento Computacional pode propiciar aos processos de ensino e aprendizagem no ambiente escolar, apresentamos e discutimos as ações que temos desenvolvido nessa escola de tempo integral.

O ESPAÇO DE FORMAÇÃO

A dinâmica do espaço de formação foi flexível para que os professores pudessem expor suas opiniões e sugerir ideias para serem abordadas nesse espaço colaborativo, possibilitando a todos os participantes (tanto os proponentes quanto os professores) a oportunidade de transformarem seus saberes prévios e construir novos, do modo como sugere Tardif (2000).

No primeiro encontro, mostramos o protótipo de um carrinho e de um semáforo, desenvolvidos por estudantes da escola em anos anteriores na disciplina “Práticas de Matemática” com o kit de robótica Arduino Uno, no desenvolvimento das pesquisas de Mestrado mencionadas na introdução deste estudo. Os professores ficaram surpresos e animados com as criações dos estudantes e começaram a conjecturar o que eles poderiam criar e como utilizar esses recursos, pensando nos conteúdos específicos de suas disciplinas.

Isso nos despertou a atenção porque mostrou que, por uma falta de comunicação, ou até mesmo pela carga horária extensa dos pro-

fessores, ou ainda, por motivos diversos, ocorre um desconhecimento acerca dos recursos disponíveis na escola, bem como dos potenciais de tais recursos para o processo de ensino e aprendizagem. E mais, que há um desconhecimento acerca dos projetos desenvolvidos entre os professores, e dos saberes produzidos pelos alunos a partir da realização das atividades inerentes a esses projetos.

Dando continuidade ao nosso relato acerca desse primeiro encontro, esclarecemos ainda com qual linguagem os estudantes desenvolveram a programação dos protótipos criados, apresentando, portanto, aos professores, a linguagem referente ao software *Scratch*. Os estudantes programaram tais protótipos com o software *Scratch for Arduino*, que possui a mesma linguagem de programação do *Scratch*. Por esse motivo, optamos por realizar uma ambientação com essa linguagem de programação, primeiramente.

Assim, sugerimos aos professores que explorassem a interface do *Scratch* (versão on-line), e analisassem alguns projetos desenvolvidos pelos alunos da escola. Foram selecionados seis projetos para desenvolvermos com os professores, que seriam divididos em três grandes áreas: Linguagens e Códigos; Ciências Humanas; Matemática e Ciências da Natureza.

De modo geral, esses projetos foram compostos por: jogos como ping-pong, *quiz* de perguntas e respostas, histórias, animações etc. Além disso, cada um deles envolveu programação específica, sendo algumas mais complexas (incluindo a definição de variáveis, utilizando a funcionalidade de mesmo nome no *Scratch*, articulação de som e imagens, utilização de conceitos matemáticos, como posição no plano cartesiano, ângulos, divisão, funções etc.), e algumas mais simples (utilizando comandos de deslocamento mais básicos como “mova x passos” etc.).

Após esse momento inicial, solicitamos a eles que utilizassem a funcionalidade “ver interior”, presente nesse software, para entenderem a programação referente ao que estavam visualizando. Ao fazerem isso, os professores tiveram noção da linguagem e da complexidade da programação, e também já fizeram analogias com as suas respectivas áreas.

No segundo encontro, selecionamos alguns projetos da rede mundial do *Scratch* e dividimos os professores por áreas (Ciências Humanas, Linguagens e Códigos, Ciências da Natureza e Matemática), para que refletissem acerca das especificidades conceituais dentro de tais áreas, pensando em como aproveitar as potencialidades do software para o desenvolvimento de conteúdos de suas respectivas disciplinas. Alguns deles não se interessaram no começo, enquanto outros professores já começaram a mexer no interior e mudar/criar novas programações das atividades que foram propostas.

Nesse segundo encontro, houve acontecimentos que nos chamaram a atenção, por exemplo: no grupo de Ciências Humanas, uma das duplas começou a explorar a programação de um *quiz*, e decidiu trocar os atores do *Scratch*. Após algumas tentativas sem sucesso, nos requisitaram, e esclarecemos sobre a necessidade de copiar primeiro a programação, antes de apagar o ator, e apenas depois disso, mencionamos que é possível inserir um novo ator, e então colar a programação. Outras situações semelhantes a essa ocorreram, como casos em que os professores queriam inserir sons à programação, modificar cenários, exportar imagens da internet etc.

De modo geral, percebemos nesse encontro que os professores estavam focalizando essencialmente o software, ou seja, estavam buscando compreender suas funcionalidades, limitações, entre outras coisas, sem adentrar em ideias de abordagens para serem desenvolvidas em sala de aula, tampouco em aspectos teóricos subjacentes ao pensar com esse software. Contudo isso nos pareceu natural, porque, de fato, a maioria deles ainda não tinha interagido com esse software, então, essa ambientação foi necessária e, de certo modo, contribuiu significativamente para as discussões que permearam o terceiro e quarto encontros, em que fizemos um estudo dirigido sobre o texto *Estimulando o Pensamento Computacional com o projeto Logicando*, de Silva, Miroelli e Kologeski (2018).

No terceiro encontro, debatemos o texto em questão, o qual apresentou oficinas lúdicas, que foram abordadas com alunos de 8º e 9º anos do ensino fundamental, cuja finalidade foi o desenvolvimento do Pensamento Computacional por meio do uso de distintas tecnologias. As oficinas foram divididas em três, sendo: Raciocínio Lógico e Lógica de Programação; Prática de Lógica de Programa-

ção e Lógica de Programação com Jogos. Para que o encontro não ficasse cansativo, os professores se dividiram em grupos (por áreas), e, assim, cada um ficaria responsável pelo estudo acerca de uma oficina, especificamente.

Para o desenvolvimento do debate, pensamos em três perguntas diretrizes:

- De acordo com a análise das oficinas, como podemos definir Pensamento Computacional?
- Como o desenvolvimento do Pensamento Computacional, por meio de atividades como as que foram por eles analisadas nos encontros posteriores, poderia contribuir para abordar conteúdos específicos de suas áreas?
- Como os professores trabalhariam o pensamento computacional para promover a interdisciplinaridade?

Durante o debate com os professores, surgiram algumas ideias sobre o que seria o Pensamento Computacional, sintetizadas em argumentos defendidos por eles, tais como: “aprender a aprender o computador”, “pensar como o computador pensa”, “se refere a dar e seguir instruções por meio de comandos e códigos”, “entender a linguagem do computador”, “saber organizar o pensamento como o computador organiza”, “está nos pensamentos do cotidiano”, “tomadas de decisões estratégicas para resolver um problema”. E, por fim, um dos pesquisadores proponentes do curso lançou uma nova questão disparadora, a qual norteou as discussões do quarto encontro. Esta foi a questão: qual a diferença entre o Pensamento computacional e o pensamento “normal”?

Assim, no quarto encontro, cada grupo sintetizou suas concepções acerca do que seria Pensamento Computacional, tendo como base a leitura do texto e a ambientação desenvolvida com o *Scratch*. O grupo das Ciências Humanas argumentou que o Pensamento Computacional é análogo ao funcionamento do computador, ou seja, corresponde a um tipo de pensamento lógico. Já o grupo de Linguagens e Códigos argumentou que se refere ao pensar por meio de linguagens, com a lógica semelhante aos comandos dados em programação, de forma geral, como o uso de “se...então”, “se...

então...senão". E para reforçar esse argumento, uma das professoras desse grupo mencionou um exemplo envolvendo as rotas para ir de sua casa ao consultório de seu dentista, reforçando as similaridades entre a sintaxe presente nessa situação com a das linguagens de programação.

Outro tema pertinente que se destacou, sendo discutido pelos três grupos, foi em relação ao sentimento. Ou seja, os professores debateram sobre a possível relação entre o sentimento e o Pensamento Computacional. Inicialmente, questionaram se o primeiro fazia parte do segundo, depois reiteraram que é exatamente o sentimento que difere o modo de operar dos computadores com o modo humano de operar, e, por fim, refletiram sobre até que ponto o sentimento interfere no Pensamento Computacional.

As discussões foram calorosas e significativas em termos de compartilhamento de ideias e exposição de argumentos, sendo estes embasados teoricamente pelo texto que estava em pauta e nas vivências dos professores. Afirmamos isso por causa do envolvimento deles com as atividades propostas, que foi se intensificando a cada encontro, inclusive até mesmo de alguns professores que não queriam participar dessa formação *a priori*, que na apresentação da proposta da criação desse espaço de formação se mostraram resistentes em participar, fato que consideramos ter sido uma resistência natural ao desconhecido, ou mesmo falta de interesse pessoal.

No quinto encontro, construímos com os professores diversas ideias de atividades que poderíamos criar e desenvolver com os estudantes em uma perspectiva interdisciplinar de modo que os estudantes pudessem desenvolver o Pensamento Computacional no processo de aprendizagem dos temas que fossem abordados. Os professores centraram-se no uso do *Scratch* e se uniram em grupos para começar a planejar aulas para serem desenvolvidas.

Assim, no sexto encontro, os professores começaram a executar os projetos que foram pensados colaborativamente no encontro anterior. Uma professora de Geografia se juntou com um professor de Matemática para trabalharem com bacias hidrográficas no Brasil. As professoras de Inglês, Arte e História se uniram para trabalhar historicamente e geograficamente pontos turísticos conhecidos internacionalmente, e as respectivas questões artísticas referentes

a essas obras, também utilizando o *Scratch*. Duas professoras de Ciências se uniram com o professor de Educação Física e dois professores de Matemática para trabalharem com a pirâmide alimentar e a quantidade de calorias ingeridas pelo ser humano ao consumir determinados alimentos.

Antes de iniciarem a programação no *Scratch*, os professores traçaram uma estratégia para conseguirem alcançar o objetivo final da atividade a ser desenvolvida com os estudantes. Para isso, fizeram várias anotações em seus cadernos, de modo a planejar cada ação e etapa do processo de construção da atividade e da programação. Nesse momento, os professores decompuseram as atividades de modo que conseguissem pensar e encontrar a solução de cada etapa para depois unirem em uma única solução para, então, realizarem a programação. Ou seja, o Pensamento Computacional começou a ser desenvolvido no momento de planejamento das ações de elaboração das atividades (Sousa & Lencastre, 2014; Wing, 2006).

Dando sequência a essas atividades, no sétimo encontro, os professores continuaram trabalhando nos projetos mencionados e iniciaram a programação. Eles decidiram programar para sentirem quais seriam as possíveis dificuldades dos estudantes e para que eles aprendessem programação. Além disso, era necessário esse momento para que eles pudessem perceber e sentir o Pensamento Computacional sendo desenvolvido nas ações da atividade, de modo que pudessem elaborar ações pedagógicas que propiciassem o seu desenvolvimento pelos estudantes no processo de aprendizagem dos conteúdos envolvidos nas atividades.

No encontro posterior, perguntamos como eles estavam visualizando o Pensamento Computacional sendo desenvolvido nesses projetos e como eles estavam avaliando a possibilidade de aplicá-los em sala de aula, quais as potencialidades que poderiam ser desenvolvidas e quais as dificuldades que eles poderiam enfrentar. A maioria dos professores relatou que uma das dificuldades estava centrada no aprender a programar, que estava sendo algo novo para eles, e nesse processo teriam ainda que pensar em desenvolver a programação com os estudantes, e consequentemente tirar as suas dúvidas.

Essa falta de domínio referente à programação estava deixando-os aflitos, pois se questionavam sobre como agir se os estudantes fizessem perguntas que eles não soubessem responder, ou ainda, como mediar uma discussão de forma que fosse criada uma investigação com todos da turma para obter a solução do problema, visto que eles não estavam se sentindo seguros em relação à programação. Mas salientaram que o espaço de formação já estava modificando esse cenário, porém precisariam de mais tempo para obter essa fluência em programação, de modo a se sentirem suficientemente seguros para incorporá-la em suas práticas pedagógicas.

Com isso, percebemos que, no tocante à integração de programação e desenvolvimento do Pensamento Computacional por professores em suas aulas no ensino fundamental, é necessário que haja uma formação sólida do professor, isto é, que dê a ele a possibilidade de adquirir conhecimento suficiente que o encoraje a trabalhar com esses recursos em suas práticas pedagógicas, ou seja, muitas vezes não é a falta de vontade do professor, mas a falta de experiência e conhecimento em relação à programação que dificulta essa inserção em suas ações didáticas.

E, nesse ponto, esse espaço de formação possibilitou que os professores tivessem suas formações impactadas pela programação e pelo desenvolvimento do Pensamento Computacional, pois, ao elaborarem planos de aula pensando no desenvolvimento desse modo de pensar atrelado ao uso de programação, tiveram a oportunidade de conhecer e aprender sobre esses recursos, mobilizando conhecimentos sobre como modificar suas práticas pedagógicas de modo a incorporar esses recursos.

No nono encontro, os professores começaram a sentir mais segurança em relação ao trabalho com programação em seus projetos e iniciaram a construção de planos de aula para a (possível) aplicação dos projetos finalizados, com seus estudantes. Na próxima seção, apresentaremos uma dessas aulas que foram planejadas durante esse espaço de formação e que foi realizada com estudantes do sétimo ano do ensino fundamental, ministrada pela professora de Geografia e que envolveu, também, conteúdos de Matemática, programação e Pensamento Computacional.

No décimo encontro, finalizamos o espaço de formação, retomando com os professores todas as ações desenvolvidas durante todos os encontros e consolidando alguns aspectos teóricos sobre o Pensamento Computacional e o uso de programação em suas práticas pedagógicas, de modo que foi possível perceber o impacto disso para as suas formações como docentes. Coincidentemente, foi nesse mesmo dia que acompanhamos a professora de Geografia em sua aula no sétimo ano. A aula terminou apenas meia hora antes do início desse último encontro. Por causa disso, sugerimos à professora que relatasse essa experiência em sala de aula, com o uso do *Scratch*, para os demais professores.

Ela argumentou que, mesmo tendo sido contra a formação no início, por não ter interesse no software e nem na temática do Pensamento Computacional, é favorável a outras alternativas metodológicas em sala de aula. Dando sequência à sua fala, destacou ainda que se surpreendeu com o desempenho dos alunos, pois, no seu modo de ver, eles compreenderam rapidamente as funcionalidades do *Scratch*, alterando cenários, personagens, incrementando suas programações, e ainda foram bastante criativos em formular perguntas e respostas para os jogos criados (no formato de *quiz*, conforme detalharemos a seguir). Ela finalizou seu relato incentivando os demais professores a tentarem novas metodologias em suas aulas (não necessariamente com o *Scratch*). É importante mencionar que três desses professores trabalham com o *Scratch* em suas aulas de Matemática, e uma delas participa das pesquisas de mestrado mencionadas no início deste capítulo; não chegaram a se manifestar a esse respeito, mas deram algumas sugestões para o desenvolvimento de atividades futuras com esse software.

Em seguida, solicitamos aos professores que elencassem aspectos positivos e negativos da formação, para buscarmos um aprimoramento, e então sugeriram que, caso haja a continuidade desse espaço no ano que vem, que façamos a ambientação de forma análoga à que fizemos na aula da professora de Geografia. Mais sugestões foram dadas, por exemplo, um dos professores reiterou a importância de estudar os componentes da Microsoft Office, a importância de que os encontros da formação sejam menos espaçados etc. Assim, encerramos a formação na escola nesse ano de

2018, mas com a sensação de que estamos trilhando o caminho certo no que se refere à pesquisa acerca de formação de professores, pois, assim como argumenta Tardif (2000), saímos de nossos gabinetes dentro da universidade e nos juntamos aos professores, em seus locais de trabalho, para compreendermos o que eles fazem, pensam e são. Uma das consequências dessa iniciativa é a própria mobilização em sala de aula, que será discutida a seguir.

GEOGRAFIA E MATEMÁTICA: UM EXEMPLO DE INTERDISCIPLINARIDADE

Conforme mencionamos, uma professora participante do espaço de formação desenvolveu uma aula dupla com estudantes do sétimo ano do ensino fundamental, envolvendo programação no software Scratch e o desenvolvimento do pensamento computacional. O tema da aula foi Migração, particularmente estes tipos: migração pendular, migração sazonal ou transumância, migração permanente, êxodo rural, migração externa, migração interna, migração definitiva, migração temporária, migração espontânea e migração forçada.

Desenvolvemos em parceria com a professora o plano de aula que teve o objetivo de que os alunos aprofundassem seus conhecimentos acerca dessas migrações, compreendessem a noção de variável, por meio da funcionalidade de mesmo nome no *Scratch* e que aplicassem os conceitos de números inteiros na programação do *quiz*, que deveria ser criado nesse software.

Assim, no início da aula, foi solicitado aos alunos que se dividissem em pequenos grupos (duplas ou trios), e que estudassem os tipos de migrações: pendular, sazonal, permanente, exôdo rural, migração interna, definitiva, temporária, espontânea e forçada. Os pesquisadores fizeram uma breve ambientação com os alunos acerca das funcionalidades do *Scratch*, concentrando-se nas funcionalida-

des básicas para se constituir um *quiz*. Para isso, foi apresentado a eles um modelo de *quiz* pronto, e esclarecemos cada uma das funcionalidades envolvidas para a programação que havia sido feita.

Em seguida, foi solicitado a eles que programassem no *Scratch* um *quiz* com dez perguntas, uma sobre cada conteúdo, para que outro grupo respondesse posteriormente. A ideia era de que o grupo que obtivesse mais pontos ficasse em primeiro colocado. Algumas regras foram colocadas para a construção do jogo:

As perguntas deveriam ser feitas de modo que as respostas fossem simples, ou “sim”, ou “não”, ou os próprios nomes de cada migração, de modo que o *Scratch* reconhecesse as respostas, para que o jogo fosse justo para todos; quando estivessem respondendo ao *quiz* criado por outro grupo, a programação deveria estar oculta, e a cada resposta dada (certa ou errada), os alunos deveriam anotar em uma folha o que observavam em relação à resposta dada e a pontuação que obtiveram com essa resposta.

Em seguida, cada grupo foi convidado a relatar sobre o raciocínio e estratégias utilizados enquanto desenvolviam o *quiz*, ou seja: por que escolheram certas funcionalidades para a programação ao invés de outras, por que escolheram aquelas perguntas etc., de modo a identificar se houve o desenvolvimento do Pensamento Computacional durante o processo de preparo do *quiz* e como esse desenvolvimento contribuiu para a compreensão dos dez tipos de migração e a compreensão dos conceitos matemáticos envolvidos.

Essa aula não foi gravada, mas há relatos de diários de campo de dois pesquisadores e da própria professora, o qual ficou registrado no vídeo do último encontro da formação. Há também fotos que a professora tirou (ela tem autorização dos responsáveis dos alunos para publicar as imagens, entretanto compreendemos que essa autorização não se estende para fins de pesquisa), e todas as atividades desenvolvidas foram gravadas. A aula aconteceu no laboratório de informática, conforme mostra a Figura 1 a seguir:



Figura 1 – Aula de Geografia com Scratch na sala de informática

Fonte: autores – dados de pesquisa do projeto de extensão

Ao longo do desenvolvimento das aulas, os alunos nos chamavam para tirar as suas dúvidas, e quando percebemos que já havia terminado a primeira aula, começamos a nos preocupar se de fato conseguiríamos cumprir com a ideia da proposta, que seria eles trocarem de computadores entre os grupos, para que uns respondessem às perguntas dos outros. Assim, sugerimos a eles que fizessem o máximo de questões que conseguissem (e não necessariamente as dez questões, como havia sido planejado), para que tivessem tempo de responder às questões dos quiz de outros grupos. E assim aconteceu.

No entanto percebemos que quando eles foram responder os *quiz* dos outros grupos, pelo fato de alguns jogos estarem incompletos, não foi possível definir o grupo que teria a primeira colocação, assim, decidimos não fazer nenhum tipo de competição e investir no debate posterior às atividades, no qual os alunos destacaram os aspectos que mais lhes chamaram a atenção. Nesse momento, muitos disseram que gostaram da aula, porque propiciou uma forma diferente de estudo, mais dinâmica, além do fato de terem gostado muito de trabalhar com o software, pois suas funcionalidades permitem a criação de diferentes tipos de atividades, além de alterar cenários e personagens. Na Figura 2 a seguir, observamos o *quiz* criado por uma das duplas:

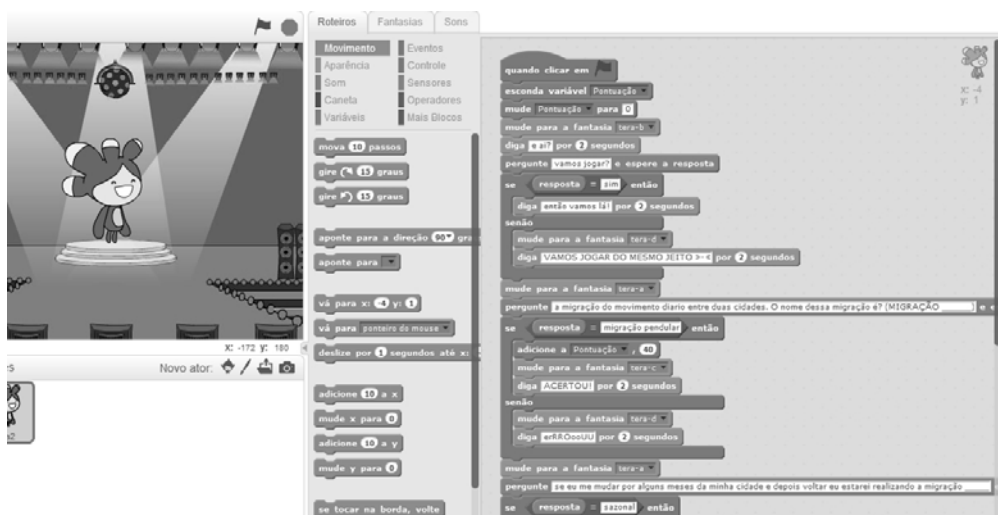


Figura 2 – Programação do *quiz* no Scratch

Fonte: Scratch – dados de pesquisa do projeto de extensão

Algo muito importante destacado pela professora, e que estamos focalizando em nossa análise, foi em relação ao processo de construção do *quiz* por cada grupo. O que mais nos chamou a atenção foi que houve uma mobilização diferente em sala de aula do que tradicionalmente acontece. Ou seja, ao invés dos alunos pensarem em responder perguntas (que é o mais comum em sala de aula), eles tiveram que se debruçar em estratégias para pensar em perguntas

significativas para colocarem no *quiz*, de forma a ter uma resposta simples, ou seja, sim ou não, verdadeiro ou falso, ou ainda, os nomes de cada tipo de migração. Ao mesmo tempo, embora tivessem respostas simples, as perguntas não poderiam ser tão “fáceis”, porque outros grupos teriam que respondê-las, e, até então, tratava-se de uma competição.

Isso os colocou como centro de seus próprios processos de aprendizagem, indo ao encontro do que é preconizado pela BNCC, no que tange a “assumir uma visão plural, singular e integral da criança, do adolescente, do adulto – considerando-os como sujeitos de aprendizagem” (Brasil, 2018, p. 12).

Além disso, os alunos mostraram criatividade ao efetuarem programações diferentes daquela que propusemos de modelo ao longo da ambientação, e também lançaram mão de estilos diversos para compor seus cenários e personagens. Essa criatividade pode ser observada na Figura 2 anterior.

Assim, junto com a professora, fizemos uma avaliação dessa aula posteriormente, e refinamos nosso planejamento para aplicar a mesma atividade para o outro sétimo ano na semana seguinte, o qual sofreu algumas alterações, motivadas principalmente pela sensibilidade e intuição da professora, que, por sua vez, tem pleno conhecimento acerca dos perfis de aprendizagem e dificuldades de seus alunos. Essa aula aconteceu na semana seguinte, e, no presente momento, estamos trabalhando na análise para divulgar resultados em publicações futuras.

CONSIDERAÇÕES FINAIS

Apresentamos neste estudo um espaço de formação desenvolvido com professores que ministram disciplinas em três grandes áreas – Ciências Humanas, Linguagens e Códigos e Ciências da Natureza e Matemática – em uma escola pública que atende estudantes do sexto ao nono ano do ensino fundamental, pertencente ao

Programa Ensino Integral. O objetivo desse espaço de formação foi constituir um trabalho colaborativo com esses professores de modo que pudessem conhecer e produzir conhecimento sobre a integração do Pensamento Computacional em suas abordagens pedagógicas, por meio do uso de softwares e materiais manipulativos, e sobre o uso de robótica nos processos de ensino e aprendizagem, além de promover reflexões acerca da interdisciplinaridade em suas práticas pedagógicas

O espaço de formação foi constituído por 10 encontros, em que realizamos com os professores uma ambientação com linguagem de programação por meio do software *Scratch*, discutindo suas potencialidades no ensino e refletindo sobre o uso de programação nas disciplinas que compõem o currículo dos anos finais do ensino fundamental. Realizamos a leitura de um artigo que aborda o tema Pensamento Computacional, para discutirmos com os professores sobre esse tema de modo que eles pudessem construir suas próprias concepções sobre o tema, mediante suas experiências didáticas em sala de aula, de modo a aproximar ainda mais o desenvolvimento desse aporte teórico ao currículo da Educação Básica.

E, ainda, discutimos com eles o trabalho com o Pensamento Computacional em uma perspectiva interdisciplinar, de modo a utilizar os recursos educacionais presentes na escola, como kits de robótica, no desenvolvimento desse pensamento, promovendo a interdisciplinaridade entre as áreas. No fim, foram desenvolvidos planos de aula para o trabalho com programação e Pensamento Computacional em sala de aula com seus estudantes.

Por último, apresentamos uma experiência fruto do espaço de formação que envolveu o ensino de Geografia e Matemática com estudantes do sétimo ano, utilizando programação e desenvolvendo o Pensamento Computacional desses estudantes no estudo de conteúdos dessas disciplinas curriculares. Vale destacar que essa experiência contribui para uma avaliação feita pela professora, em relação à aula, ao desempenho dos alunos, ao seu próprio desempenho e à formação proposta. Em suas palavras:

Gostei também da minha participação, afinal, no início como participante não gostava da formação no Scratch, mas no final amei [...]. Vocês foram sensacionais [se referindo à equipe proponente da formação] pois me mostraram o que eu não estava vendo! E isso é mérito de vocês! [...] achei importante também porque além de trabalharem com a tecnologia [se referindo aos alunos, durante essas aulas com Scratch] por meio do lúdico, vi a alegria, determinação, foco, entusiasmo, alcançaram objetivos, demonstraram empatia, mantiveram relações sociais positivas [...] (professora de GeoGrafia, relato de 5 de dezembro de 2018).

A professora ainda reiterou que percebeu que essa experiência contribuiu para que os alunos desenvolvessem os quatro pilares da Educação definidos em 1999, em relatório publicado pela Unesco: aprender a conhecer, aprender a fazer, aprender a conviver e aprender a ser (Sayão, 2008).

Ela também se surpreendeu com o desempenho de alguns alunos diagnosticados com problemas de *déficit* de aprendizagem, pois manifestaram interesse e iniciativas que dificilmente teriam acontecido em uma aula comum. Isso vai ao encontro do que defende a BNCC sobre a necessidade de seleção e aplicação de metodologias diversificadas em sala de aula, a fim de atender distintos grupos de alunos (Brasil, 2018).

Por fim, mesmo ainda sem terminar a análise de todos os dados produzidos nessa formação, acreditamos que, ao final, fomentamos o desenvolvimento do Pensamento Computacional por meio de abordagens interdisciplinares nas práticas pedagógicas dos professores da escola onde ocorreu esse espaço de formação, e ainda construímos junto com os professores reflexões que nos permitem aprofundar e expandir teoricamente a noção de Pensamento Computacional. Foi notório o impacto desse espaço de formação para o letramento em programação desses professores em suas formações e as contribuições dessas ações para as suas práticas pedagógicas em salas de aula com estudantes dos anos finais do ensino fundamental.

REFERÊNCIAS

- Brasil (2018). *Base Nacional Comum Curricular*. Ministério da Educação.
- D'Ambrosio, U. (2008). *Educação Matemática: da teoria à prática*. 16ª ed. Campinas: Papirus.
- Morais, A. D., Basso, M. V. A., & Fagundes, L. C. (2017). Educação Matemática & Ciência da Computação na escola: aprender a programar fomenta a aprendizagem de matemática? *Ciência & Educação (Bauru)*, v. 23, n. 2.
- Phillips, P. (2009). Computational Thinking: a problem-solving tool for every classroom. *Communications of the CSTA*, v. 3, n. 6, p. 12-16.
- São Paulo (2012). *Diretrizes do Programa Ensino Integral*. Recuperado em 24 jan. 2019, de <http://www.educacao.sp.gov.br/a2sitebox/arquivos/documentos/342.pdf>.
- São Paulo (2017). *Proposta pedagógica escolar 2017*. Recuperado em 24 jan. 2019, de <http://escolaintegral-cas.blogspot.com.br/p/proposta-pedagogica-escolar.html>.
- Sayão, S. C. (2008). O sentido do humano e os quatro pilares da educação da UNESCO: questões para se pensar sobre o ensino de Filosofia. Congresso Internacional de Filosofia: debate de ideias e cidadania, Caxias do Sul/RS. *Anais do VIII Simpósio Sul-Brasileiro sobre o Ensino de Filosofia: Filosofia, formação docente e cidadania*. v. 8. p.1.
- Silva, E. C. (2018). *Pensamento Computacional e a formação de conceitos nos Anos Finais do Ensino Fundamental: uma possibilidade com kits de robótica*. 264 p. Dissertação de Mestrado em Educação Matemática, Universidade Estadual Paulista Júlio de Mesquita Filho, Rio Claro.
- Silva, E. C., & Javaroni, S. L. (2018a). O Pensamento Computacional e a compreensão do conceito de congruência (módulo n) desenvolvido por duas estudantes. Congresso Internacional de Educação e Tecnologias / Encontro de Pesquisadores em Educação a Distância, 2018, São Carlos. *Anais CIET:EnPED:2018 – Educação e Tecnologias: Aprendizagem e construção do conhecimento*. S.l.: CIET:EnPED. v.4. p.1-17.
- Silva, E. C., & Javaroni, S. L. (2018b). Pensamento Computacional e atividades com robótica para a promoção da aprendizagem sobre o significado

do resto da divisão euclidiana. Simpósio Brasileiros de Informática na Educação, 2018, Fortaleza. *Anais do SBIE 2018 (Proceedings of the SBIE 2018)*. Porto Alegre: Sociedade Brasileira de Computação – SBC. p. 815-824.

Silva, F. F., & Javaroni, S. L. (2017). Resultados iniciais acerca do uso de tecnologias digitais no ensino de Matemática em escolas de Ensino Integral. Encontro Brasileiro de Estudantes de Pós-Graduação em Educação Matemática, 21., 2017, Pelotas. *Anais do 21o Encontro Brasileiro de Estudantes de Pós-Graduação em Educação Matemática*. Pelotas: Universidade Federal de Pelotas, p. 1-12. Recuperado em 31 jan. 2019, de <https://wp.ufpel.edu.br/xxiebrapem/anais-xxi-ebrapem-2/>.

Silva, M. M., Miorelli, S. T., & Kologeski, A. L. (2018, abr.). Estimulando o pensamento computacional com o projeto logicando. *Revista Observatório*, [s. l.], v. 4, n. 3, p. 206-238. Recuperado em 21 maio 2018, de <https://sistemas.uft.edu.br/periodicos/index.php/observatorio/article/view/4080>.

Sousa, R. M., & Lencastre, J. A. (2014). Scratch: uma opção válida para desenvolver o pensamento computacional e a competência de resolução de problemas. In Carvalho, A. A. A. et al. (org.). *Atas do 2º encontro sobre jogos e mobile learning*. Braga: CIEd, p. 256-267. Recuperado em 29 nov. 2017, de <http://repositorium.sdum.uminho.pt/handle/1822/29944>.

Souza, P. H. G. (2018). Modelagem Matemática, Pensamento Computacional e suas relações. Encontro Brasileiro de Estudantes de Pós-Graduação em Educação Matemática, 22., 2018, Belo Horizonte. *Anais do 22o Encontro Brasileiro de Estudantes de Pós-Graduação em Educação Matemática*. Belo Horizonte: Universidade Federal de Minas Gerais, Faculdade de Educação, p. 1-11. Recuperado em 7 dez. 2018, de <http://sbem.iuri0094.hospedagemdesites.ws/eventos/index.php/EBRAPEM/index/pages/view/anais2018?fbclid=IwAR0RaTEl0xdJ2VrYfRA1ykSjLfkynz8Y0wcmYY5rT9lr0sHGfeuqZIldPM>.

Tardif, M. (2000). Saberes profissionais dos professores e conhecimentos universitários: Elementos para uma epistemologia da prática profissional dos professores e suas consequências em relação à formação para o magistério. *Revista Brasileira de Educação*, n. 13.

Tomaz, V. S., & David, M. M. M. S. (2017). *Interdisciplinaridade e aprendizagem da Matemática em sala de aula*. 3. ed. Belo Horizonte: Autêntica.

Wing, J. M. (2006). Computational thinking. *Commun. ACM*, 49(3), p. 33-35.

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical transactions of the royal society of London A: mathematical, physical and engineering sciences*, 366(1881), p. 3717-3725.

Wing, J. M. (2016, maio/ago.). Pensamento Computacional – Um conjunto de atitudes e habilidades que todos, não só cientistas da computação, ficaram ansiosos para aprender e usar. *Revista Brasileira de Ensino, Ciência e Tecnologia (RBECT)*, Ponta Grossa, v. 9, n. 2, p. 1-10.

CAPÍTULO 8

ESTIMULANDO O PENSAMENTO COMPUTACIONAL POR MEIO DE OFICINAS LÚDICAS NO LITORAL NORTE GAÚCHO

Vitória de Souza Fabrício

Maurício Braga Julio

Augusto Weiland

Anelise Lemke Kologeski

INTRODUÇÃO

O uso do pensamento computacional se faz cada vez mais necessário para a execução das mais diversas atividades do cotidiano, uma vez que o uso do computador se tornou essencial em nossas vidas atualmente. Conforme o aprendizado da linguagem de programação e o desenvolvimento do raciocínio lógico são estimulados e introduzidos na vida de aluno, desde cedo, podem servir de auxílio em diversos aspectos, facilitando a aprendizagem do estudante, contribuindo para a tomada de decisões e resolução de problemas do dia a dia, promovendo um aumento na capacidade do pensamento de forma criativa e para o desenvolvimento das mais diversas habilidades (Alvarez, 2018). Por esse motivo, o projeto “Programando Fácil: Conhecendo a Computação” visa a inserir os conceitos de programação básica nas escolas de ensino fundamental, a partir da prática extensionista, estimulando a participação dos estudantes por intermédio de atividades lúdicas que despertem a curiosidade, fazendo uso do raciocínio lógico e do pensamento computacional, a fim de contribuir para um aprendizado mais rico e cheio de significados.

De acordo com a Base Nacional Comum Curricular (BNCC, 2019), o letramento matemático assegura aos alunos reconhecer que os conhecimentos matemáticos são fundamentais para a compreensão e a atuação no mundo e perceber o caráter de jogo intelectual da matemática, como aspecto que favorece o desenvolvimento do raciocínio lógico e crítico, estimulando a investigação e o prazer. O

desenvolvimento dessas habilidades está intrinsecamente relacionado com algumas formas de organização da aprendizagem matemática, com base na análise de situações da vida cotidiana, de outras áreas do conhecimento e da própria Matemática (BNCC, 2019). Os processos matemáticos de resolução de problemas, de investigação, de desenvolvimento de projetos e da modelagem podem ser citados como formas privilegiadas da atividade matemática, motivo pelo qual são, ao mesmo tempo, objeto e estratégia para a aprendizagem ao longo de todo o ensino fundamental. Esses processos de aprendizagem são potencialmente ricos para o desenvolvimento de competências fundamentais para o letramento matemático (raciocínio, representação, comunicação e argumentação) e para o desenvolvimento do pensamento computacional (BNCC, 2019), e esse é um dos motivos pelos quais optamos por realizar as atividades do projeto com turmas de ensino fundamental, e com conteúdos relacionados ao ensino da matemática.

Então, nosso projeto oferece oficinas com programação básica, permitindo relacionar diferentes conteúdos juntamente com as atividades, envolvendo principalmente noções matemáticas, interpretação de texto, raciocínio lógico e pensamento computacional, favorecendo uma educação integral, como defende a BNCC (2019). Assim, as atividades propostas pelo projeto visam, de uma maneira geral, a estimular o desenvolvimento da lógica e da criatividade a partir do uso de recursos digitais, trabalhando também com a correta compreensão de um enunciado, fazendo uso de temas e personagens de acordo com o interesse dos professores que acompanham essas oficinas. Independentemente da duração da jornada escolar, o conceito de educação integral com o qual a BNCC está comprometida se refere à construção intencional de processos educativos que promovam aprendizagens sintonizadas com as necessidades, as possibilidades e os interesses dos estudantes e, também, com os desafios da sociedade contemporânea (BNCC, 2019).

Para o desenvolvimento do raciocínio lógico e do pensamento computacional, optamos por utilizar um recurso disponível gratuitamente, em diversas línguas, por meio da plataforma Code.org (Code, 2018), que permite a inicialização dos usuários de todas as idades em conceitos relacionados à programação, com o envolvimento de

jogos digitais que incluem temáticas e personagens que se destacam na faixa etária do público-alvo, como por exemplo, Star Wars, Frozen, Minecraft e Angry Birds.

Outra forma de contextualizar nosso projeto ocorre por intermédio dos dados divulgados no ano de 2018 pelo Índice de Desenvolvimento da Educação Básica Ideb (Ideb, 2018), que são de grande relevância, conforme apresenta a Tabela I para as séries finais do ensino fundamental. O Ideb é obtido a partir da Prova Brasil, realizada pelo Governo Federal, e visa a analisar a qualidade do ensino oferecido pelo sistema educacional brasileiro, levando em conta dois aspectos para a promoção da qualidade da educação brasileira: a aprovação escolar e o desempenho dos estudantes em língua portuguesa e matemática, durante a Educação Básica. Como se pode observar, de acordo com a Tabela I, nenhuma das três últimas edições da prova conseguiu atingir a meta nacional esperada, considerando as séries finais do ensino fundamental, independentemente de serem da rede estadual ou municipal, pública ou privada. Isso deixa evidente que algo precisa ser feito para que a qualidade da educação seja melhorada. Dessa forma, nosso projeto visa a desenvolver importantes habilidades de raciocínio lógico e pensamento computacional, a fim de impactar diretamente no aprendizado dos estudantes, na tomada de decisões, na compreensão de enunciados e na capacidade de raciocínio e autonomia dos estudantes, contribuindo, assim, para a promoção de melhorias na educação, em sala de aula, oferecendo novas e diversificadas estratégias de ensino aos alunos e aos professores participantes, promovendo uma educação integral e com mais qualidade.

Rede	2013 Atingido	2013 Esperado	2015 Atingido	2015 Esperado	2017 Atingido	2017 Esperado
Estadual	4.0	4.2	4.2	4.5	4.5	4.8
Municipal	3.8	3.9	4.1	4.3	4.3	4.6
Privada	5.9	6.5	6.1	6.8	6.4	7.0
Pública	4.0	4.1	4.2	4.5	4.4	4.7

Tabela 1 – Ideb atingido e esperado das três últimas edições, para as séries finais do ensino fundamental, no Brasil (Ideb, 2018)

Fonte: Ideb, 2018

Por meio da realização das oficinas, apresentamos aos participantes do projeto um pouco da área tecnológica, a partir do uso de jogos digitais, a fim de despertar o interesse e a curiosidade deles pela Informática. Nesse sentido, a criação de um projeto como esse constitui também uma ferramenta importante para inserir os alunos do Curso Técnico de Informática, que são bolsistas do projeto, em contato com os alunos da Educação Básica, a fim de promover mais qualidade para a educação, colocando em prática o aprendizado obtido em sala de aula, oferecendo novas experiências aos participantes. Com isso, promovemos o aprendizado de ambas as partes, capacitando tecnicamente os bolsistas envolvidos no projeto, apresentando novos horizontes aos alunos da Educação Básica, oferecendo também visibilidade para a nossa instituição, e proporcionando o compartilhamento de conhecimento entre todos os envolvidos, fomentando a relação indissociável entre o ensino, a pesquisa e a extensão. Assim, o presente projeto também coloca em prática o aprendizado dos estudantes do Curso Técnico de Informática, oportunizando a eles o contato com a comunidade, permitindo a integração da comunidade com a instituição de ensino, e tornando-se uma forma de experiência para os bolsistas envolvidos, proporcionando a todos os envolvidos uma educação integral e de qualidade.

Os responsáveis pela elaboração e aplicação das oficinas desse projeto de extensão no ano de 2018 foram quatro bolsistas, todos alunos do segundo ano do curso Técnico de Informática Integrado ao Ensino Médio, juntamente com a orientação da professora coordenadora do projeto, e com a coorientação de um técnico em Tecnologia da Informação. Em conjunto, o grupo planeja e aplica as oficinas, com duração média aproximada de duas horas, realizando as atividades de forma dinâmica e lúdica, sob a supervisão dos professores da Educação Básica que atuam nas turmas envolvidas.

Este estudo está organizado da seguinte maneira: na Seção 2, serão apresentados trabalhos similares ao nosso projeto; na Seção 3, a metodologia utilizada no desenvolvimento e realização das oficinas é descrita; na Seção 4, os resultados obtidos no ano de 2018 são apresentados; e, para finalizar, na Seção 5, encontram-se as considerações finais, incluindo as conclusões e as sugestões de trabalhos que poderão ser desenvolvidos futuramente para dar continuidade ao projeto.

TRABALHOS RELACIONADOS

Muitos trabalhos já foram desenvolvidos para promover melhorias na educação, por intermédio da inserção de programação básica no ensino fundamental. Por isso, levar novos recursos de ensino e aprendizagem para dentro da sala de aula, a partir de projetos de extensão, com o uso de jogos, que promovam o raciocínio lógico e o pensamento computacional, não é um assunto novo. De acordo com França e colaboradores (2015), é necessário aos estudantes que desenvolvam diversas habilidades, como o pensamento computacional, por exemplo, e defendem que a integração desse conteúdo deve ser feita ao currículo escolar. Indo ao encontro dessa ideia, atendemos nossa comunidade local, promovendo melhorias pontuais que estão ao nosso alcance, por meio da prática da extensão, apresentando esse conteúdo aos alunos da Educação Básica no formato de oficinas. Muitos projetos são similares ao nosso, e utilizam metodologias e ferramentas semelhantes, como será descrito a seguir. Contudo cada projeto tenta contribuir para a melhoria da educação em suas respectivas comunidades, dentro do seu alcance, abrangendo as mais diversas regiões do Brasil, trazendo benefícios para a comunidade local.

No trabalho de Gomes (2015), os autores afirmam que os jogos permitem possibilidades inovadoras para apresentar conteúdos de maneira mais atraente e motivadora, em sala de aula, fazendo uso da interatividade. Os autores aplicam uma oficina para estimular o pensamento computacional, com a ajuda de um jogo chamado *The Foos*, para crianças acima de 5 anos, do ensino infantil. Nesse jogo, os movimentos do personagem são realizados a partir de blocos de instruções que representam ações como: andar, pular, capturar um objeto, lançar um objeto, laços de repetição e condicionais, por exemplo.

Já o projeto Logicando (Kologeski, 2016), implementado no ano de 2016, teve por finalidade oferecer oficinas lúdicas para os alunos de 8º e 9º anos do ensino fundamental, a fim de trabalhar e desenvolver o pensamento computacional por intermédio do uso de múltiplas ferramentas da Tecnologia da Informação e Comunicação. Os autores do projeto, desenvolvido na Universidade Feevale, em Novo

Hamburgo, no Rio Grande do Sul, tinham como principal propósito contribuir para um melhor aprendizado dos alunos, bem como trabalhar com o desenvolvimento de habilidades e competências para a resolução de problemas do cotidiano. Os mais diversos assuntos foram abordados nas oficinas, relacionando as mais diversificadas áreas do conhecimento. As atividades foram divididas em seis oficinas de 1h30min cada, oferecidas na escola ou nos laboratórios de informática da própria universidade. Para a compreensão do conteúdo abordado em cada uma das oficinas desenvolvidas, diferentes plataformas foram utilizadas, a saber: Code.org (Code, 2018), Scratch (2018), MIT App Inventor (2018), Jogo Educativo Navegática (Barbosa & Bassani, 2013), Jogo Educativo Pantanal (Cardoso & Barbosa, 2016) e a Rede Social Teia (Winter, 2014). Dessa forma, os participantes das oficinas puderam vivenciar situações que envolveram diversas competências e habilidades, conhecendo e aprendendo noções de algoritmos e a sequência de passos necessária para a realização de uma determinada atividade, bem como o uso de comandos de repetição e controle de variáveis por meio de comparações. Em geral, a participação dos estudantes foi muito satisfatória, e o retorno recebido após o desenvolvimento das atividades, tanto dos estudantes quanto dos professores envolvidos, foi bastante positivo, e pode ser comprovado a partir dos resultados quantitativos apresentados no trabalho, que mostram uma melhoria de até 45% na interpretação dos enunciados trabalhados durante as oficinas, atendendo 141 participantes, de nove escolas.

No município de Ibirama, em Santa Catarina, outro projeto similar foi desenvolvido, promovendo um ambiente de ensino de pensamento computacional para alunos das séries finais do ensino fundamental (7º, 8º e 9º anos), atendendo 34 pessoas (Schoeffel, 2015). A carga horária do curso foi de 48 horas, sendo aplicado em dois dias da semana, totalizando oito semanas. O projeto de Schoeffel (2015) também avaliou a participação dos alunos na Olimpíada Brasileira de Informática. A metodologia completa utilizada pelos autores abrange três etapas, das quais apenas a primeira está descrita em Schoeffel *et al.* (2015): ensino da lógica computacional por intermédio do raciocínio lógico e pensamento computacional, de forma lúdica e prática; programação de computadores, com a introdução ao ensino

de programação e de linguagens de programação; e robótica, com o ensino e a prática de montagem e programação de computadores, utilizando os kits LEGO Mindstorms®. A partir de uma avaliação da percepção dos alunos, identificou-se que, em geral, os alunos acharam as aulas fáceis e divertidas, e que a maioria deles deseja obter mais conhecimento sobre a Computação. Esses resultados apresentam evidências de que as metodologias de ensino-aprendizagem utilizadas tiveram êxito ao facilitar a aprendizagem e tornar o processo divertido. Além disso, os resultados indicam que a participação dos alunos no curso provocou interesse nesses estudantes para darem continuidade aos estudos na área de Computação.

No trabalho de Andrade (2016), da Universidade Federal da Paraíba, foram realizadas oficinas com o uso de uma plataforma digital para a elaboração de jogos, feita pelos próprios alunos. Porém o foco do projeto foi trabalhar com alunos que já possuem alguma familiaridade com a Internet e jogos eletrônicos, ensinando os conceitos principais da programação, como operadores, dados, condicionais, e repetições, dentre outros. Foram cinco aulas presenciais com três horas cada, ofertadas para alunos de ensino médio, com o auxílio da plataforma Scratch. Na aula piloto, os alunos deveriam desenvolver a criatividade e criar o próprio jogo. A oficina também teve como foco a competitividade entre os alunos para despertar seu interesse: seria premiado o aluno que desenvolvesse o melhor jogo. Dos participantes, 93% declararam que a metodologia despertou o interesse pelo desenvolvimento de jogos. Além disso, os autores de Andrade (2016) comentam que a iniciativa surgiu após a identificação de uma queda de 25% de interesse e procura no curso de graduação oferecido na instituição, aproveitando essa oportunidade para divulgação do curso entre os jovens.

O Campus Osório do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul está localizado na região do litoral-norte do estado do Rio Grande do Sul, e atende uma região composta por 23 municípios, totalizando em torno de 335 mil habitantes (PPC, 2019). O Campus Osório do IFRS entende como sua função primeira promover educação científica, tecnológica e humanística de qualidade, visando à formação de cidadãos críticos, conscientes e atuantes, competentes técnica e eticamente, comprometidos

efetivamente com as transformações sociais, políticas, culturais e ambientais, e que entendam a sua atuação no mundo do trabalho em prol de uma sociedade mais justa e igualitária (PPC, 2019). Em comparação aos projetos citados, o nosso projeto de extensão tem como objetivo aplicar oficinas similares, utilizando a plataforma Code.org (Code, 2018), no litoral-norte do Rio Grande do Sul, focando nas séries finais do ensino fundamental. Junto com a oferta das oficinas, aproveitamos também para divulgar nossa instituição para a comunidade, apresentando os cursos que ela oferece. Além disso, nossas oficinas são aplicadas em um único encontro com cada turma participante, pois foram encontrados muitos obstáculos para o deslocamento dos estudantes em nossa região, conforme será mencionado posteriormente.

DESENVOLVIMENTO DAS OFICINAS

Com o projeto de extensão “Programando Fácil: Conhecendo a Computação”, são oferecidas oficinas para os alunos das escolas da região do litoral-norte gaúcho, convidando e priorizando a participação de turmas que estejam nas séries finais do ensino fundamental, justamente porque os alunos já possuem uma maior capacidade de abstração e compreensão, para a introdução dos conceitos básicos de programação e de raciocínio lógico. Além disso, os alunos participantes das oficinas ingressarão em breve no ensino médio, e dessa forma nós apresentamos possibilidades para auxiliá-los na escolha de qual curso seguir, como por exemplo o curso Técnico em Informática Integrado ao Ensino Médio, ou o curso Técnico em Administração Integrado ao Ensino Médio, ambos oferecidos pela nossa instituição de ensino, de forma gratuita.

Conforme interesse e disponibilidade das escolas envolvidas, as oficinas podem ser realizadas tanto nas dependências das escolas parceiras quanto na nossa própria instituição de ensino. Essa decisão depende apenas das escolas parceiras, que, diante das dificuldades de transporte, podem optar por não se deslocarem até nossa

instituição, realizando as oficinas no laboratório de informática da própria escola. Geralmente, os computadores utilizados nas atividades independem de configurações de hardware muito específicas, precisando apenas de recursos mínimos de processamento, e de acesso à Internet.

Cada oficina é introduzida com a realização de um teste inicial, chamado de pré-teste, que tem por objetivo analisar o nível de conhecimento prévio dos alunos, antes da realização da oficina. Com base no resultado desse teste, acompanhado em tempo real, podemos organizar a complexidade dos conteúdos que serão trabalhados e aprofundados durante a oficina, dosando também a quantidade de material disponibilizado de acordo com o desempenho progressivo dos alunos. Após o término da oficina, outro teste é também realizado, chamado de pós-teste, possibilitando analisar o desempenho obtido pelo aluno durante as atividades desenvolvidas. Tanto o pré-teste quanto o pós-teste são disponibilizados pelo tempo máximo de 10 minutos, a partir da plataforma oferecida via Google Forms (2018), e são compostos por cinco questões, idênticas entre o pré e o pós-teste. Essas questões são baseadas nas atividades trabalhadas durante a realização da oficina, fazendo com que inicialmente o pré-teste apresente ao aluno as atividades que serão desenvolvidas, e que ao final o pós-teste possa ser utilizado para mensurar e comparar o desempenho dos estudantes, observando se eles conseguiram realizar a correta compreensão dos conceitos, comandos básicos e enunciados que foram desenvolvidos durante as oficinas.

As oficinas são desenvolvidas com base na plataforma Code.org (2018), que consiste em um endereço eletrônico em que o objetivo é ensinar a lógica de programação para alunos de todas as idades. O trabalho de Cavalcanti (2016) apresenta o funcionamento de toda a plataforma Code.org, mostrando que ela pode ser inserida como um curso de programação introdutório, bem como apresentando cada conceito utilizado na programação que é abordado pela plataforma. De acordo com a proposta de Cavalcanti (2016), o uso da plataforma abrange os seguintes conceitos computacionais: sequência, em que os alunos identificam uma sequência de passos para atingir um objetivo; repetição, em que os alunos devem perceber um padrão de

comandos que se repete e utilizar uma estrutura para este; eventos, em que os alunos deverão perceber os possíveis eventos e as ações para cada evento; paralelismo, que consiste em um conjunto de eventos funcionando ao mesmo tempo, de maneira paralela; condicionais, em que será necessário identificar algum caso específico para que seja exercido algum comando; e, por último, operadores e dados, vistos em todos os processos de programação, em que os alunos deverão determinar as constantes e as variáveis de um código.

Ainda, de acordo com Cavalcanti (2016), a plataforma Code.org possui como características relevantes o fato de ser gratuita, em português, e direcionada para o público iniciante, tendo seu conteúdo metodológico previamente definido e organizado dentro da plataforma, de acordo com as preferências de cada usuário. Desde o lançamento em 2013, mais de 200 milhões de alunos utilizaram o Code.org (Code, 2018), sendo 49% dos usuários do sexo feminino, fomentando também a participação de meninas na área da Ciência da Computação. Apesar do grande número de usuários, são poucas as pesquisas nacionais sobre as vantagens do ensino de programação ofertada pela plataforma (Dantas & Costa, 2013).

Em geral, as oficinas acontecem no horário disponibilizado pela escola, sendo majoritariamente no respectivo horário de aula das turmas participantes. Sendo assim, as atividades trabalhadas variam de acordo com o conteúdo que os professores responsáveis pelas turmas querem propor aos alunos. Se, por exemplo, o professor responsável pelos alunos durante a oficina for o professor de Matemática, podemos realizar atividades que sejam mais voltadas para essa disciplina, como por exemplo, atividades com o uso de formas geométricas e ângulos. Em geral, todos os jogos envolvem a capacidade de interpretação de texto do estudante, bem como o raciocínio lógico, para que o aluno consiga atingir os objetivos do jogo.

Uma grande diversidade de atividades está disponível na plataforma Code.org, e são organizadas para diferentes faixas etárias. A escolha das atividades que foram desenvolvidas nas oficinas foi proposta pelos próprios autores do projeto, após os alunos bolsistas as explorarem e realizarem, elegendo aquelas que a maioria julgou mais agradáveis para o público-alvo. Nesse caso, como já dito anteriormente, parte do grupo que compõe o projeto são alunos do

ensino médio, e por isso essa escolha das atividades foi realizada de forma muito fácil, já que a experiência das séries finais do ensino fundamental ainda é bastante recente para eles. Com base nessa decisão, as atividades definidas para serem trabalhadas durante as oficinas foram os seguintes jogos digitais, todos da plataforma Code.org (2018): A Fazendeira, O Labirinto, Minecraft e Star Wars. A escolha se deu, diante da ordem apresentada, pelo nível de dificuldade e complexidade de cada jogo, conforme apresentamos a seguir.

Na Figura 1, temos um exemplo de parte do jogo “A Fazendeira”. Basicamente, os conceitos introduzidos nessa atividade levam o aluno a conduzir a personagem por um caminho no qual é preciso preencher ou remover blocos de terra. Para isso, são apresentados os comandos “preencha 1” e “remova 1”. Além disso, comandos como “avance”, “vire à direita”, “vire à esquerda”, “repita X vezes”, “se” e “enquanto houver” também são necessários para a movimentação da personagem, fazendo com que o aluno identifique e ordene a sequência necessária de comandos para atingir o objetivo proposto pelo jogo, descrito no enunciado que aparece na parte superior da Figura 1. Cada jogo é composto por diversas fases, que podem ser visualizadas nas bolinhas que aparecem na barra superior da Figura 1. Quando a fase é concluída corretamente, a bolinha recebe destaque com a cor verde. Conforme o aluno avança as fases, a complexidade do jogo aumenta.

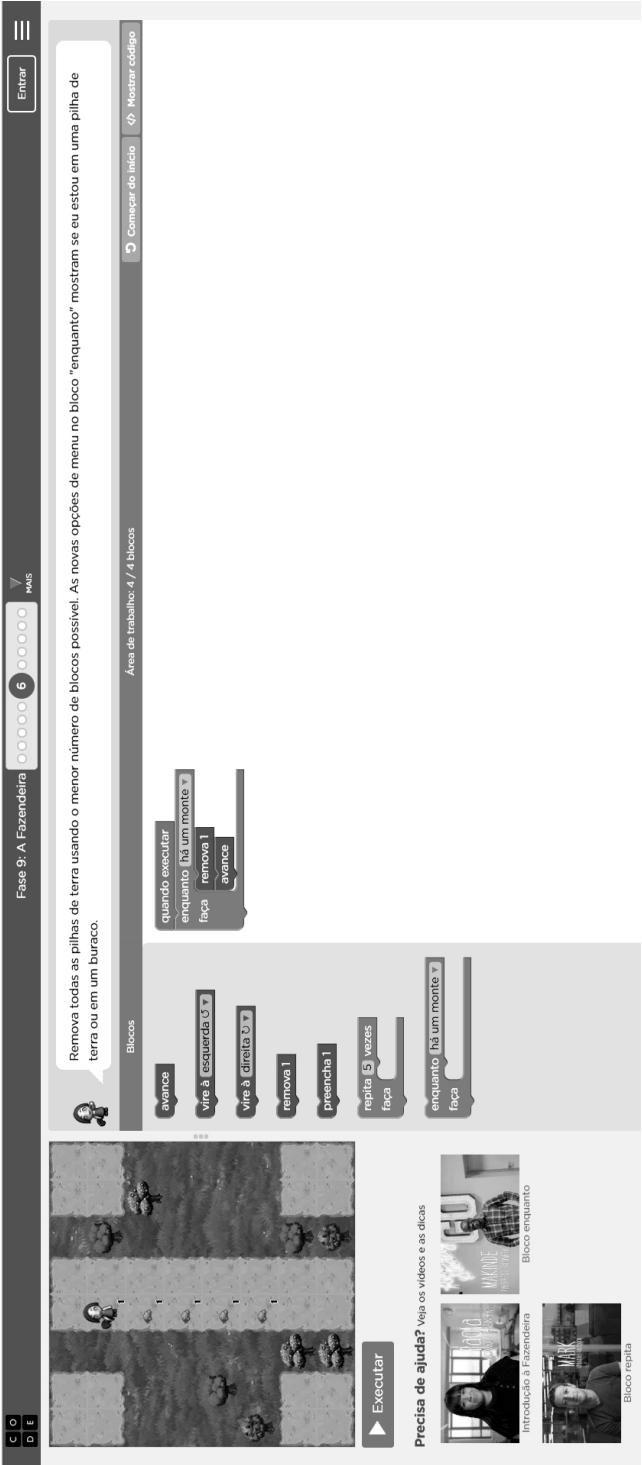


Figura 1 – Cena do jogo “A Fazendaira”
Fonte: plataforma Code.org (2018)

Na Figura 2, temos um exemplo de tela do jogo “O Labirinto”. Nesse jogo, novos comandos são introduzidos, como o “repita até”, “se... faça...” e “se...faça...senão”, aumentando a dificuldade para o aluno identificar a sequência correta de comandos necessários, diante dos diversos formatos de labirintos que são propostos ao estudante. Para esse jogo, mais fases são introduzidas, aumentando o nível de exigência para os jogadores. Os personagens envolvidos nesse jogo são divididos em dois grupos: “Angry Birds” e “Plants vs Zombies”.

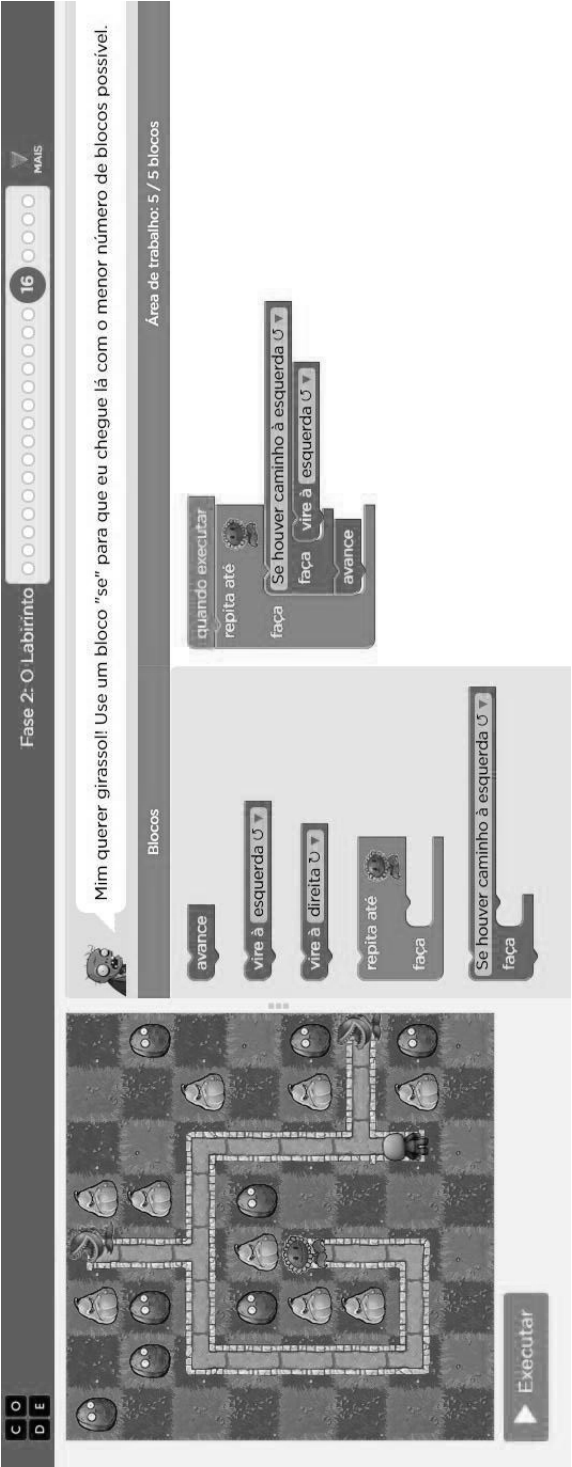


Figura 2 – Cena do jogo “Labirinto”
Fonte: plataforma Code.org (2018)

Na Figura 3, um exemplo de tela do jogo “Minecraft” é apresentado. Muitas das funcionalidades do jogo original são reproduzidas nesse cenário, como por exemplo a “placa de pressão”, que funciona como um interruptor, e os “easter eggs”, que proporcionam recompensas ao jogador. Na Figura 4, apresentamos a primeira fase do jogo “Star Wars”, no qual o aluno começa a aprofundar os seus conhecimentos por meio da programação, criando o seu próprio jogo na fase final, atribuindo pontuação, cenário, objetos e movimento próprio aos personagens de acordo com as suas preferências. Esse jogo possui a interação de setas de comando, sendo o personagem movimentado por meio do teclado para completar os seus objetivos, introduzindo o conceito de eventos em programação. Além disso, o jogo também trabalha com a consciência ecológica, abordando assuntos como a coleta de lixo, por exemplo.

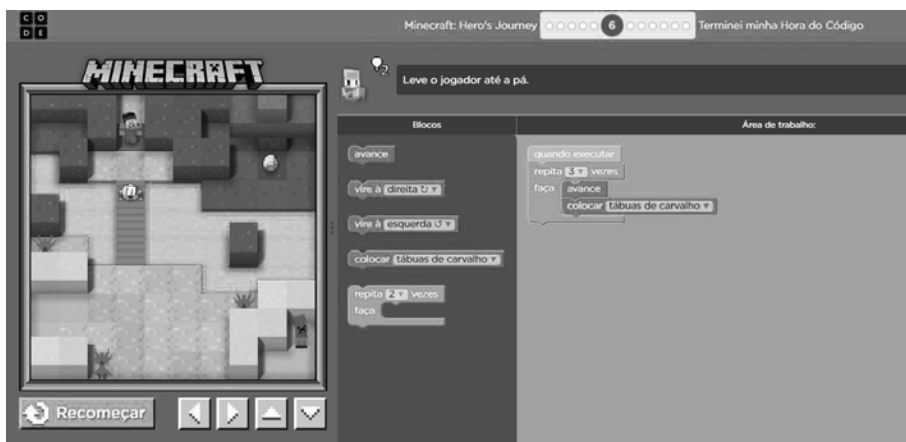


Figura 3 – Cena do jogo “Minecraft”

Fonte: plataforma Code.org (2018)



Figura 4 – Cena do jogo “Star Wars”

Fonte: plataforma Code.org (2018)

Durante as oficinas, os alunos participantes são auxiliados pelos alunos bolsistas do projeto, e cada aluno vence as etapas do jogo no seu próprio ritmo, avançando as fases por intermédio dos conhecimentos adquiridos durante o próprio jogo, pedindo auxílio aos bolsistas quando enfrentam maiores dificuldades. Nas primeiras fases dos jogos apresentados aqui, o aluno deverá apenas observar o mapa (apresentado na esquerda de cada imagem), quais os comandos estão disponíveis para a utilização (coluna do meio), e programar o que o enunciado solicita com os comandos disponíveis (coluna da direita). Para todos os jogos (exceto o Star Wars na fase final, que propõe a construção do próprio jogo), a quantidade ideal de comandos para ser utilizada é informada ao estudante, para que ele possa desenvolver a programação de forma otimizada, usando o mínimo possível de comandos.

Juntamente com a aplicação das oficinas, uma visita guiada foi proposta aos participantes, que vieram até nossa instituição de ensino, convidando-os a conhecer as dependências e espaços no IFRS Campus Osório, como por exemplo: a sala de música, os laboratórios de informática, matemática, física e química, a sala de educação física, os laboratórios de línguas e Libras, a biblioteca, os auditórios, a sala de convivência, o espaço destinado à cantina,

dentre outros. Assim, os participantes do nosso projeto tiveram a oportunidade de conhecer a nossa instituição, podendo decidir se gostariam de realizar o processo seletivo para ingressar em algum dos cursos oferecidos pelo IFRS – Campus Osório.

RESULTADOS

A maioria dos estudantes que participaram das oficinas tem idade entre 14 e 16 anos, conforme mostra a Figura 5. Até o presente momento, nosso projeto já atendeu 142 participantes, de nove instituições, localizadas em cinco cidades diferentes, conforme apresentado na Figura 6. Todas as oficinas promovidas pelo projeto de extensão fornecem certificados de participação.

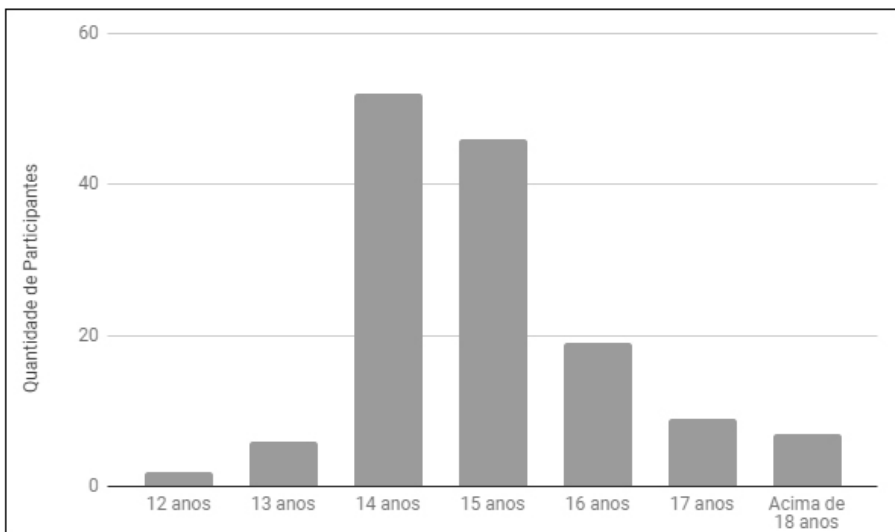


Figura 5 – Relação da quantidade de participantes das oficinas de acordo com a idade

Fonte: autores

Escola	Data
Escola Estadual de Ensino Fundamental Professora Suely Vacari Osório (Tramandaí/RS)	5 jul. 2018
Projeto STEM GEEK (Turma 1) – incluindo alunos da Escola Estadual General Osório (Osório/RS) –, Escola Estadual Raul Pilla (Cidreira/RS), Escola Estadual de Ensino Fundamental Visconde do Rio Branco (Santo Antônio da Patrulha) e do próprio IFRS, alunos do curso Técnico de Administração (Osório/RS).	5 set. 2018
Escola Estadual de Ensino Médio Albatroz (Osório/RS)	12 set. 2018
Escola Municipal de Ensino Fundamental Osvaldo Bastos (Osório/RS)	17 set. 2018
Escola Estadual Básica Professora Erica Marques (Terra de Areia/RS) (Turma 1)	18 out. 2018
Escola Estadual Básica Professora Erica Marques (Terra de Areia/RS) (Turma 2)	18 out. 2018
Projeto STEM GEEK (Turma 2) – incluindo alunos da Escola Estadual General Osório (Osório/RS) –, Escola Estadual Raul Pilla (Cidreira/RS), Escola Estadual de Ensino Fundamental Visconde do Rio Branco (Santo Antônio da Patrulha) e do próprio IFRS, alunos do curso Técnico de Administração (Osório/RS).	8 nov. 2018

Figura 6 – Escolas participantes e data de realização das oficinas

Fonte: autores

A métrica utilizada para mensurar o aprendizado dos alunos foi a comparação entre o pré e o pós-teste, aplicados no início e no final de cada oficina, respectivamente, compostos por cinco questões objetivas, idênticas entre os testes, que envolvem os jogos e comandos trabalhados durante a realização das oficinas. As questões utilizadas no pré e no pós-teste são apresentadas na Figura 7. Dessa forma, a partir das respostas fornecidas para os testes, comparamos como é o conhecimento prévio dos estudantes, e se realmente houve alguma contribuição para o aprendizado deles durante as oficinas, verificando se houve alguma influência positiva nos resultados obtidos.

1. O objetivo da fazendeira na imagem abaixo é remover o bloco de terra com o número "1". Para ela conseguir realizar essa função, você concorda que ela precisará somente de um bloco "avance" e de um bloco "remova 1", como apresentado abaixo?

1 ponto



quando executar
avance
remova 1

- ☐ Sim
☐ Não

2. O Pássaro Red deseja pegar o porco malvado. Para isso, ele precisa andar em frente e dobrar na primeira rua que encontrar, e continuar andando em frente. Seguindo os passos abaixo ele terá sucesso em capturar o porco malvado?

1 ponto



quando executar
avance
vire à esquerda 90°
avance

- ☐ Sim
☐ Não

3. Red está tentando capturar o porquinho malvado e para isso precisa passar pelo mapa abaixo. Para chegar até lá, ele precisa fazer uma série de movimentos mostrados ao lado do mapa: Você concorda que, com os movimentos abaixo, Red chegará ao porquinho malvado?



quando executar
avance
vire à direita 90°
avance
vire à esquerda 90°

- ☐ Sim
☐ Não

4. Os blocos de comando ao lado direito estão sendo utilizados adequadamente para que o zumbi chegue até a planta?

1 ponto



quando executar
repita até
faça
vire à direita 90°
avance
vire à esquerda 90°
avance

- ☐ Sim
☐ Não

5. O Scrat está com fome, então ele precisa caminhar até a noz. Se ele percorrer este caminho de acordo com o código indicado ao lado direito da imagem, você concorda que Scrat irá se alimentar?

1 ponto



quando executar
repita até
faça
se houver caminho à frente
faça
se não
vire à direita 90°

- ☐ Sim
☐ Não

Figura 7 – Questões do pré e do pós-teste, desenvolvido no Google Forms
Fonte: autores

Com a participação de 142 pessoas, obtivemos uma média de 3,15 acertos para as cinco questões do pré-teste, enquanto que no pós-teste obtivemos uma média de 4,47 acertos para as mesmas cinco questões, mostrando que a oficina contribuiu em aproximadamente 42% para a melhoria da compreensão dos estudantes, influenciando de forma positiva no aprendizado desses estudantes. Utilizar o mesmo teste permite que, ao final da oficina, o aluno possa se autocorrigir, analisando com mais propriedade as respostas que forneceu anteriormente no pré-teste, uma vez que durante a oficina ele explorou e aprendeu sobre o conteúdo ali abordado. A comparação individual da quantidade de acertos para cada questão é apresentada na Figura 8.

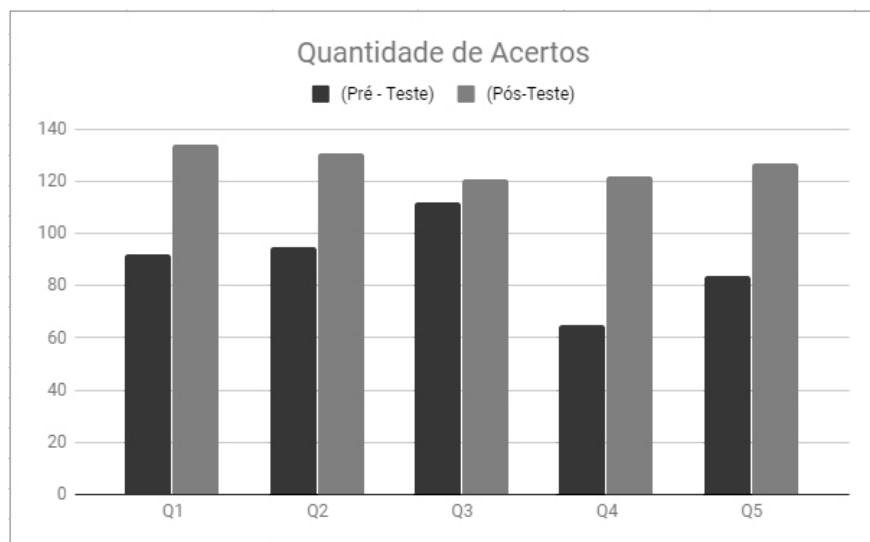


Figura 8 – Comparação de acertos entre cada questão do pré e do pós-teste
Fonte: autores

Ao final do pós-teste, um campo descritivo opcional para sugestões, críticas e demais contribuições é disponibilizado para cada um dos participantes, abrindo assim um espaço para que eles compartilhem suas respectivas percepções e experiências: todos os relatos que recebemos até o presente momento contribuíram de forma positiva para os nossos resultados, já que os estudantes demonstraram bastante interesse nas atividades, inclusive demons-

trando a intenção de retornar para a instituição no futuro. Alguns desses relatos podem ser visualizados na Figura 9.

Ótima oficina, aprendemos o básico de programação e adorei!
Acho muito legal e pretendo voltar aqui ano q vêm
Tudo ótimo, obrigado pela demonstração
Adorei a aula!!
Divertido!
Legal!
O teste é bem divertido, além de forçar a mente para descobrir as charadas.
EU GOSTEI DA OFICINA
Adorei a oficina! Organizada, objetiva e clara nas explicações. Muito obrigada pela oportunidade!
GOSTEI MUITO SERIA MUITO BOM APRENDER MAIS

Figura 9 – Alguns dos retornos recebidos após a realização das oficinas

Fonte: autores – dados da pesquisa

É importante mencionar que, dos 142 participantes das oficinas, 36 se inscreveram para a realização do processo seletivo para o ano de 2019 do IFRS Campus Osório, que aconteceu em dezembro de 2018. Esse resultado superou nossas expectativas, surpreendendo-nos de forma muito satisfatória, pois isso significa que de cada quatro alunos participantes das oficinas, um deles demonstrou interesse em ingressar na nossa instituição.

Dificuldades

Foram duas as principais dificuldades encontradas durante a execução do projeto: a locomoção dos estudantes e a indisponibilidade de um laboratório de informática em algumas das escolas que

participaram do projeto. Assim, duas escolas agendaram as oficinas conosco, e não puderam comparecer, pois não havia como realizar o transporte dos alunos até nossa instituição. Em consequência disso, oferecemos a realização da oficina na escola. Em uma delas, não havia laboratório de informática nem acesso à Internet, e então nos comprometemos a investigar formas de realizar a oficina off-line no próximo ano. Diante disso, estamos estudando alternativas para oferecer as oficinas de uma forma diferenciada, sem acesso à Internet e sem a necessidade do uso do computador, como forma de trabalho futuro, para complementar a oferta das nossas oficinas para o ano de 2019.

Participação em Eventos

Até o momento, nosso trabalho já foi divulgado em oito eventos, conforme apresentamos na Tabela 3. A participação em eventos tem sido muito positiva e estimulante para todos os participantes do projeto, em que podemos trocar e compartilhar ideias, conhecendo outros olhares, e aprendendo sobre os mais diversos recursos que podemos incluir em nossas oficinas, recebendo muitas vezes o feedback de pessoas mais experientes. Com a participação nesses eventos, estamos tendo a oportunidade de aprimorar ainda mais o projeto, enriquecendo nossas experiências e apresentando para a sociedade a forma que encontramos para contribuir e trabalhar em busca de uma educação de qualidade e de um futuro melhor para todos em nossa comunidade.

CONCLUSÕES E TRABALHOS FUTUROS

O trabalho apresentado refere-se a um projeto de extensão no IFRS Campus Osório, realizado com o objetivo de desenvolver o raciocínio lógico e o pensamento computacional em alunos das séries finais do ensino fundamental. São desenvolvidas atividades de forma lúdica, por meio da programação básica, com o auxílio de jogos digitais disponíveis na plataforma Code.org. Dessa forma, pretendemos também contribuir para uma formação integral aos estudantes envolvidos, oferecendo novas oportunidades de aprendizado, e enriquecendo os conteúdos abordados pela BNCC.

Com a implementação desse projeto, espera-se que os alunos participantes possam consolidar o seu respectivo aprendizado a partir de atividades diferenciadas e criativas. Além disso, envolvemos a comunidade do litoral-norte gaúcho em nossas atividades, e também divulgamos a nossa instituição de ensino para esses alunos, a fim de que eles possam optar futuramente por cursar o ensino médio técnico conosco, de forma gratuita.

Atualmente, nossa proposta já atendeu 142 participantes, de nove escolas distintas. Com base nos resultados já obtidos, conseguimos uma melhoria de 3,15 para 4,47 na quantidade de acertos dos testes realizados sobre os conteúdos trabalhados, representando uma melhoria de aproximadamente 42% na compreensão dos enunciados e na resolução das questões propostas após a realização das oficinas.

Feira/Mostra/Evento	Situação
XII MOCITEC - Mostra de Ciências e Tecnologias IFSul Câmpus Charqueadas.	Apresentação de pôster em 24 e 25 de agosto de 2018, em Charqueadas/RS.
VI IFCITEC - Feira de Ciências e Inovação Tecnológica do IFRS - Campus Canoas.	Apresentação de pôster em 19 de setembro de 2018, recebendo a premiação de segundo lugar na categoria Informática, em Canoas/RS.

8ª MoExP – Mostra de Ensino, Extensão e Pesquisa do IFRS – Campus Osório.	Apresentação de pôster em 25 de setembro de 2018, em Osório/RS.
IX SULCOMP – Congresso Sul Brasileiro de Computação.	Apresentação oral e pôster, entre os dias 3 e 6 de outubro de 2018, em Criciúma/SC.
SaberTec 2018 – IFSul Sapucaia do Sul.	Apresentação de pôster no dia 10 de outubro de 2018, em Sapucaia do Sul/RS, recebendo destaque e credenciamento para a Feira Brasileira de Iniciação Científica (Febic) de 2019.
19ª Mostra POA – Mostra de Ensino Pesquisa e Extensão do IFRS – Campus Porto Alegre.	Trabalho apresentado na modalidade oral, no dia 24 de outubro de 2018, em Porto Alegre/RS.
7ª Mostra Técnica do IFRS – Campus Feliz.	Trabalho apresentado na modalidade de pôster, entre o período de 25 e 27 de outubro de 2018, em Feliz/RS.
3º Salão de Pesquisa, Extensão e Ensino do IFRS e 6º Seminário de Extensão (SEMEX).	Trabalho apresentado de forma oral, na Sessão de Indissociabilidade, no dia 25 de novembro de 2018, em Bento Gonçalves/RS.

Quadro 1 – Participação em Eventos, Feiras e Mostras

Fonte: autores

Além de todas as atividades e testes que propomos nas oficinas, também temos a pretensão de trabalhar com mais uma plataforma de ensino de programação: o Scratch (2018), que pode ser utilizada até mesmo sem conexão à Internet. Nela, os alunos poderão desenvolver o próprio jogo e também jogar os jogos desenvolvidos por outros alunos, elevando ainda mais o nível de conhecimento de lógica de programação. Diferentemente da plataforma utilizada nesse projeto até o momento, a plataforma Scratch (2018) faz o aluno se envolver diretamente na produção dos códigos utilizados para o desenvolvimento dos jogos, e não apenas jogá-los. Outra alternativa que estamos analisando para ser implementada no ano de 2019 é a impressão dos comandos e dos cenários dos jogos, para que esses conteúdos possam ser trabalhados de maneira off-line, a fim de que

possamos contemplar e beneficiar mais escolas com o projeto, sem a necessidade do uso de computadores.

REFERÊNCIAS

Andrade, R., Mendonça, J., Oliveira, W., Araujo, A. L., & Souza F. (2016). *Uma Proposta de Oficina de Desenvolvimento de Jogos Digitais para Ensino de Programação*. Congresso Brasileiro de Informática na Educação (CBIE).

Alvarez, L. (2014, nov.). Ensino de programação é aposta de colégios em todo o mundo. *Revista Educação*.

Barbosa, D. N. F., & Bassani, P. B. (2013). Em direção a uma aprendizagem mais lúdica, significativa e participativa: experiências com o uso de jogos educacionais, tecnologias móveis e comunidade virtual com sujeitos em tratamento oncológico. *Renote – Revista Novas Tecnologias na Educação*, Porto Alegre – RS, Brasil, v. 11, p. 1-10.

Base Nacional Comum Curricular – BNCC (2019). Recuperado em jan. 2019, de http://basenacionalcomum.mec.gov.br/wp-content/uploads/2018/12/BNCC_19dez2018_site.pdf.

Cardoso, W. M., & Barbosa, D. N. F. (2016). Programando Jogos educacionais para auxiliar na aprendizagem de crianças e adolescentes em tratamento oncológico. *Anais da Feira de Iniciação Científica*. Novo Hamburgo: Feevale.

Cavalcante, A., Costa, L. S., & Araujo, A. L. (2016). *Um Estudo de Caso Sobre Competências do Pensamento Computacional Desenvolvidas na Programação em Blocos no Code.Org*. Congresso Brasileiro de Informática na Educação (CBIE).

Code, 2018. *Plataforma Code.org*. Recuperado em set. 2018, de <http://www.code.org>.

Dantas, R. F., & Costa, F. E. A. (2013). CODE: O ensino de linguagens de programação educativas como ferramentas de ensino/aprendizagem. *Simpósio Hipertexto e Tecnologias na Educação*, v. 5, Recife.

França, R., & Tedesco, P. (2015). *Desafios e oportunidades ao ensino do pensamento computacional na educação básica no Brasil*. Congresso Brasileiro de Informática na Educação (CBIE).

Gomes, T., Barreto, P., Lima, I. R. A., & Falcão T. P. (2015). *Avaliação de um Jogo Educativo para o Desenvolvimento do Pensamento Computacional na Educação Infantil*. Congresso Brasileiro de Informática na Educação (CBIE).

Google Forms (2018). *Formulários do Google*. Recuperado em set. 2018, de <https://www.google.com/forms/about/>.

Ideb (2018). *Índice de Desenvolvimento da Educação Básica*. Recuperado em set. 2018, de <http://portal.inep.gov.br/ideb>.

Kologeski, A., Barbosa, D. N. F., Miorelli, S. T., & Grings, C. (2016). Logi-cando: Ensinando Lógica com as Tecnologias da Informação. XV *Seminário Internacional de Educação*, Novo Hamburgo. XV Seminário Internacional de Educação - Educação e Interdisciplinaridade: Percursos Teóricos e Metodológicos.

PPC (2019). *Projeto Político-Pedagógico do Curso Técnico de Informática Integrado ao Ensino Médio do IFRS Campus Osório*. Recuperado em jan. 2019, de [https://osorio.ifrs.edu.br/site/midias/arquivos/20162420155561ppc_emi_info_atual_\(1\).pdf](https://osorio.ifrs.edu.br/site/midias/arquivos/20162420155561ppc_emi_info_atual_(1).pdf).

Schoeffel P., Moser P., Varela G., Durigon L., de Albuquerque G. C., & Nique-latti M. (2015). *Uma Experiência no Ensino de Pensamento Computacional para Alunos do Ensino Fundamental*. Congresso Brasileiro de Informática na Educação (CBIE).

Scratch (2018). *Plataforma Scratch*. Recuperado em set. 2018, de www.scratch.mit.edu.

MIT (2018). *Plataforma MIT App Inventor*. Recuperado em set. 2018, de www.appinventor.mit.edu.

Winter, N. J., Santos, G. N., Strack, T. L., Mossmann, J.B., Barbosa, D. N. F., & Bez, M. (2014). Incentivo ao Estudo Através dos Jogos: Experiências no Desenvolvimento de uma Rede Social Gamificada. *Revista Hipertexto*, v. 4, p. 1-20.

SEÇÃO 3

RELATOS DE EXPERIÊNCIAS INOVADORAS

Nesta seção, serão apresentados, ao todo, dez Relatos de Experiências Inovadoras, que representam as mais de 80 escolas e 200 professores participantes do programa Letramento em Programação do Instituto Ayrton Senna, na edição 2018, que contou com a participação dos núcleos do interior de São Paulo – SP, Norte Gaúcho – RS, Caruaru – PE e Manaus – AM.

O primeiro relato tem o título “Letramento Digital Norte Gaúcho: Uma Possibilidade Para o Desenvolvimento de Habilidades Sociocomportamentais em Alunos Com TDAH e Comorbidades”, de autoria de Irani Bernadete Roani e Ivânia Campigotto Aquino, de Passo Fundo/RS, e tem como objetivo apresentar a experiência vivida durante o desenvolvimento do Projeto Letramento em Programação com alunos, pertencentes a diferentes escolas da rede municipal de educação, diagnosticados com distúrbios sociocomportamentais, TDAH e comorbidades.

O segundo relato, intitulado “Emef Antonio Zambrzycki de Getúlio Vargas abrindo as portas para o Letramento em Programação”, de Vanessa Polli, do município de Getúlio Vargas/RS, cita que em um ano em que o Brasil todo está discutindo a nova Base Nacional Comum Curricular (BNCC), inteiramente voltada à introdução das tecnologias digitais, inclusive no que se refere aos gêneros textuais, esse projeto representou uma oportunidade de ensaiar práticas até então apenas debatidas no papel, colocando os participantes como protagonistas e criadores diante da máquina, pois o interesse e a empolgação em participar só aumentaram.

Nosso terceiro relato tem o título “Relato da Experiência de Utilização da Ferramenta Scratch e APP Inventor no Processo de Ensino-Aprendizagem de Alunos do Município de Marau/RS”, de autoria de Diego Fernando Grzibowski, Ramon Fioravanso, Simone Costenaro Ribeiro e Maikyeli Orsato Decesaro, e abrange as experiências didáticas e metodológicas promovido pela Secretaria Municipal de Educação por meio do Núcleo Tecnológico Educacional Municipal (NTM) no fomento e promoção do Programa Letramento em Programação dos alunos da rede pública municipal de ensino dos 6^a, 7^a e 8^a anos durante os anos de 2017 e 2018.

O quarto relato, intitulado “A cultura maker inserida no letramento em programação”, de Peter Rodrigo Trento, ocorreu em

Vinhedo/SP, e apresenta o trabalho realizado pelos alunos de 9º ano da escola municipal Darci A. Briski, durante o contraturno escolar, os quais realizaram um projeto maker durante as aulas de programação, e o resultado dessas ações foi premiado com o Tech Oscar, além de proporcionar um grande engajamento dos alunos de programação e que levaremos para os próximos anos com esse projeto na rede municipal.

Nosso quinto relato tem o título “Realizando a Atividade da Construção do Jogo de Labirinto”, de autoria de Érica Luiza de Oliveira, de Tapejara/RS, e cita que os alunos se divertiram ao perceber o quanto podem aprender e estudar não somente da forma tradicionalmente imposta e sim de uma maneira lúdica. Também cita que, quanto ao desempenho nas disciplinas do componente curricular do sexto ano, notou-se nítida melhora das notas dos estudantes, que mostraram um envolvimento maior com as aulas e, de uma maneira especial, fizeram com que a escola inteira participasse e se envolvesse no programa.

Já o sexto relato, também de Vinhedo/SP, tem o título “As distintas experiências docentes que favorecem as práticas pedagógicas”, de autoria de Eduardo Mauricio Moreno Pinto, e apresenta a experiência de alunos dos 9º anos da escola E.M. Prof.^a Darci Ana Dêgelo Briski, que tinha por objetivo experimentar a implementação das propostas do programa Micro:bit, e destaca alguns momentos vividos durante o primeiro semestre de 2018.

O sétimo relato vem de São Roque, zona rural de Passo Fundo/RS, e tem o título “Um Olhar Sobre as Construções – Analisando Interações e Aprendizagens”, de autoria de Sonia Mara Martins de Mattos Souza, tendo como objetivo descrever a experiência realizada em ambiente informatizado, no contexto do projeto Letramento em Programação, com os alunos de uma escola do campo. A experiência ocorreu envolvendo nove alunos de uma turma multisseriada do 4º e 5º anos.

O oitavo relato é da cidade de Manaus/AM, com o título “Trololó: Que barulho é esse?”, de Daniel Pereira de Souza, que cita que ao inserir atividades, utilizando o Scratch e outras ferramentas, para trabalhar a linguagem de programação por blocos, foi percebido o quanto os alunos se interessam pelas atividades e conhecimentos

propostos, e que podem estar relacionadas muito além de uma temática específica, mas podem contribuir para disseminação do pensamento computacional e ser utilizado como recurso para trabalhar os direitos de aprendizagem dos alunos e os conteúdos distribuídos pelo currículo escolar.

O nono relato, de Itatiba/SP, tem o título “Tecnologia, desenvolvimento cognitivo e autoconhecimento caminhando juntos, é possível?”, com autoria de Eliana Maria Fattori Calza, e teve por objetivo apresentar a atividade desenvolvida no segundo semestre do ano de 2018, intitulada “Eu como comunicador”, com 17 alunos de 4º ano da Emeb Prof. Luiz Pântano, que tiveram o desafio de criar o seu próprio projeto, no Scratch, sob orientação de uma temática “cyberbullying”, um assunto tão atual e relevante dentro e fora da escola.

Por fim, o décimo relato vem de Caruaru/PE, com o título “A desafiante e encantadora missão chamada letramento em programação”, de Nariana de Oliveira Tavares, e apresenta como se deu o processo de formação de professores e a implantação do programa Letramento em Programação em uma das escolas de tempo integral, que desafiou e encantou os professores envolvidos, e despertou nos estudantes curiosidade, motivação e engajamento.

Todos os dez relatos são uma pequena, mas rica amostra de como o Letramento em Programação do Instituto Ayrton Senna está impactando centenas de escolas e de professores, e proporcionando a milhares de estudantes de escolas públicas brasileiras uma oportunidade real de transformação pela Educação Integral, por meio do desenvolvimento do pensamento computacional, integrado à BNCC e respeitando a realidade e relevância de cada localidade onde atua.

CAPÍTULO 9

LETRAMENTO DIGITAL NORTE GAÚCHO: UMA POSSIBILIDADE PARA O DESENVOLVIMENTO DE HABILIDADES SOCIOCOMPORTAMENTAIS EM ALUNOS COM TDAH E COMORBIDADES

*Irani Bernadete Roani
Ivânia Campigotto Aquino*

INTRODUÇÃO

Apresentamos aqui um breve relato da experiência vivida durante o desenvolvimento do Projeto Letramento em Programação com alunos, pertencentes a diferentes escolas da rede municipal de educação do município de Passo Fundo/RS, diagnosticados com distúrbios sociocomportamentais, TDAH e comorbidades. Até 2017, o acompanhamento desses estudantes era semanal, por profissionais especialistas no Centro Municipal de Atendimento ao Educando (Cemae), núcleo responsável pelo atendimento a educandos da rede, com equipe multidisciplinar, vinculado à Secretaria Municipal de Educação. Incluiu-se, em maio de 2017, o projeto Letramento em Programação como mais uma atividade, complementando as demais, para esses alunos, oportunizando o contato com a programação de computadores. Neste relato, também está citado o caso de um dos meninos portador de TDAH, que concluiu com sucesso os módulos I e II. Simultaneamente, dominou a programação e tornou-se um grande leitor, inspirando seus colegas de escola para a leitura.

DESCRIÇÃO DA EXPERIÊNCIA

O projeto Letramento em Programação Digital, inserido como complemento das atividades para esse grupo de alunos, teve continuidade em 2018. Aconteceu com encontros semanais, na sala de multimídias na Biblioteca Pública Municipal Arno Viuniski, em Passo Fundo/RS. Utilizam-se os computadores com sistema operacional Linux Educacional 4.0. Embora o sistema seja adequado, enfrentam-se dificuldades técnicas, como o acesso à internet, pois, mesmo havendo banda larga, as máquinas são defasadas e o sistema operacional não suporta muitas solicitações simultâneas de acesso. Alguns alunos conseguem trabalhar *online* o Scratch (<https://scratch.mit.edu>), onde eles criam uma conta com usuário e senha, enquanto os outros trabalham *offline*, com o software instalado. As tarefas realizadas são salvas no computador e, depois, passadas para o pendrive. Em casa, a professora acessa o ambiente dos alunos e posta a atividade realizada no encontro.

O referido projeto, em 2017, iniciou com um grupo composto por alunos do 3º ao 6º ano, diagnosticados com distúrbios socio-comportamentais, TDAH e comorbidades. Ao envolver esse grupo específico, tinha-se como principal objetivo, justificado pelas queixas vindas da escola e das famílias, a melhoria na qualidade das relações interpessoais dos alunos no ambiente escolar, resgatando a autoestima e diminuindo sintomas de oposição à aprendizagem formal.

No decorrer da proposta, com a observação e registros constantes na evolução dos estudantes, foram confirmadas algumas habilidades demonstradas por eles, em desacordo com a padronização que a escola faz em relação à forma de ensinar, justificando a oposição à aprendizagem formal.

As ações seguem um roteiro com atividades desplugadas e digitais. Todas as tentativas, erros e acertos são registrados por eles em um documento, podendo ser digital ou físico, o qual chamamos Diário de Bordo. Ao final de cada encontro, faz-se um bate-papo para avaliar como foi o desenvolvimento das tarefas, se acertaram ou não, se conseguiram realizar sozinhos, se precisaram de ajuda, quem ajudou, se a professora interferiu ou se entre eles houve cooperação.

Nos primeiros encontros, quando o assunto ainda era desconhecido, a participação era mínima. Com a evolução do trabalho, a percepção da capacidade em desenvolver as atividades propostas *on* e *offline* cresceu muito, tanto na participação quanto na troca de informações. O auxílio aos colegas com dificuldades já iniciou o ciclo de esclarecimento em relação aos distúrbios sociocomportamentais. Percebeu-se, naquele momento, uma contrariedade no discurso da escola, pois esses meninos não pertenciam a uma escola só, eram de diferentes comunidades e estruturas familiares, estranhos entre si. Concorde-se que há um diferencial no projeto, o interesse dos que permanecem é comum, estão fazendo algo de que gostam e que lhes mostrou capacidades não valorizadas em outros espaços.

Então, cabe aqui uma reflexão acerca da questão metodológica. É urgente que se mudem metodologias, pois a era industrial ainda permanece nos padrões da escola. Acredita-se nesse projeto também como incentivo à busca por novas metodologias. Considere-se aqui a metodologia ativa de aprendizagem baseada em pares. Não na essência teórica do *Peer Instruction*, mas pela colaboração e cooperação entre os pares. Todos com diagnóstico, relatos da escola por dificuldades da e na aprendizagem, pelos distúrbios sociocomportamentais, queixas das famílias em relação a comportamento e aprendizagem. Ao longo do ano, pode-se perceber algumas evoluções positivas em relação aos pareceres iniciais. Acontecia entre eles a colaboração, os ajustes nos trabalhos nos quais se auxiliavam, sendo a professora chamada só em último caso.

Um desses meninos, diagnosticado com TDAH, tornou-se um grande leitor. Como o projeto acontece na biblioteca pública, e percebendo-se que o aluno passou a explorar os ambientes da biblioteca antes do horário do projeto, sugeriu-se que ele fizesse um cadastro como leitor. A partir de então, ele passou a fazer a retirada semanal de livros. A devolutiva da escola em relação a esse fato foi bastante interessante. Seus professores relataram que, durante o horário do recreio, ele afastava-se do grupo para ler. Os colegas passaram a observá-lo, e, curiosos, sentiram-se incentivados a fazer o mesmo. Em sala de aula, o estudante repetia o processo ao concluir suas atividades, retirava seu livro da mochila e dava sequência na leitura, enquanto aguardava seus colegas terminarem para avançar às próximas tarefas.

Sua mãe também relatou a melhora nas notas, na socialização com os amigos e com os familiares. Segundo ela, seu filho tinha “ataques de asma”, talvez como consequência da dificuldade na fala, causando ansiedade, inquietação. Aguarda avaliação e atendimento fonoaudiológico em função da dicção e articulação. No final de 2017, avaliando esse aluno em sua participação e evolução durante o módulo I, percebeu-se um avanço no sentido geral: notas, socialização, comportamento e até na saúde. Sua fala também melhorou, embora não tenha tido ainda a avaliação fonoaudiológica, mas acredita-se que, pelos livros lidos, algum processo tenha se desencadeado, facilitando-lhe mostrar sua voz, falar em público, embora ainda com certa dificuldade de se fazer entender, deixando de lado a ansiedade que o atrapalhava. E o mais importante, as crises de asma praticamente desapareceram.

RESULTADOS E REFLEXÕES

O estudante passou a ser uma referência no grupo, pois, ao frequentar o projeto, tornou-se um leitor e inspirou alguns colegas para a leitura. Mostra conhecimentos avançados na programação, os desafios são resolvidos muito além do que é proposto. Faz curso de inglês e quer estudar japonês também. Neste link, o projeto final de 2018 do referido aluno: <https://scratch.mit.edu/projects/245424096/>.

Durante os encontros do projeto, ele auxilia os colegas do módulo I. Continua também a retirar livros e praticar a leitura. Esse fato incentiva todos os envolvidos no projeto, na exigência para que melhorem ainda mais, pois uma simples proposta de ensinar programação de computadores, sobre a qual não se tinha a mínima ideia de como seria a aceitação dos meninos TDAH, em dois anos de atividades, mostra resultados bastante positivos.

Segundo Vygotsky (1987, p. 101), “o aprendizado adequadamente organizado resulta em desenvolvimento mental e põe em

movimento vários processos de desenvolvimento que, de outra forma, seriam impossíveis de acontecer.” O uso de tecnologias ainda não estava na escola quando apresentou suas teorias socio-construtivistas e zona de desenvolvimento proximal e real (ZDP/ZDR), mas já previa o futuro, nesse caso vemos a ZDP na prática.

Nesses dois últimos anos, aprendemos muito sobre ensino da programação. Perceber a passividade no início, e depois o progresso no pensamento computacional vivo nas produções, tem um sentido muito positivo e motivador para todos os envolvidos, alunos, professores, escola e família. Resnick (2017) diz que é preciso “piso baixo e teto alto” e, citando Papert, conclui, “e paredes amplas”. Dois gênios da tecnologia, mesmo não sendo educadores, perceberam as limitações que vivemos e dificuldades enfrentadas no ‘chão da escola’. O uso da tecnologia não resolve os problemas da/na aprendizagem, mas pode atuar como um colaborador/mediador para que alguns mitos se desfaçam e ela aconteça.

No processo, houve a necessidade de fundamentar alguns comportamentos demonstrados pelos alunos e por suas escolas. O comprometimento das famílias no acompanhamento de seus filhos também merece ser destacado, tendo em vista o rendimento deles estar relacionado diretamente ao incentivo dos familiares. Mais uma vez comprovado que as metodologias utilizadas no processo de ensino e de aprendizagem devem ser revistas com frequência, tendo em vista a educação integral.

REFERÊNCIAS

- Resnick, M. (2017). *Palestra 'Plantando as sementes para uma sociedade mais criativa'*. Conferência Scratch Brasil, USP, São Paulo, 5 de outubro de 2017.
- Vygotsky, L. S. (1987). *Pensamento e Linguagem*. São Paulo, Martins Fontes.

CAPÍTULO 10

EMEF ANTONIO ZAMBRZYCKI DE GETÚLIO VARGAS ABRINDO AS PORTAS PARA O LETRAMENTO EM PROGRAMAÇÃO

Vanessa Polli

INTRODUÇÃO

Participar do Projeto Letramento em Programação foi a experiência mais inovadora e, ao mesmo tempo, desafiadora de 2018. Em um ano em que o Brasil todo está discutindo a nova Base Nacional Comum Curricular (BNCC), inteiramente voltada à introdução das tecnologias digitais, inclusive no que se refere aos gêneros textuais, esse projeto representou uma oportunidade de ensaiar práticas até então apenas debatidas no papel.

A maioria dos participantes do Projeto utilizava a tecnologia para fins comuns, como: interagir nas redes sociais, jogar, pesquisar e comunicar-se. Quando perceberam que o objetivo do Letramento em Programação ultrapassava a mera recepção, colocando-os como protagonistas e criadores diante da máquina, o interesse e a empolgação em participar só aumentaram.

Encerramos o ano de 2018 com 100% dos participantes ativos e sedentos pela continuação do Projeto. No mês de novembro, realizamos a “I Mostra de Saberes e Fazeres da EMEF Antonio Zambrozkycki”, em que, na presença de toda comunidade escolar, pudemos apresentar os objetivos e funcionamento do Projeto, bem como sua importância para o cidadão do século XXI. Na oportunidade, a garotada, pais e corpo docente, puderam interagir com os projetos desenvolvidos no Scratch (histórias interativas e jogos), e todo trabalho, até então no anonimato, foi enfim divulgado.

DESCRIÇÃO DA EXPERIÊNCIA

A experiência aqui reproduzida deu-se no município de Getúlio Vargas, região norte do estado do Rio Grande do Sul. A escola participante, que oferta ensino fundamental completo, leva o nome Antonio Zambrzycki e está situada no bairro Nossa Senhora Consoladora. Além de ser um espaço muito desejado por pais e estudantes, a escola é ponto de encontro e referência em educação.

A escola ingressou no projeto no ano de 2018, por meio da adesão da Secretaria Municipal de Educação, Cultura e Desporto (SMECD) e a consequente anuência da equipe gestora escolar. Esse projeto foi uma oportunidade ímpar de possibilitar aos estudantes o contato com as ferramentas digitais, tão fundamentais para a construção do ser humano do século XXI. O desejo de toda comunidade é que essa semente ora plantada e germinada continue a crescer e produzir frutos.

Os protagonistas dessa experiência são 15 estudantes do 6º ano do ensino fundamental, na faixa etária de 11 a 12 anos, com características muito diversas entre si, mas semelhantes em um quesito: na vontade de “dominar” a máquina, ir além da receptividade passiva de conteúdos. Contudo foram muitas as vezes em que, curiosos pelas atividades do contraturno dos colegas, estudantes alheios ao projeto ficaram à porta do laboratório interessados em participar.

Antes de dar início às atividades, os pais foram convidados a participar de uma reunião – nas dependências do laboratório de informática – cujo objetivo era explicar o funcionamento do Projeto e sua proposta, bem como obter o consentimento para a participação de seus filhos. A alegria e o sentimento de satisfação foi aparente nos olhos de cada um deles.

O professor David Zanoni, também orientador pedagógico da SMECD e conhecedor da área de informática, é quem coordenada e dá suporte ao projeto no município. Já a professora que ministrou as aulas, e agora revira a memória e seus guardados para descrever essa experiência, possui formação na área de Letras e, por isso mesmo, sentiu-se movida pelo novo desafio: aprender, experimen-

tar e pôr em prática as atividades antes de propor aos estudantes. Não possuir formação na área computacional não representou uma barreira ou um empecilho, ao contrário, reforçou ainda mais o seu desejo de crescer e evoluir pessoal e profissionalmente.

As aulas do Letramento aconteciam no contraturno escolar, em dia da semana e horário mais convenientes para estudantes e professora, no laboratório de informática da própria escola. De posse do material repassado pelo Instituto Ayrton Senna, por intermédio do *classroom*, desenvolvido pela Imed de Passo Fundo, imprimia as aulas e formava um *portfólio*, onde fazia meus apontamentos pessoais. Em casa, em computador próprio e com uma internet mais privilegiada, desenvolvia as atividades propostas e, ao mesmo tempo, controlava o tempo aproximado de cada atividade, bem como se havia a necessidade de executar tudo aquilo que estava proposto no material, sempre observando as peculiaridades de cada sujeito e suas necessidades.

A execução das atividades plugadas dava-se comumente com os estudantes dispostos em duplas ou trios nas máquinas. Isso porque os computadores de que a escola dispõe já estão bastante obsoletos e, como se não bastasse, a internet disponível não é das mais velozes. Não foram raros os casos em que, no meio de uma atividade e no auge da empolgação, a máquina simplesmente apagou. Ao reler os diários de bordo dos estudantes e retomar os relatos dos pais, sem dúvida, estes são os entraves mais citados: falta de qualidade no equipamento e lentidão da internet.

Porém é, sobretudo, em meio à adversidade que a criatividade ganha espaço. Nas aulas em que não havia acesso à internet, era possível trabalhar várias atividades desplugadas do Módulo I que, sem dúvida, contribuíram sobremaneira para desenvolver habilidades necessárias para o letramento digital, como por exemplo, atividades que envolviam raciocínio lógico.

Nas atividades desplugadas, os participantes organizavam-se em círculo, o que favorecia a interação olho no olho, a organização dos turnos de fala, incentivo ao respeito e à cooperação mútua.

RESULTADOS E REFLEXÕES

Abraçar o Projeto Letramento em Programação envolve, no mínimo, a transformação direta de três pilares: estudantes, professor participante e escola. Quanto aos estudantes, o ganho pessoal é imenso, pois eles desenvolvem o relacionamento interpessoal e a cooperação por meio do trabalho em equipe; o raciocínio lógico e a criatividade, tendo em vista o sem número de situações que terão de resolver no principal programa utilizado nesse módulo (Scratch). Além disso, colocam em prática muitos conhecimentos adquiridos em sala de aula ao programar: o cuidado com a oralidade e expressividade ao narrar uma história no Scratch, a escrita correta das palavras em todas as ações que demandem o uso da língua materna: nomear, descrever e produzir as instruções de um projeto.

Relendo os depoimentos dos estudantes, é possível confirmar todos esses benefícios:

O que eu mais gostei no Scratch é que nós mesmos criamos o movimento dos personagens, e é de graça. Pensei em criar uma história.

A animação interativa é uma animação que podemos controlar. Hoje eu evolui no quesito atenção. (Pedro)

Desafiamos o nosso raciocínio com o desafio desplugado, que se trata de matemática.

Hoje aprendi com a minha amiga, ela que me ajudou. (Martina)

Hoje fizemos um vídeo coletivo sobre a importância do Projeto. <https://www.youtube.com/watch?v=oQw-5VgaH6el&feature=youtu.be>. Para mim, como aluna do Letramento, é fantástico, pois fui uma sortuda de ganhar essa oportunidade. A tecnologia vamos usar muito no futuro, por isso devemos saber cada vez mais. E sempre nos dedicar nisso, para ser orgulho da família e ser destaque no Letramento. (Marina)

Auxiliei as pessoas a olharem as coisas compartilhadas e usarem a mochila. (Eloisa)



Figura 1 – Produção de história interativa no Scratch

Fonte: autores



Figura 2 – Ajudando o colega na movimentação de ator no palco

Fonte: autores

Outro ponto que merece destaque é um dos combinados sugeridos no material das aulas: “pergunte a três antes de mim (o professor)”. Esse procedimento favorece e consolida ainda mais a aprendizagem, pois é sabido que ensinar os outros é uma das melhores formas de aprender. Assim, aqueles estudantes que têm mais facilidade na manuseio do programa sentem-se valorizados ao assumir uma função análoga à do professor. Com o passar do tempo, a tendência é que se trabalhe mais com perguntas do que com respostas, visto que o Projeto também permite desenvolver a autonomia do sujeito programador. A construção dessa autonomia pode também ser atribuída a outro fator: à interface intuitiva e simples do Scratch.

No que se refere ao professor participante, integrar-se no Letramento em Programação só tende a enriquecer o seu trabalho, pois, ao passo que se apropria das tecnologias nele propostas, começa a vislumbrar ações efetivas de inserção dessas práticas em sua sala de aula, isto é, olha para uma coisa e vê outra. Sem dúvida, inserir o uso das tecnologias em sala de aula é uma demanda crescente e torná-la uma aliada no ensino-aprendizagem é algo que todo professor que se importa com sua formação, com a inovação e com a motivação dos estudantes precisa buscar.

O Letramento, já em seu primeiro módulo, oferece muitas ferramentas para esse fim. Por que não inserir jogos na aprendizagem e aprender de forma divertida? Que tal utilizar o Scratch para recriar um livro lido nas aulas de Língua Portuguesa? O que você acha de desenvolver, na prática e de forma interativa, conteúdos da aula de Artes, como a arte abstrata? Outra ferramenta que também pode ser utilizada na escola é o *classroom*, plataforma que permite a comunicação entre professores e alunos. Nele, é possível lançar tarefas para que os estudantes interajam e, com isso, renovar metodologia e avaliação.

De fato, no princípio, havia muita incerteza acerca da proposta e dos objetivos do Projeto, mas, à medida que os encontros de formação aconteciam, tudo ficava cada vez menos nebuloso e mais instigante. Um importante papel do professor é também disseminar na escola as práticas desenvolvidas no Letramento e instigar outros professores a fazer parte dele. Dessa forma, todos saem ganhando.

Outrossim, o letramento digital oferece aos estudantes a oportunidade de concluir o ensino fundamental com uma formação mínima adicional, o que poderá facilitar a tomada futura de decisões em relação à profissão, já que, cada vez mais, a tecnologia está presente na vida do homem moderno. Assim, a tendência é que aquele que souber “dominar a máquina”, ao invés de ser “dominado”, tenha maiores chances de alcançar o sucesso e tornar-se um cidadão mais pleno.

CAPÍTULO 11

RELATO DA EXPERIÊNCIA DE UTILIZAÇÃO DA FERRAMENTA SCRATCH E APP INVENTOR NO PROCESSO DE ENSINO- APRENDIZAGEM DE ALUNOS DO MUNICÍPIO DE MARAU/RS

Diego Fernando Grzibowski

Ramon Fioravanso

Simone Costenaro Ribeiro

Maikyeli Orsato Decesaro

INTRODUÇÃO

O presente relato compreende e abrange as experiências didáticas e metodológicas promovidas pela Secretaria Municipal de Educação de Marau/RS por meio do Núcleo Tecnológico Educacional Municipal (NTM) no fomento e promoção do Programa Letramento em Programação dos alunos da rede pública municipal de ensino dos 6º, 7º e 8º anos durante os anos de 2017 e 2018.

DESCRIÇÃO DA EXPERIÊNCIA

O município de Marau, situado no estado do Rio Grande do Sul, hoje contando com uma população estimada de 43.435 habitantes segundo o IBGE (2018) e possuindo cerca de 5.060 alunos matriculados na rede pública municipal entre educação infantil e ensino fundamental, foi, juntamente com os municípios de Passo Fundo e Sananduva, pioneiro no estado do Rio Grande do Sul a firmar no ano de 2016 parceria com o Instituto Ayrton Senna e com a Faculdade Imed e disponibilizar o curso de Letramento em Programação aos estudantes interessados.

Em uma reunião realizada em janeiro de 2017 na secretaria de educação do município entre a secretaria municipal de educação, coordenação pedagógica e coordenadora municipal do Projeto Letramento em Programação, a fim de fazer uma avaliação e um prospecto do desenvolvimento do curso no âmbito municipal para os anos seguintes, foram tomadas algumas decisões de ordem estrutural e pedagógica.

A primeira decisão de ordem estrutural foi oferecer o curso no turno inverso das aulas, concentrando as atividades no laboratório do Centro de Referência da Criança e do Adolescente Romilda Mistura – Projeto AABB Comunidade –, local no qual já são ministradas outras 42 oficinas em diferentes áreas, durante o turno inverso.

A decisão decorreu do entendimento de que o “projeto piloto” havia contribuído para a formação e desenvolvimento dos jovens marauenses, fato que pode ser observado não apenas em sala de aula pelos professores, como também pela primeira colocação no projeto final do Letramento em Programação, entre os três municípios participantes. Lembrando que foram 11 alunos da Escola Municipal de Ensino Fundamental 28 de Fevereiro que obtiveram no final do ano de 2016 esse mérito.

Como o laboratório do Centro de Referência da Criança e do Adolescente Romilda Mistura – Projeto AABB Comunidade – apresentava uma estrutura de melhor qualidade e maior quantidade de equipamentos e o município já disponibiliza o serviço de transporte escolar para que os alunos pudessem realizar atividades no turno inverso, o poder público municipal, ao se deparar com elementos favoráveis, considerou pertinente e necessário fomentar o curso. Assim, a partir do ano de 2017 foi estendido aos estudantes das demais escolas não contempladas no primeiro ano.

A decisão de expandir o Letramento exigia uma carga horária de que os professores não dispunham, em razão de sua docência em sala de aula, por isso, a segunda decisão mencionada, como sendo de ordem pedagógica, foi destinar os profissionais do Núcleo Tecnológico Educacional Municipal (NTM) como corpo de recursos humanos responsáveis pela condução das atividades do letramento em programação, para que o plano de expansão pudesse ser concretizado.

No ano de 2018, o município de Marau disponibilizou o turno da manhã e tarde das terças-feiras para atender os alunos do Módulo I, e disponibilizou o turno da manhã e tarde das quartas-feiras para atender os alunos dos Módulos II e III.

Ofertado sempre no contraturno ao do período de aula, as alterações realizadas permitiram uma democratização de acesso ao curso e o aumento do número de alunos atendidos. Contudo, apesar

dos esforços despendidos na ampliação da quantidade de alunos atendidos, ficou diagnosticada a permanência de um problema, a evasão no curso.

O problema da evasão se dá por alguns fatores já diagnosticados, como: grande oferta de outros cursos, transferência de alunos para outras cidades e também dificuldade com propostas curriculares do Letramento.

Diante dessa dificuldade, o município de Marau aproveitou o Tech Oscar Municipal para convidar os demais alunos, cerca de 200 que participavam de outros cursos e oficinas, na tarde do dia 6 de novembro de 2016, no Centro de Referência da Criança e do Adolescente Romilda Mistura – Projeto AABB Comunidade –, muitos dos quais compõem o público-alvo em 2019, para que pudessem conhecer, por meio dos trabalhos desenvolvidos pelos seus colegas de escola, um pouco mais do que é o letramento em programação na prática.

Percebe-se que, quando ocorre a divulgação do curso, muitos dos alunos interessados sentem-se atraídos pela possibilidade de criar jogos. Contudo a prática parece tolher-lhe a facilidade que imaginou, e muitos não perseveraram, mesmo que se busque um diálogo de incentivo para sua permanência. Consideramos que, apesar de indesejável, o elevado número de evasão ainda reflete um fator cultural, por isso, necessita-se de críticas ponderadas e comedidas, pois os jovens do século XXI, também conhecidos como geração Alpha pelo fato de terem desde o seu nascimento um aparato tecnológico muito grande, foram treinados para apenas consumir tecnologia e não a consumir e produzir, e isso pode estar substanciando e explicando pelo menos parte das desistências que ocorreram.



Figura 1 – Formatura alunos Módulo I, II e III

Fonte: autores

Em termos práticos, os alunos dos 6^{os} anos, Módulo I, portanto, passaram ao longo do ano conhecendo a ferramenta Scratch, suas funcionalidades e suas possibilidades. Inserir cenário, ator, interação entre atores, áudio, movimento, atividades que foram sendo assimiladas e aprofundadas gradativamente, de modo que a atividade final, com a qual demonstraram maior satisfação, foi elaborar uma história com a utilização de atores e cenários da turma do Chaves. De todas as atividades que foram desenvolvidas, junto aos estudantes do Módulo I, observou-se que a maior dificuldade deles foi com relação à programação do movimento dos personagens, para tal, utilizou-se o palco “xy-grid”, que apresenta o desenho de um plano cartesiano com as coordenadas X e Y o que colaborou para dirimir dúvidas.

Os alunos do 7^o ano, Módulo II, por já possuírem o embasamento introdutório realizado no Módulo I, tiveram suas atividades direcionadas para o desenvolvimento de jogos. Nesse módulo, os alunos fizeram uso de variáveis, de modo que começaram a dar vida em seus projetos à medida que a programação do jogo passava a contemplar tempo,

pontuação, vida e fases. Entre os trabalhos desenvolvidos dos quais os estudantes mais gostaram, consta corrida de turfe, salto com obstáculos, labirinto, adivinhe um número de 0 a 100 e acerte o alvo.

No módulo III, os alunos do 8º ano utilizaram a ferramenta do *app inventor* para o desenvolvimento de aplicativos para celulares. Como trabalho final do curso, foi sugerido que desenvolvessem dentro dos limites do aprendizado individual aplicativos funcionais, ou seja, que pudessem ser utilizados pelas pessoas ou algum tipo de profissional como ferramenta de trabalho no seu dia a dia. Dessa tarefa, criou-se um aplicativo de boletim, no qual o professor insere as notas das provas e trabalhos de todas as disciplinas e o aplicativo calcula a média e deixa armazenado o seu valor no banco de dados. Outro aplicativo desenvolvido foi uma calculadora que permite que as pessoas possam efetuar o cálculo para saber se economicamente é mais viável abastecer o veículo com gasolina ou álcool; o terceiro aplicativo foi desenvolvido para uso pedagógico, em que constam imagens e descrições das partes do corpo humano, e o quarto aplicativo foi desenvolvido para apresentar a indumentária gaúcha, em que, também, cada imagem trazia uma pequena descrição.

A importância que o letramento proporciona para aos jovens por meio do desenvolvimento de habilidades e competências, como a do raciocínio, a lógica, do trabalho em equipe, entre outros, é visível aos olhos dos profissionais envolvidos. No entanto as conquistas e/ou benefícios podem alcançar uma dimensão maior da que nossos olhos podem enxergar. Na aula do dia 26 de junho de 2018, quando solicitado para que os alunos do módulo I redigissem em seus cadernos de bordo o que tinham aprendido durante a aula e o que estavam achando do curso, surgiram vários apontamentos significativos, contudo um chamou atenção em especial, conforme segue:

É possível fazer jogos, aprender comandos se divertir, programação e eu gosto muito das aulas por que nós estamos aprendendo coisas boas que só aqui oferece pra nós por que em vez da gente estar na rua aprendendo o que não precisa aprender nós estamos aprendendo só coisas boas por que o mundo hoje só tem uma parte de pessoas boas

por que hoje em dia tu vira pra um lado tem um fumando maconha tu vira pro outro lado tem outro e aqui é diferente se tu olhar para a esquerda tem um aprendendo tu olhar pra direita tem outro e é errando que se aprende. (Aluno Módulo I)

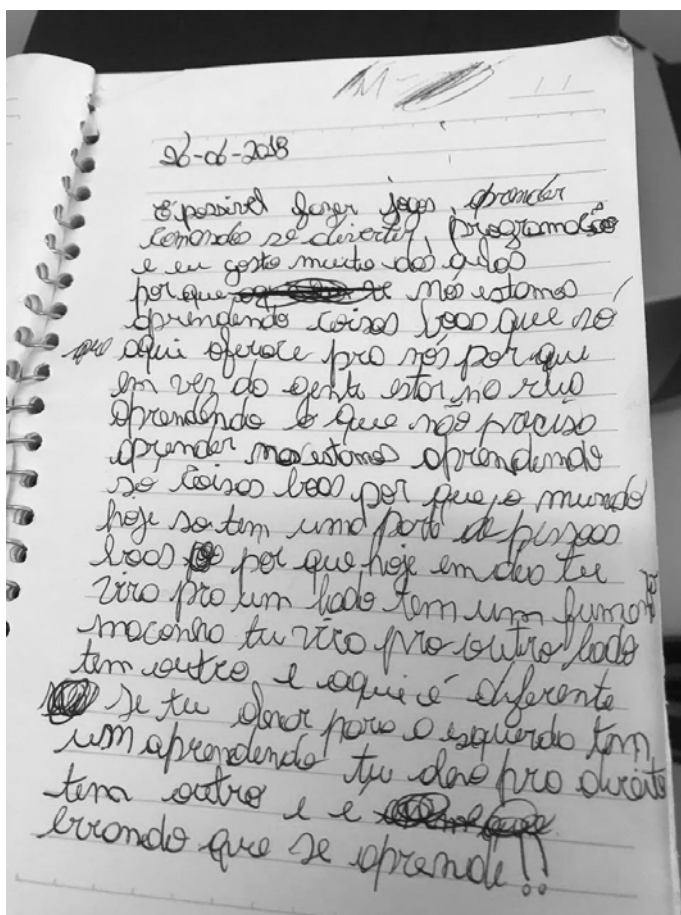


Figura 2 – Relato de aluno

Fonte: autores – dados de pesquisa

Mesmo que a reprodução da escrita do estudante contenha erros ortográficos e de coerência verbal, próprios de quem ainda

está se desenvolvendo, a mensagem foi muito clara. Residente de uma área urbana de vulnerabilidade social, o aluno relata as possibilidades que ele teria caso “estivesse na rua”. Contudo, fazendo uso de uma das orientações dadas em aula de que, quando estiver com dúvidas. “pergunte a três antes de mim”, ou seja, pergunte a três colegas antes de perguntar ao professor, o aluno coloca que, participando do letramento, ele “olha para esquerda tem um aprendendo, olha para a direita tem outro”.

Assim sendo, percebemos que o Letramento atinge as dimensões desejadas no que tange à educação e percepção de mundo, de possibilidades e de mudança. Em conformidade com a evolução digital, entendemos a importância de oportunizar a participação no Letramento em Programação. Ensinar na rede pública utilizando as tecnologias de informação preparando alunos para a sociedade contemporânea, para o mercado de trabalho em constante evolução, é uma forma democrática de garantir igualdade de direitos e equidade a uma parcela da população que não teria condições de frequentar cursos semelhantes se estes não fossem gratuitos.

Assim, cumprimos nosso papel de garantir educação de qualidade a todos, conforme preconiza a Constituição Federal de 1988. Nelson Mandela dizia que a educação é a arma mais poderosa que você pode usar para mudar o mundo, e nós, enquanto coordenadora e professores do curso de letramento em programação do município de Marau/RS, acreditamos que o letramento em programação é uma das armas mais promissoras e que possui um enorme potencial de contribuir para a mudança no mundo.

REFERÊNCIAS

IBGE – Instituto Brasileiro de Geografia e Estatística. *IBGE CIDADES*. Recuperado em 23 nov. 2018, de <https://cidades.ibge.gov.br/brasil/rs/marau>.

CAPÍTULO 12

A CULTURA MAKER INSERIDA NO LETRAMENTO EM PROGRAMAÇÃO

Peter Rodrigo Trento

INTRODUÇÃO

Este relato de experiência ocorreu em Vinhedo-SP, e apresenta o trabalho realizado pelos alunos de 9º ano da escola municipal Darci A. Briski, durante o contraturno escolar, os quais realizaram um projeto *maker* durante as aulas de programação. A proposta de letramento digital proposta pelo Instituto Ayrton Senna foi desconfigurada em nosso município, onde realizamos atividades que não estavam inicialmente no currículo, como oficinas *maker* e trabalho de campo, e o resultado dessas ações foi premiado com o Tech Oscar, além de proporcionar um grande engajamento dos alunos de programação e que levaremos para os próximos anos com esse projeto em nossa rede.

DESCRIÇÃO DA EXPERIÊNCIA

O município de Vinhedo-SP começou a participar do projeto de letramento em programação do Instituto Ayrton Senna no ano de 2017, espelhado a partir das experiências bem-sucedidas do município vizinho, Itatiba-SP, que havia iniciado o projeto já há 10 anos, e é um modelo de sucesso entre os participantes do Brasil.

Em seu primeiro ano, Vinhedo participou com apenas uma escola e seis alunos, e em seu segundo ano, ampliamos para três escolas da rede municipal e aproximadamente 70 alunos fazendo programação.

Esse relato de experiência foi realizado, no ano de 2018, na Escola Municipal Professora Darci Ana Dêgelo Briski, com os alunos dos 9º anos (13 a 14 anos) do módulo avançado do letramento em

programação. Esses alunos eram os seis participantes iniciais do projeto do ano de 2017, que já haviam trabalhado programação com Scratch e seguiram no próximo módulo em 2018. As aulas foram abertas a todos os interessados da escola, mas, infelizmente, foi muito baixo o número de alunos dispostos a voltar à escola no contraturno.

O Instituto Ayrton Senna elaborou um currículo para o letramento digital e vem refinando esse documento conforme vão ocorrendo as experiências das escolas participantes, daí a importância dos relatos de experiência como este. Para os módulos avançados do programa, esse currículo propõe um avanço cognitivo, aprofundando os trabalhos com o site de programação conhecido como Scratch (scratch.mit.edu), uma plataforma gratuita, criada pelo grupo do Professor Mitchel Resnick, no grupo Lifelong Kindergarten no Media Lab do Massachusetts Institute of Technology (MIT). Para esse ano, além do trabalho com Scratch, nos foi proposto um trabalho com uma placa programável, que consiste em um miniprocessador, chamada Microbit, idealizada pela British Broadcasting Corporation (BBC).

No início, os alunos tiveram aulas com o orientador pedagógico de matemática de nossa rede municipal, Eduardo Mauricio Moreno Pinto, e posteriormente com o coordenador de inovação e tecnologia educacional, Peter Rodrigo Trento. As primeiras aulas foram específicas para lembrá-los da experiência anterior com a programação em Scratch, mas logo ao conhecerem a plaquinha programável Microbit, não quiseram mais voltar para a programação com blocos no Scratch, pois a Microbit, na visão dos alunos, “trazia a programação para o mundo real”, ou seja, com a microbit, o que eles programavam no computador era visto no mundo físico, a partir de sons, luzes de led e interações, indicando uma significância muito maior para o contexto da turma.

Dessa maneira, a placa de programação foi adotada por esse grupo de alunos como principal ferramenta de trabalho no letramento em programação desse ano, e os professores de programação buscaram fontes e estudaram um pouco sobre metodologias ativas com o livro *Metodologias Ativas para uma educação Inovadora*, assim como discutiram sobre o trabalho do pesquisador Valente (2005): “A espiral da espiral de aprendizagem: o processo de compreensão do papel das tecnologias de informação e comunicação na educação”.

As aulas com a Microbit, inicialmente, foram focadas em apresentar cada um dos componentes presentes na placa. A polivalência desse hardware é enorme, onde estão presentes sensores integrados de acelerômetro (movimento e aceleração), magnetômetro (campos magnéticos), termômetro, luxímetro (intensidade luminosa), além de um painel composto de 25 lâmpadas de led, dois botões programáveis, entre outras saídas. Foram feitas diversas experimentações com cada sensor, empolgando os alunos a cada novo teste. Foram trabalhadas habilidades específicas com cada componente, como por exemplo, com o magnetômetro, com o qual foi realizada uma revisão de geografia e matemática, em que falamos de campo magnético da terra, pontos cardeais, o funcionamento de uma bússola, além de trabalharmos o pensamento matemático por meio do plano cartesiano, ângulos gerados por um raio de circunferência e seus intervalos. Dessa maneira, conseguimos fazer uma conexão da Microbit com alguns conteúdos vistos pelos alunos em suas aulas regulares.

As aulas do primeiro semestre foram praticamente focadas em conhecer a Microbit, minuciosamente testando cada sensor com os alunos. Em uma reunião dos coordenadores municipais desse projeto, verificou-se que a Microbit não foi utilizada de maneira tão efusiva como estava sendo utilizada nas aulas em Vinhedo. Já no semestre seguinte, decidimos sair do computador, fazendo atividades mais “mão na massa”, ligadas à cultura maker (aprendizagem a partir do fazer), buscando pelo conceito de programação em outros lugares. Houve uma discussão sobre o conceito de programar algo, como a própria vida é programada pela genética do nosso DNA, como nosso corpo e metabolismo está programado para certas ações, e outras programações do cotidiano, como do despertador, da televisão, do videogame, entre outros casos. Para complementar a ideia, quebramos mais uma vez a sequência didática proposta e trouxemos uma programação para ser trabalhada no papel, com canetas, por intermédio de um pequeno robô esférico conhecido como Ozobot e que foi muito bem recebida pelos alunos. A partir de tal experiência, levamos a atividade para os outros alunos de módulos diferentes de programação de nossa rede.

A ideia de se trabalhar a programação além de um prompt na tela do computador gerou um entusiasmo enorme por parte

dos estudantes. Isso influenciou nosso projeto final, que deveria ser entregue até o mês de dezembro. Ao invés de entregarmos um projeto em Scratch, faríamos um projeto com as Microbits.

Já no segundo semestre, os alunos iniciaram um diário de bordo, criando um site (<https://sites.google.com/view/codingnaescola>), relatando todo o processo de estudo com a Microbit, desde o início do ano. Algumas ideias de projetos surgiram, como a de um carrinho remoto, mas, ao estudarem alguns projetos, viram que não dependiam apenas da própria placa, mas de muitos outros componentes que não conseguiriam adquirir, alguns por serem caros e também por não haver tempo hábil para aprender a escrever os algoritmos necessários e mais complexos.

Como esse era um ano de eleições no Brasil, surgiu a ideia a partir dos próprios alunos de fazer um projeto com a microbit, utilizando-a de alguma maneira para consultar a opinião e desejos dos colegas de escola, assim, depois de uma discussão, os alunos decidiram programar os botões para captarem respostas de um censo escolar. Primeiramente, estudaram como criar um código para isso e, com a ajuda dos professores Peter e Eduardo, fizemos um código simples para captação de dados com respostas: “sim”, “não” e “talvez”. O botão A da plaquinha seria o “sim”, o botão B seria o “não” e os dois botões juntos seriam o “talvez”, e as respostas eram mostradas nas lâmpadas de LED. Depois de alguns testes, viram que não funcionava a opção do botão conjunto A+B. Assim, pedimos ajuda para o programador Luis Alves Neto, do próprio Instituto Ayrton Senna, que nos aconselhou sobre nossos erros de programação para o A+B e pudemos consertá-lo, e depois de muitos testes, pudemos aplicar com outros alunos com questões propostas pelos alunos de programação. Após avançarmos na construção da caixa de censo escolar com a microbit, decidimos levar todos os alunos de letramento em programação para uma visita até o Fablab em Vinhedo.

Dessa maneira, aproximadamente 60 alunos da rede municipal, que fazem o letramento em programação no contraturno das aulas, puderam conhecer e vivenciar uma atividade prática de construção de uma placa eletrônica no Fablab, em Vinhedo. Os Fablabs são espaços públicos de inovação, ou seja, espaços de criatividade, aprendizado, acessíveis a todos os interessados em desenvolver e

construir projetos. A partir de processos colaborativos de criação, compartilhamento do conhecimento, e do uso de ferramentas de fabricação digital, o Fablab traz a possibilidade de aprender, projetar e produzir diversos tipos de objetos, e em diferentes escalas. Os alunos adoraram a experiência de conhecer um laboratório tecnológico, onde puderam entender um pouco melhor as possibilidades de utilização das mais diversas ferramentas tecnológicas e de programação.



Figura 1 – Trabalho maker realizado durante as aulas de programação

Fonte: Trento, P.R. (2018)



Figura 2 – Alunos de 9^{os} anos interagindo com alunos do 5^o ano no Fablab
Fonte: Trento, P.R. (2018)

RESULTADOS E REFLEXÕES

A inserção da cultura *maker* dentro da sequência didática que nos foi proposta pelo Instituto Ayrton Senna enriqueceu muito o currículo do letramento, pois, por meio da microbit, os alunos trouxeram para o mundo concreto os algoritmos computacionais, e isso teve um impacto grande na evolução e entusiasmo dos alunos pelas aulas. Além disso, o trabalho de campo até o Fablab foi uma experiência que levaremos para os próximos anos do projeto, devido também ao papel motivador que a visita ao laboratório proporcionou aos alunos, como também para os professores que visitaram. Essas ações *makers* e de campo não estavam previstas no curso, mas acreditamos que foi essencial para o envolvimento dos alunos e dos professores do projeto.

Ao final das aulas de programação, o projeto realizado com a placa de programação (microbit) possibilitou a captação de dados consultando pessoas sobre qualquer assunto, podendo ser utilizado para censo, pequenas eleições, enquetes e entrevistas. Esse projeto, realizado pelos alunos Thaís Akemi Taniguti e Victor Hugo Bonifácio Marreca, da escola E.M Prof.^a Darci Ana Dêgelo Briski, orientados pelo coordenador de tecnologia educacional Peter Trento, foi o vencedor do Tech Oscar na categoria de “senso de equipe”, voltado para trabalhos de impacto coletivo e de cunho social para a comunidade escolar.

Espera-se que, com nossa experiência nesse ano, o Instituto Ayrton Senna repense o currículo do projeto, levando em conta o envolvimento e entusiasmo dos nossos alunos ao realizarem as atividades “mão na massa”, que estavam previstas apenas para os módulos seguintes (finais), com turmas mais avançadas, trabalhando Arduino e aplicativos. Foi muito importante o trabalho *maker* também com os alunos introdutórios do curso. Também deve ser inserida no currículo a visitação a locais ligados à tecnologia, como espaços makers, laboratórios e empresas de tecnologia, possibilitando aos alunos conhecerem ferramentas tecnológicas e profissões que muitos ainda não conheciam ou a que nunca tiveram acesso em seu cotidiano.

REFERÊNCIAS

Bacich, L., & Moran, J. (2018). *Metodologias Ativas para uma educação Inovadora*. São Paulo: Penso. 237 p.

Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009, nov.). *Scratch: Programming for All*. *Communications of the ACM*, v. 52, n. 11, pp. 60-67.

Valente, J. *A espiral da espiral de aprendizagem: o processo de compreensão do papel das tecnologias de informação e comunicação na educação*. Tese de doutorado, Campinas/SP, Unicamp, 2005.

CAPÍTULO 13

REALIZANDO A ATIVIDADE DA CONSTRUÇÃO DO JOGO DE LABIRINTO

Érica Luiza de Oliveira

INTRODUÇÃO

A atividade da Construção do Jogo de Labirinto foi escolhida para o relato por ter se destacado nas reações positivas que gerou entre as turmas em que o programa é desenvolvido. Os alunos se divertiram ao perceber o quanto podem aprender e estudar não somente da forma tradicionalmente imposta, e sim de uma maneira lúdica.

Quanto aos seus desempenhos nas disciplinas do componente curricular do sexto ano, notou-se nítida melhora em suas notas e um envolvimento maior dos alunos com as aulas, e de uma maneira especial, fizeram com que a escola inteira participasse e se envolvesse no programa, pois eles desenvolvem suas histórias e seus jogos e mostram para as outras turmas, gerando comentários e conversas sobre o letramento nos intervalos das aulas. Trabalharam bastante a concentração, conseguindo soltar a imaginação para realizar essa atividade, desenvolveram o raciocínio lógico e ficaram encantados em ver as tantas e diferentes possibilidades que a programação no Scratch permite.

DESCRIÇÃO DA EXPERIÊNCIA

A experiência relatada foi desenvolvida em duas escolas da rede municipal da cidade de Tapejara, no estado do Rio Grande do Sul. São estas: Escola Municipal de Ensino Fundamental Marieli Andreola e Escola Municipal de Ensino Infantil e Fundamental São Paulo.

As escolas estão situadas em diferentes bairros na cidade de Tapejara e participam do Programa Letramento em Programação com os alunos do 6º ano. Os encontros do Letramento acontecem semanalmente nas sextas-feiras no turno da tarde. A atividade “Construção do Jogo de Labirinto” teve início em ambas as escolas no dia 14 de setembro de 2018, e foi concluída no dia 5 de outubro de 2018. Participaram da ação os estudantes do 6º ano que fazem parte do programa e que estavam presentes nas datas mencionadas, sob orientação da professora Érica Luiza de Oliveira.

No primeiro momento, foi retomada a ideia do que é um jogo de labirinto. Os alunos mencionaram jogos na mesma temática, recordando que o objetivo seria passar de um lado para o outro ou de um ponto inicial até o ponto final, e que geralmente um personagem quer encontrar outro personagem ou objeto com o qual tenha afinidade.

A partir desse momento, cada aluno começou a pensar e anotar no seu diário de bordo qual seria o seu personagem e qual objeto esse personagem gostaria de encontrar ao final do trajeto do labirinto. Buscaram imagens na internet para inserir como personagens nos seus projetos. As aulas foram baseadas na atividade 7, “Mídias Digitais – Jogos”, do material disponibilizado. O primeiro passo dado foi desenhar o labirinto na própria ferramenta Scratch, construindo, assim, o cenário do jogo.

Os alunos, inicialmente, tiveram dificuldade em desenhar o cenário, pois queriam desenhar com linhas retas e não estavam conseguindo. A orientação passada foi de que utilizassem a linha ao invés do lápis e, ainda, antes de começar a fazer o traço com a linha, mantivessem pressionada a tecla “shift” até terminar o traço, e só depois a soltassem. Assim, como resultado, tiveram cenários dos labirintos agradáveis visualmente, retos, como tanto desejavam.

Em seguida, inseriram no Scratch os atores, que foram: um personagem e um objeto que esse personagem encontraria no final se conseguisse percorrer todo o labirinto sem encostar nas linhas. A imaginação das crianças chamou a atenção, foram muito criativas encontrando os personagens e os objetos dos seus jogos. Empolgaram-se pensando em motivos que realmente fizessem sentido para que determinado personagem quisesse encontrar o objeto.

Depois de construir o cenário e inserir os atores, foi programada a movimentação do personagem. Foi o primeiro contato com a programação utilizando as teclas de setas de direção; em um primeiro momento, acharam estranho, mas logo perceberam e disseram que fazia todo o sentido, pois a própria seta já indicava para que lado o personagem iria se movimentar. Adoraram quando perceberam que eles estavam controlando o personagem que eles mesmos inseriram e, ainda, que eles haviam programado inclusive para que lado iria se movimentar se pressionassem determinada tecla. Precisaram realizar alguns ajustes nos tamanhos dos atores, pois havia a necessidade de que fossem menores que as linhas do labirinto. Redimensionaram para que o tamanho ficasse adequado, e isso também foi novidade para muitos, que ficaram realizados em aprender mais essa condição.

Para finalizar, programaram a interação entre o personagem com o cenário e com o objeto. O personagem, ao encostar no labirinto, retornava para o ponto inicial e deveria começar a jogar novamente. E ao encostar no objeto que se encontrava no final do labirinto, o objeto se movimentava e continha caixas de mensagens parabenizando o jogador por ter conseguido finalizar o jogo. Alguns alunos gostaram tanto da atividade que demonstraram interesse em continuar implementando o jogo inserindo novas fases, o que não conseguimos realizar por falta de tempo. Encerramos essa atividade e seguimos o material disponível, com outras das atividades planejadas.

RESULTADOS E REFLEXÕES

Os alunos adoraram essa atividade, pois conseguiram visualizar diversas possibilidades para serem inseridas nas suas histórias, que até então nem imaginavam que era possível e que conseguiriam realizar. Proporcionou a todos conhecimento em todas as partes de programação para desenvolver a atividade. Utilizaram sua criatividade, imaginando quais seriam os personagens, como seria o cenário, além do raciocínio para poder desenvolver de forma coerente as suas programações.

No decorrer do programa, houve uma melhora significativa do comportamento dos alunos, que se dedicaram mais às aulas, suas notas obtiveram acréscimos expressivos, resultante da participação das crianças no Letramento em Programação.

Além disso, a experiência de ter aula integralmente agradou tanto aos alunos quanto aos pais, que aprovam a educação integral em que seus filhos têm a oportunidade de aprender e se desenvolver mais intelectualmente e humanamente, pois, além de vencerem seus medos, como a timidez e a vergonha, conseguem se desenvolver na dicção enquanto apresentam seus trabalhos, aprendem sobre respeitar os possíveis erros do colega e, ao invés de criticá-lo, sanar suas dúvidas, procurando soluções para os seus códigos.

Obtiveram maior independência, pois procuram o que até então não sabiam, sem ficar dependendo da resposta pronta da professora, e essa independência levam para a sala de aula, recorrendo aos livros e ao material anotado no caderno para só posteriormente, não sanando a dúvida, recorrerem à professora, caso contrário, resolvem por conta própria as suas indagações sobre os desafios que lhes são propostos.

As atividades também despertaram o querer bem ao próximo, eles não concorrem um com o outro para ver quem sabe mais ou quem termina primeiro, pelo contrário, se ajudam, compartilham suas dificuldades, tomando como prioridade a gentileza e o respeito com o próximo.

Também trocam experiências, o que é muito gratificante, pois eles conseguem ter o olhar da gentileza, e aqueles que vão encerrando suas atividades, antes de irem para a próxima atividade, verificam se não há algum colega que esteja enfrentando algum obstáculo que já foi por ele enfrentado.

Sendo assim, é visível o quanto os alunos que participam do Programa Letramento em Programação se desenvolvem como pessoa, isso já no primeiro módulo que os nossos alunos estão frequentando, que é o sexto ano. Acreditamos que, no decorrer dos próximos módulos, todas as qualidades que eles acrescentaram para suas vidas serão ainda mais notáveis.

Nas figuras a seguir, será possível a verificação dos alunos desenvolvendo e testando seus jogos de labirintos. Como se percebe, cada aluno programou os seus projetos conforme orientação da professora e utilizaram da sua imaginação na hora da criação do cenário e da escolha dos personagens. Todos foram coerentes e escolheram cenários e personagens que tivessem nexos, que fizessem sentido dentro do contexto.

Os alunos se interessaram em jogar e ver os cenários, personagens e códigos dos colegas, verificando o que poderiam utilizar para aprimorar nos seus códigos, fazendo uma espécie de feedback com os colegas, formando opiniões de como poderia ser escrito de maneira diferente, buscando possíveis falhas, testando todas as funcionalidades dos jogos.

Eles, de uma forma rigorosa, são seus próprios jurados, pois tentaram encontrar seus erros para que cada vez programem de maneira mais eficaz, com menos repetição de código desnecessário, e assim distribuem dicas quando percebem que algo poderia ser modificado. Pretendem seguir no Letramento em Programação, pois dizem que estão só começando a desvendar o mundo de possibilidades que esse universo da programação está apresentando a eles.

A Figura 1 mostra alguns dos alunos da Escola Municipal de Ensino Fundamental Marieli Andreola, da turma do 6º ano, programando seus projetos, cada um em seu tempo. Não é imposto um tempo rígido para que encerrem a programação; como se nota na foto, os alunos estão em diferentes fases da criação do labirinto.

A Figura 2 é de uma aluna da turma do 6º ano da Escola Municipal de Ensino Infantil e Fundamental São Paulo, já testando e jogando no projeto elaborado por ela.



Figura 1 – Alunos da Emef Marieli Andreola desenvolvendo a atividade de Construção de Labirinto

Fonte: autores

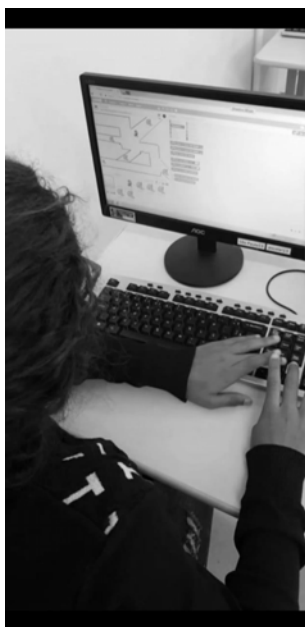


Figura 2 – Aluna da Emeif São Paulo praticando o seu projeto do Labirinto

Fonte: autores

CAPÍTULO 14

AS DISTINTAS EXPERIÊNCIAS DOCENTES QUE FAVORECEM AS PRÁTICAS PEDAGÓGICAS

Eduardo Mauricio Moreno Pinto

INTRODUÇÃO

O projeto piloto ocorreu com alguns alunos dos 9^{os} anos de uma escola pública da Rede Municipal de Vinhedo, E.M. Prof.^a Darci Ana Dêgelo Briski, as aulas foram planejadas e implantadas por dois docentes de distintas áreas do conhecimento e funções dentro da Secretaria de Educação. O objetivo da dupla de professores estava em experimentar a implementação das propostas do programa Micro:bit pertencente ao projeto Ayrton Senna, porém optamos por reavaliar o planejamento de cada uma das aulas. Em geral, houve diversas adequações, possibilidades e inovações na sequência didática. Este artigo apresentará o meu ponto de vista particular, destacando alguns momentos vividos durante o primeiro semestre de 2018, influenciado pelas minhas experiências com a Informática na Educação e a Educação Matemática. Dessa forma, espero contribuir com toda a comunidade da própria Rede Municipal de Vinhedo, da Tecnologia Educacional, dos professores de Matemática e, ao mesmo tempo, contribuir com as pesquisas acadêmicas.

DESCRIÇÃO DA EXPERIÊNCIA

A implementação do programa Micro:bit ocorreu em uma escola pública da Rede Municipal de Vinhedo/SP, E.M. Prof.^a Darci Ana Dêgelo Briski, localizada na Av. Presidente Castelo Branco, 140. As aulas foram programadas para ocorrer em todas as sextas-feiras (segundo as possibilidades do calendário escolar) e durante o

contraturno (período da tarde, das 13h às 14h30). O projeto escolar envolveu apenas alguns alunos dos 9^{os} anos e essa decisão em escolher um número limitado de alunos se deve ao fato de a sala de informática da escola disponibilizar apenas nove CPU's e cada um com o seu próprio monitor, destes, três não funcionavam, logo, havia apenas seis alunos participando do projeto piloto (três meninos e três meninas). O projeto prosseguiu durante todo o ano de 2018, mas a minha participação ocorreu apenas no primeiro semestre e, nesse período, as aulas foram planejadas e implantadas por dois professores que trabalham na Secretaria de Educação da Rede Municipal de Vinhedo, o orientador e professor de Matemática Eduardo Mauricio e o coordenador de Tecnologia Educacional e professor de Geografia Peter Trento.

O intuito estava em mesclar as experiências profissionais e acadêmicas de cada um dos professores, visando à melhoria do projeto Micro:bit. Por um lado e de uma forma resumida, o Professor Eduardo Mauricio apresenta a seguinte experiência: atuou como tutor no Núcleo de Informática Aplicada na Educação (Nied) da Unicamp, desenvolveu projetos próprios nas salas de informática das escolas da região e elaborou uma dissertação de Mestrado – ver Pinto (2018) –, envolvendo as seguintes temáticas: *Software Educativo Scratch*, conceitos da Didática amparados pela Educação Matemática e da Matemática amparados pela Educação e Informática na Educação.

Por outro lado, o professor Peter Trento apresenta as seguintes experiências: revisor do *Youtube Educação*, líder do Grupo de Educadores Google (GeG) e vencedor do prêmio Professores do Brasil do Ministério da Educação (MEC), trabalha com a formação de Tecnologia Educacional para Professores desde 2014, envolvendo diversas ferramentas tecnológicas, dentre elas *Google for Education*.

Uma das primeiras preocupações da dupla de professores estava em trabalhar os conceitos da cultura *maker* e da Informática na Educação. Neste artigo, vou seguir a descrição da segunda temática, pois se trata de uma abordagem vinculada com a minha própria experiência acadêmica. Inicialmente, a Informática na Educação apresenta uma diversidade de trajetórias e, neste artigo, oriento a discussão para a distinção entre dois conceitos sugeridos por Papert (1993): Instrucionismo vs Construcionismo. Em geral, entende-se

por Instrucionismo aquilo que “expressa a ideia de que o aperfeiçoamento da instrução implica na melhoria da aprendizagem, se a Escola tem deficiências então sabemos o que fazer: ensinar melhor” (Papert, 1993, p. 124); refere-se a um modelo tradicional de ensino e de aprendizagem que valoriza apenas a instrução, o procedimental, a memorização, o conceitual, a reprodução de ideias expostas no livro didático, o trabalho individual do aluno.

O Construcionismo está fundamentado a partir do Construtivismo de Jean Piaget e surge como uma oposição ao Instrucionismo, ou seja, existe a possibilidade de o professor compreender um conjunto de fundamentações teóricas do Construtivismo junto com as investigações metodológicas que exploram a diversificação dos estágios de desenvolvimento da criança, a partir da sua própria relação direta com o objeto. O Construcionismo parte do mesmo princípio, porém a criança mantém uma relação direta com o *hardware* Micro:bit e a linguagem de programação disponível no site de edição <http://makecode.microbit.org>.

Porém o aluno dificilmente irá desenvolver plenamente as suas habilidades por conta própria, daí vem a importância de um trabalho que mescle o Instrucionismo e Construcionismo, ou seja, aulas planejadas havendo a oscilação entre os dois conceitos, utilizando as instruções de modo a explorar o campo cognitivo do aluno com perguntas, indagações, situações-problema, disponibilizando espaço para os alunos desenvolverem a autonomia e a socialização de suas ideias.

Portanto, a dupla de professores passou a refletir e a evitar uma reprodução fiel dos vídeos disponíveis no *Youtube* e dos *slides* ou *cards* desenvolvidos anteriormente pelos programas Micro:bit vinculados com o letramento em programação organizado pelo Instituto Ayrton Senna, ou seja, imaginamos que seguir uma sequência didática pronta, apenas com o intuito de reproduzir os conteúdos, significaria valorizar apenas a reprodução do conjunto de conteúdos, algo que apresentaria uma similaridade com o Instrucionismo. Isso poderia tornar a aula pouco atraente aos alunos. O conjunto de aulas elaboradas pelos dois professores está sintetizado na próxima seção, cada aula apresenta um resumo dos objetivos, do desafio do dia, resultados e/ou reflexões pessoais.

RESULTADOS E REFLEXÕES

Data: 06/04/2018. **Objetivos aula 1:** introdução ao curso; revisão de alguns conceitos desenvolvidos com Software Educativo *Scratch*; apresentação superficial dos componentes do Micro:bit e possibilidades. **Sobre a aula:** o intuito estava em apresentar o projeto, perspectivas futuras, apresentação pessoal de cada um dos professores, resgatar conhecimentos prévios desenvolvidos no projeto de 2017 envolvendo o Software Educativo *Scratch*. A ideia estava em discutir com os alunos a necessidade de desenvolver o conceito procedimental envolvendo a “**tentativa e erro**”, o “**senso crítico**”, “**autonomia da reflexão**” e “**tomada de decisão**”, seguindo uma proposta de trabalho que envolva uma mescla entre Instrucionismo e Construcionismo (Papert, 1993) e a prática da depuração (Valente, 2005) como forma de unir os dois conceitos apresentados pela Informática na Educação. Durante a aula, os alunos apresentaram interesse pelas possibilidades futuras, perguntando e interagindo com os professores e colegas.

Data: 13/04/2018. **Objetivos aula 2:** apresentação detalhada dos componentes do Micro:bit; visita ao site oficial <https://makecode.microbit.org/> e orientações sobre o seu funcionamento no site; exploração livre dos alunos envolvendo os blocos de comandos; finalização da aula com o desafio: “contador de passos”; erro inesperado; usar um *led* para sincronizar com o passo; zerar memória utilizando o botão B. **Sobre a aula:** o planejamento desta aula envolve a retomada do final da aula anterior, ambos os professores compreendem que o processo de depuração (Valente, 2005) ocorre de duas maneiras: ao tratar do hardware Micro:bit de forma aprofundada e ao tratarmos da diversidade e possibilidades da linguagem de programação. Em ambos os casos, estamos correlacionando a temática da aula anterior e os conhecimentos prévios discutidos em torno do Software Educativo *Scratch*. Na Figura 1, há um momento da aula no qual a pesquisa no *google* poderia trazer ideias de projetos, soluções ou novas problematizações.



Figura 1 – Discussão da situação-problema entre professores e alunos
Fonte: autores

Um momento para ampliar horizontes e evitar que os alunos perdessem o rumo do projeto. Assim, ao apresentar o objetivo da aula, “contador de passos”, deixamos em aberto a discussão de como poderíamos resolver, ou seja, cada aluno, ao ler os comandos disponíveis, sistematizaria o procedimento de “tentativa e erro”, ao mesmo tempo, os professores foram acompanhando o trabalho dos alunos e buscando o aluno que pudesse socializar alguma ideia ou situação problemática. Esse tipo de aula é algo atípico para os alunos que estavam acostumados com o modelo tradicional de aula. Talvez a motivação desses alunos em participarem da aula era um indício do impacto causado pela experiência de um modelo de aula alternativo. No fim da aula, descobrimos o primeiro obstáculo para os professores, testamos o “contador de passos” no pátio da escola, alguns alunos chegavam ao valor aproximado ao número de passos, outros apresentavam o valor diferenciado. Pedimos um tempo aos alunos e propomos trazer uma solução ou posicionamento até a próxima aula.

Data: 20/04/2018. **Objetivos aula 3:** retomada da aula anterior e correção do erro inesperado; conhecendo o JavaScript; alterações no blocos de comandos e suas implicações. Explorando os *leds* e as suas possíveis combinações de comandos. Explorando a Matemática no Micro:bit. **Sobre a aula:** descobrimos durante o planejamento e

utilizando a mesma ideia sugerida aos alunos, tentativa e erro, que a problemática da aula anterior trata-se da forma do manuseio do *hardware*. Na aula anterior, todos os alunos e professores estavam carregando na mão o Micro:bit, porém, ao colocar o *hardware* na meia, o “conta passos” apresentava a mesma contagem do número real de passos contados verbalmente. Isso me trouxe uma nova sensação: o conceito procedimental envolvendo a “tentativa e erro” é válido para todos, inclusive aos professores. Aqui, entra mais uma sensação pessoal, pois esse modelo de aula alternativo sugere uma nova relação professor-aluno, algo inexistente na escola e na vida do aluno, mas que passa a trazer novas possibilidades com relação à construção do conhecimento. Além disso, apresentamos a correlação entre *JavaScript* e os blocos de comandos utilizando um exemplo simples, declaração de variáveis. Em seguida, passamos para um momento de socialização de ideias na lousa e de instruções envolvendo a Matemática (Plano Cartesiano) e o Micro:bit, isso possibilitou a exploração livre por parte dos alunos dos blocos de comandos e que poderiam apresentar algum vínculo com a temática, ou seja, outro momento de aprendizagem, sob viés do Construcionismo.

Data: 27/04/2018. **Objetivos aula 4:** apresentação do desafio: o Micro:bit irá mostrar os números em contagem regressiva. **Sobre a aula:** sem perder o vínculo com a Matemática, propomos uma aula envolvendo a construção dos Números Naturais, sucessor e antecessor, e novamente a lousa foi utilizada para socializarmos as ideias e instruir sobre os conceitos formais que estavam sendo envolvidos. Portanto, o desafio estava em vincular uma ideia simples da linguagem Matemática com os Blocos de Comandos, a modificação da representação simbólica traz ao aluno o desafio, um momento de adaptação dos alunos em explorar os comandos até conseguirem algo adequado. No final, os alunos socializaram as distintas soluções que eles construíram.

Data: 04/05/2018. **Objetivos aula 5:** projeto luz; breve conversa em roda sobre procedimentos importantes envolvendo a seguinte temática: “posso estar com alguma nova ideia ou indagação e isso pode ser uma possibilidade para uma grande ideia”. Desafio: como medir a temperatura? Construir o *script* e testar no Micro:bit (temperatura ambiente da sala, geladeira e temperatura fora da escola).

Como medir a intensidade de luminosidade? **Sobre a aula:** a aula foi planejada para envolver questões que envolvem a disciplina de Ciências. A unidade da medida da luz (lux) e as influências do sol na temperatura ambiente. A partir dessa contextualização, os alunos ficaram livres para explorar os comandos que permitissem o Micro:bit funcionar como um termômetro. Seguindo o mesmo modelo de aula alternativo, socializando as ideias, indagações e soluções, partimos para o teste real do experimento, utilizando como local do experimento: armário (ambiente escuro), geladeira e o próprio ambiente da sala de informática e do sol.

Data: 18/05/2018. **Objetivos aula 6:** magnetômetro; revisão de geografia: campo magnético da terra, pontos cardeais, significado e funcionamento da bússola; revisão de matemática: plano cartesiano, ângulos gerados por um raio de circunferência, intervalo, adaptação dos conceitos discutidos para o Micro:bit. **Sobre a aula:** a aula foi planejada para atender a características multidisciplinares, ou seja, o magnetismo é um tema abordado pelas disciplinas do PEB II de Ciências, Geografia e Matemática. Junto à linguagem de programação, a aula ganha uma alta complexidade teórica (ver Figura 2), da fundamentação teórica, socialização das ideias dos alunos descrevendo a criação de estratégias de resoluções próprias.

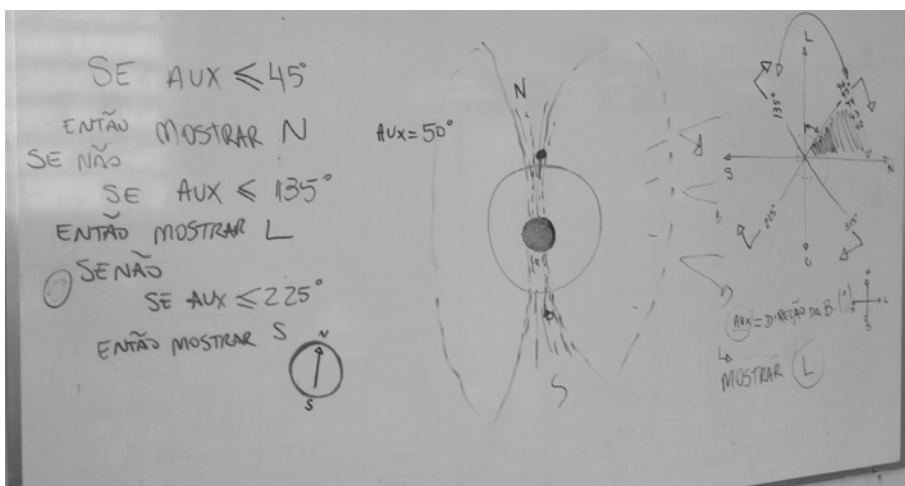


Figura 2 – Discussão da situação problema entre professores e alunos

Fonte: autores

Em geral, o progresso das aulas demonstrou uma evolução da aprendizagem dos alunos, pois eles puderam participar e acompanhar todos os momentos, partindo de um menor grau de complexidade para um de maior complexidade. Experimentaram o conceito procedimental da “tentativa e erro” e a construção do conhecimento. O engajamento dos alunos em todas as aulas representa um processo de valorização da depuração (Valente, 2005), o modelo de aula alternativa traz um impacto na vida do aluno e da escola, simplesmente por apresentar uma proposta diferenciada da relação entre os pares e entre professores-alunos. O projeto colaborou para implementar uma ideia em uma escola e uma possibilidade de proposta para toda a rede, evidenciando as práticas pedagógicas alternativas.

REFERÊNCIAS

Papert, S. (1993). *The children's machine: rethinking school in the age of the computer*. Library of Congress Cataloging in Publication.

Pinto, E. M. M. (2018). *A didática e a matemática no ensino da linguagem de programação: uma experiência com o software educativo Scratch no ensino fundamental*. Dissertação de mestrado, Unicamp. Campinas.

Valente, J. (2005). *A espiral da espiral de aprendizagem: o processo de compreensão do papel das tecnologias de informação e comunicação na educação*. Tese de doutorado, Campinas/SP. Unicamp.

CAPÍTULO 15

UM OLHAR SOBRE AS CONSTRUÇÕES – ANALISANDO INTERAÇÕES E APRENDIZAGENS

Sonia Mara Martins de Mattos Souza

INTRODUÇÃO

O presente relato tem o propósito de descrever a experiência realizada em ambiente informatizado, no contexto do projeto Letramento em Programação, com os alunos de uma escola do campo. A experiência ocorreu envolvendo nove alunos de uma turma multisseriada do 4º e 5º anos. O eixo norteador deste relato é o meu “olhar” sobre as construções, em que procuro analisar as interações e aprendizagens realizadas nesse contexto, no sentido de identificar fragilidades, dificuldades e limitações, buscando formas de intervenção de acordo com as necessidades apresentadas.

Ao observar e acompanhar o desenvolvimento da capacidade do uso do pensamento computacional do aluno, é fundamental fazer as intervenções e mediações necessárias para que ele consiga seguir, apropriando-se dos códigos da linguagem de programação, desenvolvendo e aprimorando o pensamento lógico e a capacidade para resolver problemas. Foi com essa intenção que criei um diário de bordo, no qual descrevi as atividades propostas para os alunos, o meu olhar sobre as construções, analisando as interações e aprendizagens e o olhar do aluno sobre as suas construções, no momento em que ele socializava as produções, além da autoavaliação, momento em que eles refletiam sobre o que haviam aprendido naquele dia. Esse recurso possibilitou que eu pudesse rever a metodologia que eu estava utilizando para trabalhar a prática, replanejar as minhas atividades e retomar noções básicas iniciais. Muitas vezes, tive que retornar e buscar outras formas de trabalhar a compreensão de alguns conceitos, levando mais tempo do que o previsto para algumas atividades simples, que foram significativas para a consolidação das relações e aprendizagens.

DESCRIÇÃO DA EXPERIÊNCIA

O contexto em que ocorreu essa prática é uma escola estadual localizada em São Roque, zona rural do município de Passo Fundo. Passo Fundo está localizado na Região Sul do Brasil, pertencendo ao estado do Rio Grande do Sul. Iniciamos o trabalho com os alunos em 26 de junho de 2018 e concluímos no dia 1º de novembro de 2018, com a participação na cerimônia de formatura.

A turma com a qual tive o privilégio de vivenciar essa experiência pertence a uma classe multisseriada do 4º e 5º anos da Escola Estadual de Ensino Fundamental Abhramo Ângelo Zanotto. Os professores envolvidos neste projeto foram a diretora da escola, a professora da turma, os pais dos estudantes e o NTE Passo Fundo – Núcleo de Tecnologia Educacional.

A diretora buscou a parceria para levar o Letramento em Programação para a escola, pois, ao conhecer o programa, percebeu que este podia ser uma ferramenta de suporte no processo de alfabetização e letramento, facilitando a compreensão de conteúdos curriculares, pois possibilita o desenvolvimento do raciocínio lógico e a capacidade de resolução de problemas, além de estimular a criatividade, a autonomia, o desenvolvimento do trabalho em equipe, a cooperação/colaboração, e especialmente o desenvolvimento do pensamento computacional e das competências cognitivas e socioemocionais, necessárias para o século XXI. A professora da turma acompanhou o trabalho e auxiliou no desenvolvimento da produção escrita dos estudantes, para que estes pudessem construir as histórias animadas, criadas com o software Scratch. Os pais deram apoio aos filhos, incentivando e valorizando cada aprendizagem realizada por eles. O NTE Passo Fundo disponibilizou suporte técnico, instalando o software Scratch para rodar *offline*, pois o laboratório de informática da escola não possui internet.

A experiência foi realizada com encontros semanais, no turno da tarde, por um período de três horas. A metodologia utilizada foi alternar momentos de manipulação, exploração do Scratch, de forma lúdica, seguido de momentos de desafios por intermédio da criação de um jogo, uma história etc. A etapa seguinte era a da socialização

das criações realizadas naquela aula, e no final tínhamos o momento da autoavaliação das aprendizagens. Para que eu pudesse analisar as interações e aprendizagens, organizei um quarto momento, que foi quando observava e anotava as descobertas deles, enquanto exploravam o software, e especialmente quando criavam, atendendo ao desafio proposto.

O registro da prática que chamei de “**Diário de Bordo**” era realizado antes de cada aula, com descrição da atividade do desafio e, após a aula, com as impressões/análise das aprendizagens, descobertas, dificuldades, potencialidades, além do registro da fala dos alunos sobre as criações, quando socializavam e no momento da autoavaliação, quando falavam sobre o que aprenderam. Para exemplificar, o Diário de Bordo seguia esta sequência de registro:

1º Momento:

Aula n.º X – (data)

Desafio: Criar uma animação com personagens.

2º Momento: Meu olhar sobre as construções – Analisando interações e aprendizagens.

3º Momento: Olhar do aluno sobre sua criação – Socializando as produções.

4º Momento: Autoavaliação: o que aprendemos hoje?

RESULTADOS E REFLEXÕES

A dinâmica descritiva, relatando uma prática vivenciada, a partir de registros como um diário de bordo ou uma memória da prática, implica um movimento reflexivo dessa prática. Nesse movimento, as aprendizagens são muitas e significativas, uma vez que surgem das nossas inquietações, impulsionando-nos a pensarmos formas de intervenção e mediação, diante da análise das aprendizagens

dos alunos. Com isso, nossas próximas ações precisam ser repensadas, replanejadas, redimensionadas para atender às necessidades apresentadas.

A seguir, trago o registro de uma das aulas em que é possível perceber as descobertas e dificuldades nas aprendizagens realizadas.

5ª AULA – 31/07/2018

1. Desafio: criar um diálogo entre personagens (Sprites). Escolher dois personagens para criar um diálogo (observar a escolha dos comandos utilizados pelos estudantes para criar o diálogo entre os personagens).

Meu olhar sobre as construções – Analisando interações e aprendizagens:

Os alunos descobriram sozinhos quais comandos usariam para criar a fala do personagem. Ao executarem o comando, perceberam que os personagens falavam ao mesmo tempo. Desafiei-os a buscarem um comando que controlasse o tempo de fala de cada um e o personagem que falaria primeiro. Alguns colocaram o tempo igual para os dois personagens e as falas continuaram ao mesmo tempo. Fiz a intervenção fazendo-os olharem para os comandos usados para cada personagem e perceberem se estavam iguais ou diferentes. Nesse olhar, entenderam que, se um personagem fala primeiro, o outro deve esperar um tempo (que é o tempo de ler a fala), para depois responder. Consequentemente, os comandos para os dois personagens não poderiam ser iguais. Nesse momento, compreenderam as diferenças entre os tempos de fala e resposta dos diálogos que estavam criando e fizeram as modificações nos comandos. Alguns alunos só compreenderam que conseguiriam organizar os comandos do tempo de espera entre uma pergunta e outra do diálogo quando criaram várias frases para o diálogo.

Alguns criaram diálogos interessantes com perguntas e respostas variadas, indo além do proposto, criando movimentos para os personagens executarem, além dos diálogos. Outros criaram diálogos simples, com somente uma ou duas frases.

Olhar do aluno sobre sua construção – Socializando as produções/construções:

O grupo foi organizado para que olhassem as produções de todos e percebessem as diferenças entre os comandos escolhidos pelos colegas e o seus comandos. Enquanto olhavam as produções, solicitei que observassem o que o(a) colega fez diferente além do que havíamos proposto. Um aluno mostrou seu trabalho em tela cheia. Uma parte dos colegas não sabia como fazer isso e questionaram o colega sobre como exibir em tela cheia e o colega mostrou. Quando os diálogos criados eram simples e curtos (somente uma frase), alguns colegas perguntavam: “Só isso?”. Deixei que eles percebessem que poderiam ter avançado mais na construção dos diálogos.

Autoavaliação: o que aprendemos hoje?

Feedback sobre as aprendizagens construídas:

- O que deu certo?
- O que precisaram refazer e por quê?
- O que mais poderiam ter feito, para que compreendessem o processo do que construíram durante a aula?
- Quais foram as aprendizagens realizadas?

Essas questões levantadas no momento da autoavaliação possibilitaram aos alunos perceberem o que sabiam e o que ainda precisavam aprender. Uma atividade que levou mais tempo do que eu havia previsto foi a criação de histórias no Scratch. Solicitei que os alunos criassem uma história. Depois do esboço realizado, pedi que digitassem no computador, para depois construir no Scratch. Não deu certo, as histórias eram vagas, não tinham começo, meio e fim. Solicitei auxílio para a professora da turma no sentido de fazer uma análise juntamente com os alunos para que pudessem olhar seus escritos, fazendo uma revisão, não apenas para corrigir erros ortográficos e gramaticais, mas cuidando para que o texto cumprisse sua finalidade comunicativa. Organizei algumas questões para os alunos analisarem as histórias.

Questões para analisar a história criada:

1. Minha história possui um título que dá uma ideia do acontece no seu decorrer?
2. Minha história possui uma linha coerente de narrativa com começo, meio e fim?
3. Quem são os personagens? Defina as características de cada um: quem e como são, o que fazem (aparência, gestos, fala).
4. Quais são as principais ações dos personagens e o que elas desencadeiam na história?
5. Onde vivem? Essa questão é importante, pois o local onde se passa a história será o cenário que é o Palco, no Scratch. Para cada local onde se passa a sua história, você terá que criar um palco.
6. Onde a história aconteceu?
7. Explique o que aconteceu com eles (problema) e como eles resolveram a situação.
8. Quantos diálogos (falas) existem na minha história?
9. Na minha história existem pensamentos? Quantos?
10. Na minha história existem onomatopeias? Quantas?

Meu olhar sobre as construções – Analisando interações e aprendizagens:

Ao responderem as questões propostas para analisar as histórias criadas, os alunos perceberam que alguns elementos da história não estavam bem claros, como por exemplo, a linha da narrativa com começo, meio e fim. Faltava também as ações dos personagens, os locais onde a história acontece e a definição dos diálogos, em torno da situação (problema, mistério) relatada na história. Essa atividade durou três aulas. Em todo momento, eu e a professora da turma fazíamos as intervenções para os alunos, orientando a escrita das

histórias, levando-os a perceberem os elementos que devem estar presentes em uma produção narrativa.

Com as histórias criadas, o próximo passo era construí-las no Scratch. Nessa parte, os alunos também tiveram dificuldades. Para que eles conseguissem avançar nas construções, criei um roteiro, a exemplo dos roteiros utilizados para criação de vídeos. A seguir, descrição da aula com o uso do roteiro.

10ª AULA – 11/09/2018.

Continuidade da construção das histórias que serão criadas no Scratch.

Construir um roteiro para a história criada, descrevendo as cenas com lugar, período, ambiente, personagens, ações, falas etc. Distribuir as cenas para cada palco do Scratch que irá criar, de acordo com a história. Questionar os alunos se sabem o que é um roteiro.

Depois das respostas, dizer que um roteiro é a forma como organizamos as ideias que temos para as imagens e sons, que resultarão na história. O roteiro serve ainda como um guia para o trabalho no Scratch, em que indicaremos os locais e as situações que aparecerão na história. No roteiro, a história será contada em cenas.

ROTEIRO:

Agora, você irá colocar a sua história em quadros. Veja o exemplo a seguir:

Título: A Viagem do Caracol Juca

Personagens: Juca (Caracol), Baloca (Balão)

Local: Floresta

Autor: Sonia Mara Souza

PALCO	IMAGEM – Descrever o lugar, o período (dia ou noite), o ambiente (externo ou interno), quem está ali (personagens) e a ação deles (o que fazem).	SOM – Descrever o som de acordo com a cena que será mostrada.
1	Fundo colorido, retângulo com outra cor, com o título da história e o nome do autor (a).	Som de tambor
2	Dia. O Caracol Juca andando feliz num caminho com árvores e pensando: que lindo dia!	Pássaros cantando.
3	Floresta. Juca encontra um balão amarelo, amarrado em um barbante e fica surpreso. Onomatopeia de surpresa, que pode ser: Hã!!	Som de suspense
4	Floresta. Juca segura o barbante do Baloca e juntos saem viajando pelo ar, por cima das árvores.	Sons da natureza
5	Fundo colorido, retângulo com outra cor contrastando com o fundo, no meio do palco, com a frase: “The End” ou “Fim”	

Na análise das interações e aprendizagens, eu e a professora da turma percebemos uma evolução significativa na aprendizagem dos alunos. Comparando a produção escrita entre o início do trabalho com o programa Letramento e Programação em junho e no final, em novembro, verificamos que a escrita se tornou mais detalhada, pormenorizada, sem repetições desnecessárias, pois passaram a usar melhor os recursos da pontuação, articular partes do texto, cuidando da concordância verbal e nominal, e o mais importante, produzindo narrativas com sentido, atendendo à finalidade comunicativa. Com certeza, as atividades de criação de diálogos e histórias no Scratch contribuíram bastante para essas aprendizagens. Percebemos também um avanço na área do pensamento lógico dos alunos, pois passaram a pensar de forma diferente para resolver os problemas, confrontando o seu jeito de resolver os problemas com

os do colega, indo além do saber individual, passando a interagir mais e produzir coletivamente.

A questão do medo de errar também foi algo que foi (re)visto como “tentativas de acertos” e valorizadas ao expor e discutir as ideias para a solução dos problemas, em que os alunos perdiam o medo de serem considerados “burros” pelos colegas e professora, e a percepção de que não há um único caminho para a resolução dos problemas. Enfim, os ganhos com a participação da escola no Programa Letramento e Programação foram muitos. Encerramos em novembro com a expectativa de continuarmos no próximo ano, aprofundando as etapas seguintes para essa turma e inserindo novas turmas.

CAPÍTULO 16

“TROLoló: QUE BARULHO É ESSE?”

Daniel Pereira de Souza

INTRODUÇÃO

Ao primeiro contato, a linguagem de programação pode parecer complicada, contudo a nossa rotina diária segue uma extensa e complicada programação cotidiana, que se torna simples quando passamos a olhar para ela de outra forma, como parceira de vida. O pensamento computacional nos traz uma variedade de conceitos, práticas e perspectivas que, incorporadas ao nosso cotidiano, de forma simples, deixam de ser um “monstro de sete cabeças”. Ao inserir atividades, utilizando o *Scratch* e outras ferramentas, para trabalhar a linguagem de programação por blocos, percebemos o quanto os alunos se interessam pelas atividades e conhecimentos propostos, e que podem estar relacionadas muito além de uma temática específica, mas podem contribuir para a disseminação do pensamento computacional e ser utilizadas como recurso para trabalhar os direitos de aprendizagem dos alunos e os conteúdos distribuídos pelo currículo escolar.

DESCRIÇÃO DA EXPERIÊNCIA

O projeto foi desenvolvido na Escola Municipal São Pedro, localizada bairro da Compensa, na zona oeste da cidade de Manaus/AM. O período de desenvolvimento do projeto foi de cinco meses, entre julho e novembro de 2018. A ideia inicial seria de, apenas, três meses, contudo ele foi estendido para atender às diversas demandas da escola.

A participação direta no projeto contou com quatro turmas (totalizando 116 alunos) e quatro professores, distribuídos nos turnos de funcionamento da escola (matutino e vespertino). O pensar em letramento em programação foi além do que se conhecia acerca da linguagem de programação. Um mundo de possibilidades foi sendo descoberto. Despertar pensadores computacionais, que utilizem conceitos computacionais, práticas e perspectivas em todos os aspectos da vida, seja acadêmica, pessoal ou profissional, é um dos principais benefícios que acreditamos conseguir com a inserção dessa temática na nossa escola.

A partir de uma situação-problema decorrente da poluição sonora na escola, passamos a refletir acerca das suas possíveis consequências para a comunidade escolar. A proposta de tema foi discutida inicialmente com alguns professores e em seguida exposta aos alunos membros do Clube de Linguagem de Programação e Robótica – Procurumim – da escola, que perceberam que o problema era crítico e que poderia ser desenvolvido um trabalho visando a amenizar a questão e tornar possível a construção de um ambiente mais saudável a todos.

O Procurumim é um projeto da Gerência de Tecnologias Educacionais da Secretaria de Educação de Manaus (Semed) e tem como objetivo fomentar o uso de Linguagem de Programação e Robótica Educacional nas escolas, alinhadas ao currículo escolar, como práticas pedagógicas inovadoras, promovendo o letramento digital e as competências e habilidades relacionadas ao pensamento computacional.

O Procurumim na nossa escola conta com 12 alunos de turmas dos turnos matutino e vespertino. Seguindo as orientações do programa, selecionamos quatro professores das turmas nas quais os membros do Procurumim são alunos. Esses professores passaram a participar das formações do Programa Letramento em Programação, desenvolvido em parceria com a Semed, Universidade Federal do Amazonas e Instituto Ayrton Senna. A partir das noções sobre Letramento em Programação, foi possível organizar as atividades a serem desenvolvidas no projeto “Trololó: Que barulho é esse?”. A partir de então, a proposta foi estendida às turmas regulares dos alunos do Procurumim. Os alunos se reconheceram como parte do

problema e perceberam que podiam propor diversas formas para combater a poluição sonora, num engajamento conjunto das quatro turmas, e poderiam estender à escola como um todo.

O projeto foi desenvolvido seguindo uma sequência de atividades pré-estabelecidas: *apresentação do projeto e divisão das equipes de trabalho* – em que apresentamos um esboço do projeto aos alunos das turmas envolvidas, apresentando a justificativa, objetivos e metodologia do projeto, assim como a divisão das equipes para a pesquisa do projeto –; *desenvolvimento da pesquisa e elaboração dos relatórios de pesquisa* – as turmas foram divididas em quatro equipes, em que cada uma se responsabilizou por tarefas distintas: 1) realizou a pesquisa sobre a definição do problema, apontando o que é poluição sonora, quais os padrões considerados normais e sobre a legislação ambiental pertinente ao tema, 2) pesquisou sobre os efeitos da poluição sonora para a saúde auditiva, mental, social e para a aprendizagem, 3) observou e registou a medida do som nas salas, utilizando o aplicativo *Sound Meter*, que simula um decibêlmetro, 4) pesquisou e apontou propostas viáveis para apresentar às turmas para tentar amenizar o problema.

Até então, pode-se perguntar onde que os alunos tiveram contato com o pensamento computacional no decorrer desse processo, pois, aparentemente, parece um projeto normal que não utiliza essa temática. Percorremos o início do projeto sem relacioná-lo, pois queríamos mostrar uma rotina sem o pensamento computacional, para a partir de então mostrar o quão importante e necessário é o tema e a sua inserção na nossa rotina, “passando despercebido”, pela maioria, às vezes. Na sequência, fizemos uma *apresentação sobre pensamento computacional e computação criativa* – em que foi abordado sobre conceitos de pensamento computacional, práticas e perspectivas, relacionando à prática cotidiana não computacional e o quanto o pensamento computacional pode contribuir às nossas diversas atividades, seja na organização, otimização de tempo, reforçando a aprendizagem, entre outras –; *utilização de ferramentas de programação por blocos* – paralelamente ao trabalho de pesquisa, realizamos atividades envolvendo principalmente o *Scratch*, em que possibilitamos aos demais alunos das turmas o primeiro contato com linguagem de programação, desenvolvendo algumas atividades, como

produção de animações –; *montagem dos protótipos de sucata* – essa etapa foi associada com as animações produzidas no *Scratch*, em que foram construídos protótipos de robôs com sucata que seriam utilizados de modelos para os personagens das animações e teriam como função ajudar na amenização do problema da poluição sonora.

A última etapa seria a *apresentação dos protótipos, da pesquisa e as propostas para a problemática* para todas as turmas da escola, contudo só foi possível realizar essa atividade com as turmas envolvidas, entretanto os alunos por equipe apresentaram os resultados dos trabalhos desenvolvidos entre as próprias turmas envolvidas. Em virtude da demanda de atividades no final de ano da escola e a solicitação da secretaria de educação em antecipação do término das atividades nesse ano, a apresentação às demais turmas ficará para o início do ano letivo de 2019, como forma de sensibilização preventiva das consequências da poluição sonora, despertando a solidariedade e colaboração de todos no enfrentamento desse problema.

Os alunos que participam diretamente do projeto demonstram bastante interesse na atividade, visualizando a possibilidade de tornarem-se atores de destaque no enfrentamento dessa problemática na escola. Apesar de reconhecerem-se como parte do problema, a ideia de poder apresentar propostas para solucionar a questão os encoraja a novas atitudes diante do próprio comportamento e dos colegas.

Salientamos que, no decorrer do projeto, procuramos desenvolver algumas competências, como conhecimento, pensamento científico, crítico e criativo, comunicação, argumentação, autogestão, autoconhecimento e autocuidado, empatia e cooperação, autonomia.

O empenho de todos na execução de um projeto é essencial, durante o desenvolvimento desse projeto foi possível ver que a participação foi boa, contudo as demais atividades que emanam da própria rotina escolar tornaram-se os principais obstáculos na sua execução. A partir do cronograma estabelecido, foi necessária uma adequação para que fossem cumpridas as ações necessárias para que o projeto pudesse contemplar todas as atividades propostas. A proposta desse projeto poderá ser estendida para o próximo ano letivo.

Alguns cuidados deverão ser observados nos processos de elaboração e execução do projeto. Destaco dentro das etapas do

projeto alguns pontos. Nas atividades em equipe, deve-se atentar para a questão da participação e colaboração na e da equipe. Alguns grupos costumam centralizar boa parte do trabalho nas mãos de uns. Contudo a intenção deve ser que todos, na equipe, sem exceção, façam parte do desenvolvimento das ações. O desenvolvimento da pesquisa deve ser orientado de forma a direcionar as questões que devem ser pesquisadas, como norte dessa etapa do trabalho. Algumas vezes, será comum desviar o foco da problemática, generalizando demais a questão, deixando a especificidade do problema de lado. Contudo deve-se permitir que o aluno vá além dos pontos norteadores, sem deixá-los de lado. Como uma das intenções das etapas é fortalecer o trabalho em equipe e protagonismo dos alunos, deve ser realizado de preferência sob a supervisão dos professores, não permitindo que o trabalho seja realizado por terceiros, como familiares ou outros. Os alunos devem sentir-se parte do trabalho em todas as suas etapas.

RESULTADOS E REFLEXÕES

O resultado alcançado que mais demonstra a satisfação no desenvolvimento desse projeto é o interesse despertado nos alunos pela linguagem de programação. Além dos horários reservados para as atividades, vimos os alunos, durante os horários disponíveis no laboratório de informática para pesquisas e/ou atividades diversas, utilizando o *Scratch* livremente, interagindo com os colegas e mostrando o que estavam produzindo e descobrindo por meio da ferramenta. Nesse sentido, percebemos que o interesse não forçado, sem a obrigação de utilizar durante uma atividade direcionada, estava promovendo certa autonomia na escolha desse recurso para a aprendizagem. Além do que, alunos de outras turmas, sabendo que o projeto tinha incluído esse tipo de atividade à rotina das turmas que participavam do projeto, demonstraram curiosidade e vontade de que eles também pudessem participar. Esse tipo de *feedback* nos faz querer refletir sobre a necessidade de disponibilizar a todos os alunos da escola uma vivência no letramento em programação.



Figura 1 – Produção de animação no Scratch

Fonte: autor

O resultado do trabalho de pesquisa das turmas envolvidas no projeto será feito no início do ano letivo de 2019, em que serão apresentados em forma de palestra, com apresentação de slides, contendo a definição de poluição sonora, os limites adequados, as consequências, o resultado da medição de som realizado nas salas e a proposta de solução do problema. Esta última seria a grande estratégia a ser utilizada, a partir de então, para amenizarmos o problema. Visto que será apresentada durante a palestra a problemática e suas consequências, a proposta de solução prevê uma observação sistemática do som nas salas durante os tempos de aula, visando a manter uma média considerada normal, não nociva à saúde auditiva, à aprendizagem e ao convívio social.

No gráfico a seguir, está registrada a medição do som feita durante o 2º tempo de aula por três dias, em que foi calculada a média da aferição máxima registrada. Na proposta a ser apresentada às turmas, será feita uma tabela, na qual a turma possa acompanhar os registros de aferição máxima, mínima e média. A partir dessas informações, as turmas poderão acompanhar a poluição sonora que produzem, e tentar amenizar, caso estejam fora dos padrões saudáveis.

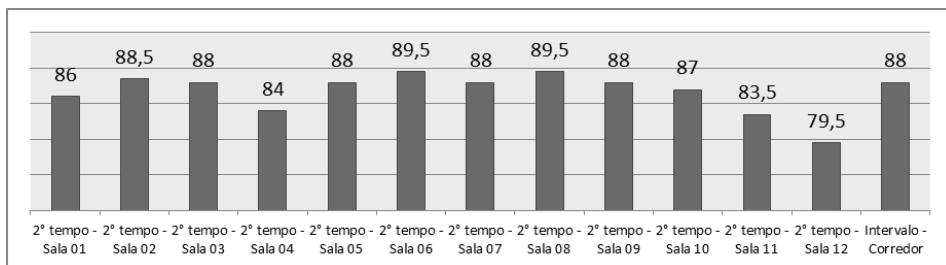


Figura 2 – Registro de medição do som nas salas

Fonte: o autor

Uma importante reflexão realizada no decorrer do projeto é que, realmente, a escola tem sido um ambiente muito barulhento. Levando em consideração que a escola é um ambiente com muitas pessoas, que falam e que se expressam utilizando a voz, conviver com esse som não seria um problema, contudo começou-se a pensar que, talvez, estejamos perdendo a noção do volume do ruído produzido, em virtude de convivermos em ambientes já ruidosos. A saúde auditiva de muitos membros da nossa comunidade escolar já deve ter sofrido danos em virtude dessa convivência, e pelo fato de não se ouvir bem, acaba-se falando mais alto, para se fazer ouvir e ser ouvido, o que pode criar uma “corrente de barulho” em que, para ser ouvido pelo outro num ambiente já barulhento, eu devo falar mais alto. E essa “corrente” acaba agravando ainda mais o problema.

Como a intenção do projeto é ter continuidade no ano próximo ano, serão desenvolvidas ações direcionadas a todas as turmas envolvendo a linguagem de programação, propondo atividades que envolvam computação *desplugada* e pensamento computacional, como forma de fazer os alunos se sentirem mais autônomos, cooperativos, críticos, criativos, entre outras competências. Nesse sentido, vamos utilizar o *Scratch* como forma de enfrentar a poluição sonora, bem como outros possíveis problemas que porventura apareçam, criando animações, histórias e jogos que possam ser um instrumento de sensibilização, até mesmo para outras escolas, ou ambientes que também sofrem com esse problema.

CAPÍTULO 17

TECNOLOGIA, DESENVOLVIMENTO COGNITIVO E AUTOCONHECIMENTO CAMINHANDO JUNTOS, É POSSÍVEL?

Eliana Maria Fattori Calza

Há dois anos, tive a oportunidade de conhecer o projeto “Letramento em Programação” do Instituto Ayrton Senna, cujo trabalho remetia a um projeto maior, uma parceria entre formadores do instituto e professores da rede municipal, que tinha a finalidade de trabalhar com programação de computadores com alunos pequenos, de 9 anos, do 4º ano do ensino fundamental.

Curiosa e interessada pelo assunto, uma vez que acredito na educação integral do indivíduo, participei de formações e fiquei muito impressionada com os relatos apresentados, tomando a iniciativa de desenvolvê-lo com a minha turma, uma vez que a Ciência da Computação é fundamental para qualquer carreira do século XXI, sendo um conhecimento básico que as escolas devem oferecer.

Partindo desse princípio, trabalhei com esse programa nos anos de 2017 e 2018, no entanto, neste relato, me debruçarei sobre o último projeto, desenvolvido no segundo semestre do ano de 2018, intitulado “Eu como comunicador”, utilizando a plataforma do Scratch.

Os participantes foram 17 alunos de 4º ano da Emeb Prof. Luiz Pântano, na cidade de Itatiba, que tiveram o desafio de criar o seu próprio projeto, no Scratch, sob orientação de uma temática “*cyberbullying*”, um assunto tão atual e relevante dentro e fora da escola. Neste relato, será visível um trabalho interdisciplinar, em que as diferentes áreas do conhecimento serão exploradas a partir do uso da tecnologia. Optei por não realizar o projeto em duplas. As crianças poderiam consultar seus colegas que estavam ao lado, pedir sugestões, ajudar, num grande trabalho colaborativo entre a turma.

Inicialmente, realizamos uma atividade “desplugada”, a brincadeira “telefone sem fio”. Essa atividade foi extremamente importante para a minha turma. Como havia tirado licença-prêmio e acabado de voltar, encontrei alguns problemas sérios de convivência da turma, principalmente com relação à situação de comunicação (“fofocas”),

portanto, não poderia haver situação melhor para me ajudar em relação ao convívio da turma e abertura do projeto.

Em um primeiro momento, mesmo a turma conhecendo a brincadeira, tivemos que parar a comunicação logo nos cinco primeiros alunos que, empolgados com a dinâmica, não se atentaram ao conteúdo do texto oral que haviam passado (“certa vez, uma menina que amava seu bichinho de estimação teve que dividi-lo com o seu irmão”), chegando ao quinto aluno somente um zunido “spichi, spichi”. Demos muita risada e começamos novamente, com a mesma frase. Dessa vez, a frase chegou a passar, porém totalmente distorcida! Ao final, demos muita risada e os alunos pediram que fizéssemos mais uma vez, dessa vez em sentido anti-horário. Iniciamos novamente, com a frase: “o menino é muito grato, ama o seu pai”. Ao final, a frase se tornou “o menino ama pera”. Rimos muito e tentamos observar em que lugar aconteceu a falha, a partir daí, discutimos sobre a falha que houve, que não foi intencional, mas que por diferentes motivos ela pode ocorrer. Portanto, chegamos à conclusão de que as falhas de comunicação são muito fáceis de acontecer e, muitas vezes, elas nem acontecem por maldade, mas por não serem ouvidas direito, por algum barulho externo, por entendimento equivocado da pessoa... Os alunos apresentaram diversas situações em que vivenciaram momentos assim e conseguiram retomar o problema que estávamos vivendo em sala de aula, inclusive, as meninas que estavam “com esse problema” se abraçaram e pediram desculpas uma à outra.

Na semana seguinte, retomamos os aprendizados da brincadeira, ressaltando a responsabilidade de cada um na comunicação de informações, e problematizei com a turma: como nos comunicamos com as tecnologias digitais (computador, tablet, celular etc.)? Em que espaços nos comunicamos (exemplos: jogos, redes sociais)? Como será que essas tecnologias entendem as informações que compartilhamos? Os alunos deram suas opiniões e eu e o monitor Diógenes Braga, meu parceiro de trabalho nesse programa, apresentamos o conceito de Código Binário.

Foi um momento muito importante, em que os alunos demonstraram interesse e curiosidade sobre como a máquina entende os comandos dados pelo ser humano. Era visível observar seus olhinhos curiosos e atentos a cada explicação, desvendando um

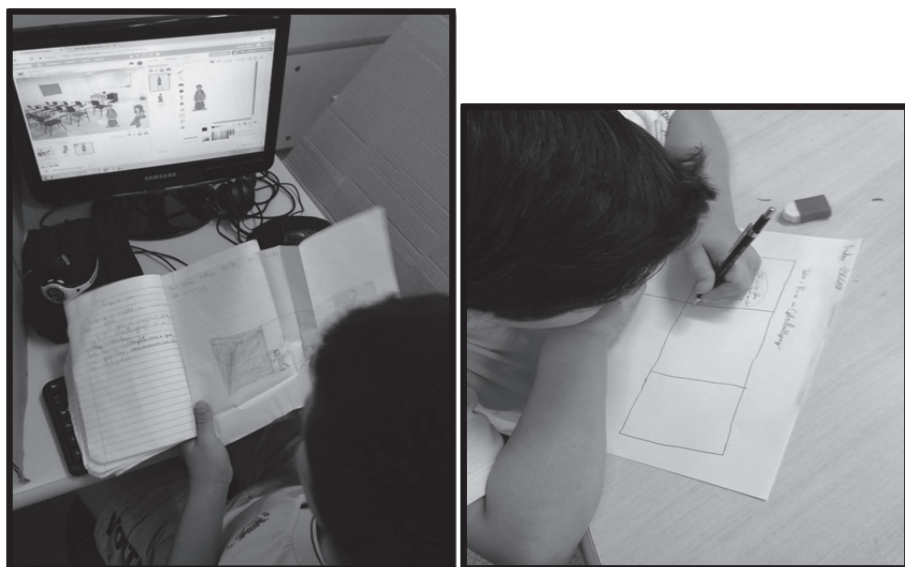
grande mistério. Depois de uma conversa inicial, os alunos foram convidados a pensar na linguagem binária como os botões de “0” e “1”, enfileirados, alguns representaram cada número e desvendamos a palavra Luiz (nome da nossa escola).

Após esse momento, entregamos o alfabeto em código para que os alunos construíssem uma mensagem aos colegas. Inicialmente, sentiram muita dificuldade, eu e Diógenes precisamos auxiliar pontualmente, porém, após pequenas intervenções, as crianças foram elaborando suas mensagens e depois trocaram com os colegas. Desvendar os códigos foi algo divertido e animado, que gostaram de realizar, fazendo-os refletir sobre essa nova linguagem e exigindo atenção e concentração dos alunos. Como tarefa dessa aula, em casa, os alunos teriam que realizar uma pesquisa sobre cyberbullying.

Na terceira semana, os alunos fizeram a apresentação das pesquisas e conversamos sobre cyberbullying. Foi um momento relevante para o início desse novo projeto. Os alunos se envolveram em explicar e também em falar que já sofreram com o bullying na escola. Houve várias colocações importantes sobre algumas situações que acontecem no dia a dia e que muitas vezes ferem alguns, um dos alunos até mesmo se emocionou. Com relação ao crime virtual, alguns contaram que pessoas próximas a eles já se envolveram e sofreram com notícias falsas ou com divulgações que os deixaram tristes e feridos. Também conversamos sobre as “fake news” e o quanto elas podem prejudicar as pessoas. Depois disso, fomos à biblioteca e os alunos tiveram a tarefa de produzir uma história em quadrinhos sobre o *cyberbullying*. Essa História em Quadrinhos seria um plano para o projeto que se desenvolveria na plataforma Scratch, nas aulas seguintes.

Na aula seguinte, os alunos iniciaram suas construções na plataforma Scratch. Gostei muito de ter a HQ no papel, isso me ajudou e ajudou as crianças a manterem o foco. Muitas vezes, passei pelos computadores e vi os alunos procurando no site de pesquisas imagens que não coincidiam com a sua HQ, então, eu solicitava que eles olhassem para a própria história e retomassem o ambiente e os personagens que deveriam utilizar; automaticamente, o aluno dizia: “nossa, é mesmo!”. Essa aula foi mais “cansativa” que as outras, pois os alunos tiveram que relembrar os comandos utilizados no semestre anterior e colocá-los em prática. Porém foi uma aula bastante produtiva.

As aulas que se seguiram foram de elaboração do projeto, em que os alunos aprenderam a utilizar recursos como “mensagem” para trocar os personagens e palcos.



Figuras 2 e 3: Alunos do 4º ano da EMEB Prof. Luiz Pântano planejando a história em quadrinhos no papel e executando o plano no Scratch

Fonte: da autora

Na semana seguinte, notei que alguns alunos estavam muito atrasados com o projeto, ainda colocando o plano de fundo, os personagens e os diálogos. Portanto, retomei esses aspectos e deixei a aula dessa semana para colocarmos em ordem esses aspectos, para isso, os alunos fizeram um “checklist” do que já haviam colocado, o que estavam com dificuldade e o que ainda faltava. Nesse momento, o trabalho com autoconhecimento e a autorregulação ficaram muito claros, além, é claro, do trabalho com a escrita (produção de texto e revisão ortográfica) e com a Matemática (projeção dos personagens nos palcos e análise dos problemas que precisavam resolver). Os alunos que foram finalizando chamavam-me para verificação. Ao passar, notei que, mesmo dizendo que tinham terminado, faltavam encaminhamentos importantes que pedi para rever. Também fiz a revisão ortográfica. Dos 17 alunos, somente duas terminaram essa

etapa. Solicitei, então, para as alunas que já haviam finalizado, que se tornassem monitoras dos colegas com mais dificuldades, auxiliando-os no processo.



Figuras 4 e 5: Projetos dos alunos do 4º ano da EMEB Prof. Luiz Pântano no Scratch
Fonte: da autora

Como próxima etapa do trabalho, os estudantes avaliaram seus projetos quanto ao tema do *cyberbullying* e quanto à organização no Scratch, apresentando o que já haviam feito aos colegas. Esse foi um momento muito importante para os alunos e para mim, pois as crianças puderam checar o que já tinham feito, observar o que o colega já tinha programado e planejar, em seu projeto, o que ainda iria acrescentar.



Figura 6: alunos do 4º ano fazendo a primeira mostra do trabalho aos colegas de sala na EMEB Luiz Pântano

Fonte: da autora.



Figura 7: alunos do 4º ano da EMEB Luiz Pântano trocando informações para melhorar ainda mais o projeto

Fonte: da autora.

Após essa primeira avaliação pública, os estudantes continuaram a construção de seus projetos, pensando no que ainda precisa-

vam alterar, revendo os detalhes de seus projetos. Esse momento foi muito importante, pois falavam: “professora, quero colocar meu personagem fazendo determinada ação, como a amiga(o) colocou”, então, chamávamos a(o) colega, que colaborava, ensinando como deveria acrescentar o comando para a determinada ação. Foi um momento rico, de valorização do trabalho dos colegas e de colaboração de todo o grupo.



Figura 8: Apresentação no pátio da EMEB Luiz Pântano para os colegas de outras salas.

Fonte: da autora.

Na aula seguinte, chegando à reta final das produções, os alunos fizeram a apresentação para seus próprios colegas, fechando detalhes e treinando para o próximo encontro, que seria a apresentação para os 3^{os} anos, a fim de mostrar e incentivar os alunos a participarem do projeto no próximo ano.

A próxima aula chegou, era o dia tão esperado! Os alunos ficaram ansiosíssimos para esse momento, pois teriam que ficar à frente de todos os colegas de sala e também de todos os alunos do 3º ano. Combinamos que cada um mexeria sozinho no computador e falaria, no microfone, seu nome e como era o seu projeto, entrando no seu Scratch e apresentando passo a passo o que fez. Alguns “suavam frio”, preocupados com tamanha responsabilidade. Foi um momento rico, os alunos convidados gostaram muito e, segundo relatos posteriores das professoras, disseram que queriam participar desse projeto. Ao final, nossa turma ficou animadíssima, parabenizei-os pelo desempenho dos projetos e das apresentações, receberam palmas de todos e uma fala especial da vice-diretora, que acompanhou esse momento.

Valeu muito a pena! Um trabalho que fechou com chave de ouro!

CAPÍTULO 18

A DESAFIANTE E ENCANTADORA MISSÃO CHAMADA LETRAMENTO EM PROGRAMAÇÃO

Nariana de Oliveira Tavares

INTRODUÇÃO

A experiência aqui relatada busca contar de forma breve como se deu o processo de formação de professores e a implantação do programa Letramento em Programação em uma das escolas de tempo integral, atendida pelo programa, no município de Caruaru-PE. Relataremos como essa experiência desafiou e encantou os professores envolvidos e como despertou nos estudantes curiosidade, motivação e engajamento. Mesmo diante do analfabetismo e da pouca desenvoltura de alguns em relação à utilização de ferramentas tecnológicas, sentir-se desafiado foi um incentivo para os resultados adquiridos. Nessa escola, alunos e professores venceram seus medos e em pouco mais de seis meses desenvolveram projetos que participaram de eventos importantes na cidade.

DESCRIÇÃO DA EXPERIÊNCIA

Letrar, ensinar, adaptar-se são palavras simples com significados extremamente importantes no meio educacional. E, tratando-se de Letramento em Programação, tornam-se indispensáveis e nos mostram um mundo de possibilidades e encantos. Transformar estudantes por meio da educação nunca foi fácil, mas, com a evolução educacional e a crescente utilização da tecnologia, tem sido um fardo mais leve. E encontrar dentro das salas de aulas estudantes que nunca tiveram acesso à tecnologia ainda é possível.

Na Escola de Tempo Integral Álvaro Lins, localizada na Rua Tropical, s/n, bairro Maurício de Nassau, Caruaru - PE, não foi dife-

rente com o Letramento em Programação, iniciado em abril de 2018, tendo 40 estudantes, divididos em duas turmas, com alunos do 4º e 5º anos. Para que a implantação do projeto fosse possível, os(as) professores(as) passaram a receber formações mensais com duração de oito horas presenciais, além de momentos virtuais. Os encontros presenciais contam com momentos enriquecedores de formação, experiências, encontros de aprimoramentos e, claro, encorajamento para realizarmos nossas atividades.

Assim que o programa nos foi apresentado, tive um susto, pois tudo que é novo assusta um pouco, mas não parei naquela dificuldade, vieram as dúvidas e o medo de não conseguir repassar para os estudantes, a cada encontro de formação uma alegria nova por conseguir assimilar melhor e explorar o programa Scratch e suas programações. O que era medo tornou-se ‘divertimento’, alegria, orgulho de não ter desistido na primeira dificuldade, ter superado o que me atormentava e ter transformado em aprendizado que seria repassado para os estudantes de forma tão simples, porém com um significado tão grande.

Foi fácil identificar em alguns estudantes que ali era a primeira vez que ele “pegava” em um notebook, percebendo o medo por estar diante de uma máquina, aquele olhar sincero e a voz um pouco rouca e trêmula por medo de “quebrar”, a vergonha de não saber “mexer”. Esses sentimentos são aos poucos modificados pela força de vontade em querer aprender, mostrar que pode e irá realizar o que foi proposto.

Nossas aulas semanais com duração de duas horas-aula sempre foram “as aulas mais rápidas”, pois iniciávamos uma atividade e, de repente, já era hora de guardar as máquinas. Atividades estas sempre focando na realidade de que alguns nunca haviam manuseado um notebook. No começo, foi um pouco difícil passar a linguagem cibernética para eles, mas, aos poucos, eles também foram adaptando-se à nossa linguagem e dominando alguns controles. Algumas vezes, era necessário parar uma explicação sobre o comando, porque o aluno(a) não encontrava a tecla que era necessária, ou a dificuldade na leitura, momentos que foram libertadores para aqueles que naquele momento puderam tirar sua dúvida e aprender.

Nossas aulas, sempre planejadas, seguiam os projetos sugeridos para o módulo 1 do Letramento em Programação, em que eles aprenderam primeiro o significado do que era programação, como programar, qual era o significado da programação hoje em dia na educação, e aos poucos os conteúdos foram sendo introduzidos, sempre intercalado com outras disciplinas, como por exemplo, o gênero textual fábula, que era estudado em sala de aula era transformado em animação por eles. Aquela dificuldade em matemática era transformada em algum diálogo, animação, programações simples, mas com um grandioso significado para eles. No fim das aulas, o diário de bordo, que antes era lembrado e exigido por nós, era algo lembrado e realizado por eles, explicando tudo o que aprenderam naquele dia, e como se sentiam após as aulas.

A alegria deles é também a nossa! Ao deparar-se com uma programação que eles fizeram sozinhos, vemos que alegria toma conta de seus rostos, o sentimento de gratidão cresce e o entusiasmo deles também, pois o medo foi superado e deu lugar à realização. Deparei-me com um dos estudantes (Paulo, 5º ano) que tinha grande dificuldade de leitura e escrita. O que mais me impressionou foi ele não ter se limitado a essa dificuldade, a cada encontro, uma nova descoberta, e de repente sua dificuldade foi superada, aquele aluno, que antes não conseguia fazer uma simples leitura, foi conquistado pelo letramento e estava programando sozinho, lendo e escrevendo pequenos textos.

Ele superou também outra coisa que o prejudicava nas aulas, a timidez. Por ser tímido, ele tinha receio de pedir ajuda, de falar que não entendeu, de sinalizar que precisava de ajuda. Mas ele entendeu que o Letramento abria um leque de conhecimentos e um mundo encantador da tecnologia. Sua timidez foi superada, em uma das nossas melhores experiências dentro da escola. Em uma apresentação para os demais estudantes, foi o protagonista daquele momento e começou a mostrar e ensinar o programa, revelando aos demais o quanto era fácil e a quantidade de possibilidades que tinha para programar. Não só esse aluno, como tantos outros sentiram medo e curiosidades, que ao longo dos anos foram transformadas em aprendizados e metas cumpridas.



Figura 1 – Estudante Paulo, programando uma fábula na Feira Nacional do Livro do Agreste (Fenagreste)

Fonte: da autora

Essas dificuldades também foram sentidas no ambiente físico, por não termos salas adequadas, temos um laboratório móvel com 40 notebooks. Nos momentos das aulas, temos que sair da sala para buscar as máquinas, ao término, temos que nos deslocar novamente para guardar e organizá-los.

Mesmo com toda e qualquer dificuldade, somos os únicos professores que temos lista de espera para entrar no Letramento, devido à limitação de turmas imposta esse ano pelo Instituto de Corresponsabilidade pela Educação (ICE). Muitos alunos queriam ter participado, ficaram tristes por não poder em função da limitação das turmas, mas tiveram uma grande alegria em saber que próximo

ano poderão fazer parte, pois a disciplina passará a atender todos os alunos de todas as turmas de 4º, 5º, 6º e 7º anos. Em 2019, eles terão a oportunidade de aprender a programar.



Figura 2 – Uma de nossas aulas – a alegria no rostinho deles é bem perceptível – Prof.ª Nariana Tavares e Prof.ª Valéria Santos

Fonte: da autora

RESULTADOS E REFLEXÕES

O Letramento em Programação vem quebrando barreiras e possibilitando o acesso à tecnologia. Com as aulas, é possível ver o brilho no olhar dos estudantes, a curiosidade, o medo, que também fazem parte, pois alguns nunca tiveram acesso a nenhum tipo de mecanismo tecnológico. E, ao deparar-se com notebooks, Datashow, slides, atividades desplugadas, a vontade de aprender torna-se mais encantadora e o tal ‘mundo digital’ fica bem próximo, sem precisar de

muito. A compreensão pela programação vai sendo construída, e em pouco tempo é possível perceber as modificações, tanto nos alunos quanto em nós, que também vamos nos apaixonando e querendo cada vez mais adentrar nesse mundo de programação, criação, do fazer e dar a possibilidade de realmente letrar para o futuro.

Esse engajamento de professores e alunos possibilitou a participação dos alunos em três importantes eventos: Oxente Day, Fenagreste e TechOscar. Na Fenagreste, foram desenvolvidos projetos baseados em fábulas infantis, e para o TechOscar foram desenvolvidos projetos voltados para a sustentabilidade dentro da escola.

SOBRE OS AUTORES

Adriano Ferreti Borgatto

Possui graduação em Estatística pela UNESP, mestrado em Estatística e Experimentação Agropecuária pela UFLA, doutorado em Agronomia (Estatística e Experimentação Agronômica) pela ESALQ/USP. Atualmente, é professor associado da Universidade Federal de Santa Catarina. Sua principal área de pesquisa é a Teoria da Resposta ao Item, atuando principalmente em temas como estilo de vida, qualidade de vida e ensino.

Número do Orcid: 0000-0001-6280-2525

Ana Grasielle Dionisio Correa

Possui graduação em Engenharia da Computação pela Universidade Católica Dom Bosco (2002), mestrado em Engenharia Elétrica pela Escola Politécnica da Universidade de São Paulo (2005) e doutorado em Engenharia Elétrica pela Universidade de São Paulo (2011). Atualmente é professora assistente doutor I da Universidade Presbiteriana Mackenzie (UPM), professora colaboradora do Programa de Pós-Graduação em Distúrbios do Desenvolvimento da UPM e pesquisadora colaboradora do Centro Interdisciplinar de Tecnologias Interativas da Escola Politécnica da USP (CITI-USP). É membro do Comitê de Ética em Pesquisa da Universidade Presbiteriana Mackenzie. Atua principalmente nos seguintes temas: reabilitação virtual, realidade virtual e aumentada em saúde e educação e tecnologias assistivas.

Número do Orcid: 0000-0002-0393-8710

Ana de Oliveira Rodrigues

Possui graduação em Engenharia Elétrica pela Pontifícia Universidade Católica de Minas Gerais (1999), doutorado em Engenharia Elétrica pela Universidade Federal de Minas Gerais (2003), com Bolsa Sanduíche no Imperial College London (abril/2001 a março/2002) e pós-doutorado pela Universidade Federal de Minas Gerais (2011). Foi professora do IBMEC MG (2012-2013), do Centro Universitário UNA (2010-2012), do Centro Universitário Newton Paiva (2011-2011), da PUC Minas (2007-2008), da Faculdade Pitágoras (2008-2008) e do Centro Universitário de Belo Horizonte UNI-BH (2004-2009). Foi destaque acadêmico

durante o curso de Engenharia Elétrica da PUC MG por sete semestres consecutivos, do 2º/1995 ao 1º/1998. Foi premiada com o primeiro lugar no Student Paper Contest em Radiofrequency and Microwaves, no IMOC2003. Foi escolhida professora homenageada dos formandos em Engenharia de Telecomunicações do UNI-BH no 2º semestre de 2005 e professora paraninfa dos formandos em Engenharia de Telecomunicações do UNI-BH no 2º semestre de 2007. É autora de seis artigos completos publicados em periódico, 24 trabalhos publicados em anais de eventos e um capítulo de livro, sendo revisora do periódico Compel. Co-orientou uma tese de mestrado (2007) e uma tese de doutorado (em andamento) no departamento de Engenharia Elétrica da UFMG. Orientou 13 trabalhos de conclusão de curso de graduação na UNI-BH (2004-2008). É sócia-fundadora da Dedutiva Consultoria em Software, onde é consultora desde 2007. Foi diretora de Produtos da Zunnit Sistemas Inteligentes, startup que desenvolve Sistemas de Inteligência Artificial durante 2014 e 2015. Atualmente é sócia fundadora da Just CODING (www.justcoding.com.br), que oferece cursos de programação para crianças, jovens, adultos e terceira idade.

Instituição que representa: Just CODING

Ana Márcia Abreu Martins de Paiva

Atua na área de Educação há mais de 25 anos, como professora dos diversos níveis de ensino (do fundamental à pós-graduação), na formação de professores e em consultoria educacional na área de Ensino-Aprendizagem, empreendedorismo para crianças, gestão do conhecimento, criação do conhecimento e democratização da informática. Mestre em Gestão Social, Educação e Desenvolvimento Local. Especializações: Educação, Design Instrucional e Gerência da Tecnologia da Informação. Autora de materiais didáticos e disciplinas "Intercâmbio de Saberes entre professores e alunos", "Design Thinking para Educadores" e "Cultura Maker" nos cursos de Educação Criativa e Metodologias Ativas da Educação da PUC Minas. Autora da disciplina Inovação e Design Thinking para os cursos do Instituto de Gestão de Tecnologia da Informação/MG, para alunos de pós-graduação da área de TI. É consultora na área de Design Thinking para Educadores pelo Instituto Educadigital – SP. Recebeu em 2014 o Prêmio Cultura Empreendedora pelo Sebrae, com o Projeto Seminário de Novas Tec-

nologias, realizado na Fundação Torino. Professora de Programação para crianças e jovens. Professora de pós-graduação na área de Educação Criativa. Educadora, estudiosa e defensora da Cultura Maker. Instituição que representa: Fundação Torino

Anelise Lemke Kologeski

É professora na área de Informática do ensino básico técnico e tecnológico no Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS), do Campus Osório. Possui Formação de Professores para a Educação Profissional (UFSM, 2016), é mestre em Microeletrônica (PGMICRO/UFRGS, 2011) e é graduada no curso de Engenharia em Sistemas Digitais (UERGS, 2008). Tem experiência na área de integração 3D e tolerância a falhas em sistemas digitais e redes intra-chip, e atualmente pesquisa sobre o uso da Informática na Educação, com atividades voltadas ao pensamento computacional a partir do raciocínio lógico e da programação básica, por meio de oficinas lúdicas com plataformas digitais gratuitas, participando e orientando projetos de extensão e pesquisa financiados pelo fomento interno do IFRS e pelo CNPq.

Email: anelise.kologeski@osorio.ifrs.edu.br

Número do Orcid: 0000-0001-9257-8915

Annette Vee

Focuses her research and teaching on literacy, composition and technology and she is an Assistant Professor in the Composition area. She received her PhD in Composition & Rhetoric and her MA in Literature at the University of Wisconsin-Madison Department of English. Her research explores digital intellectual property debates as well as intersections between alphabetic and code writing in historical context. She has published on digital games, writing center technology and intellectual property issues in software composition. Her current book project argues that the historical and social perspective of New Literacy Studies can help us understand the technical and social affordances of computer code writing. She currently serves on the CCCC's Intellectual Property Committee and Pitt's Curriculum Committee for Composition.

Augusto Weiand

IFRS – Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul, Campus Osório;

UFRGS – Universidade Federal do Rio Grande do Sul.

Atualmente é estudante de doutorado em Informática na Educação (PPGIE / UFRGS) e atua como técnico em Tecnologia da Informação no Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul – IFRS, Campus Osório. É mestre em Ciência da Computação (PUCRS, 2016), pós-graduado em Tecnologias da Informação e Comunicação na Educação (FURG, 2018), graduado em Informática (FACOS, 2013) e técnico em Informática na Internet (IFRS, 2013). Tem experiência na área de Desenvolvimento e Integração de Sistemas, Data Mining, Interação Humano-Computador e Visualização de Dados. Também participa e desenvolve projetos de pesquisa financiados por agências de fomento como CNPq e IFRS

E-mail: guto.weiand@gmail.com

Número do Orcid: 0000-0003-0117-1613

Bruno da Silva Rodrigues

Possui graduação em Engenharia da Computação pela Universidade Santa Cecília – Santos (2003), mestrado (2006) e doutorado (2011) em Engenharia Elétrica com ênfase em microeletrônica pela Escola Politécnica da Universidade de São Paulo – EPUSP e doutorado no Instituto de Eletrônica e Telecomunicações de Rennes (IETR) pela Universidade de Rennes 1 – França (2011). Com experiência como coordenador e professor do Centro Universitário Sant'Anna (2011), atualmente é professor e pesquisador da Universidade Nove de Julho (Uninove) na área de automação, professor da Universidade Presbiteriana Mackenzie e da Faculdade Impacta de tecnologia. Tem experiência na área de Engenharia Elétrica, com ênfase em Materiais e Componentes Semicondutores atuando principalmente na área de corrosão de silício por plasma, deposição de filmes finos e caracterização de sensores, SGFET, smartsensor, automação, microcontroladores e programação.

Número do Orcid: 0000-0001-9567-4910

Christiane A. Gresse von Wangenheim

PMP é professora do Departamento de Informática e Estatística (INE) da Universidade Federal de Santa Catarina (UFSC) onde coordena o GQS - Grupo de Qualidade de Software e a iniciativa Computação na Escola do INCoD - Instituto Nacional de Ciência e Tecnologia para Convergência Digital. Seus principais interesses de pesquisa são na área de qualidade de software, ensino de engenharia de software e o ensino/popularização da computação na Educação Básica. Ela recebeu o título de Dr. rer. nat. pelo Departamento de Informática da Universidade de Kaiserslautern (Alemanha).

Número do Orcid: 0000-0002-6566-1606

Curriculum Lattes: lattes.cnpq.br/3879944876244096

Clodis Boscarioli

Doutor em Engenharia Elétrica pela Universidade de São Paulo, é professor associado na Universidade Estadual do Oeste do Paraná, campus de Cascavel, onde atua desde o ano 2000 no Curso de Ciência da Computação. Professor permanente no Programa de Pós-Graduação em Educação em Ciências e Educação Matemática – nível de mestrado e doutorado e no Programa de Pós-Graduação em Ciência da Computação – nível de mestrado, ambos no campus de Cascavel, e no Programa de Pós-Graduação em Ensino – nível de mestrado e também professor colaborador no Programa de Pós-Graduação em Tecnologias, Gestão e Sustentabilidade – nível de mestrado profissional, esses últimos no campus de Foz do Iguaçu. Suas áreas de interesse envolvem, de forma multidisciplinar, Banco de Dados, Interação Humano-computador, Aprendizado Computacional, Data Mining, Sistemas de Informação e Tecnologias [Assistivas] no Processo de Ensino-Aprendizagem, além de questões relacionadas ao Ensino de Computação. Líder do GIA (Grupo de Pesquisa em Inteligência Aplicada) e do GTIE (Grupo de Pesquisa em Tecnologia, Inovação e Ensino) da Unioeste.

Número do Orcid: 0000-0002-7110-2026

Dalton Dario Serey Guerrero

Instituição: UFCG. Graduado em Ciência da Computação pela Universidade Federal da Paraíba (1996), mestre em Informática pela Universidade Federal da Paraíba (1997) e doutor em Engenharia Elétrica pela Universidade Federal de Campina Grande (2002). Dalton Serey é professor adjunto da Universidade Federal de Campina Grande e coordenador do Curso de Graduação em Ciência da Computação da UFCG. Sua área de pesquisa é engenharia de software e seus temas de interesse são: I) evolução e manutenção de software, II) verificação, validação e testes e III) educação e computação.

Daniel Pereira de Souza

Formado em Magistério em Nível Médio, licenciado em História e especialista em Psicopedagogia, é professor da Rede Municipal de ensino de Manaus há 14 anos. Lecionou na educação infantil, ensino fundamental anos iniciais e anos finais e no Atendimento Educacional Especializado, coordenou programas de ações complementares na escola, como o Programa de Saúde Escolar e Programa Mais Educação, além de projetos relacionados à temática de Cidadania, Educação para o Trânsito e Tecnologias Educacionais, hoje é coordenador de Telecentro Escolar, em que desenvolve atividades relacionadas ao uso de tecnologias na escola. Secretaria Municipal de Educação de Manaus.

Diego Fernando Grzibowski

É Mestre em Desenvolvimento e Políticas públicas pela UFFS, Campus de Cerro Largo/RS (2019), Especialista em Gestão Pública Municipal pela UFSM (2018), Especialista em Informática Aplicada a Educação pela Uri/Erechim (2012) e Tecnólogo em Gestão da Tecnologia da Informação pela Faculdade Faplan/Anhanguera (2010), Atualmente exerce o cargo de Agente Administrativo no Departamento de Recursos Humanos da Prefeitura Municipal de Marau/RS.

Eduardo Mauricio Moreno Pinto

Entusiasta e adepto de projetos e de pesquisas acadêmicas na área da Educação Matemática e da Tecnologia Educacional direcionados

ao ensino fundamental. Formação Continuada: licenciatura pedagógica em andamento, pela Univesp. Formação acadêmica: mestrado no Programa de Pós-Graduação de Ensino de Ciências e Matemática (Pecim) pela Universidade Estadual de Campinas. Bacharel em Matemática Aplicada e Computacional com Ênfase em Pesquisa Operacional (2010) e graduação em Licenciatura em Matemática (2008), ambos, pela Universidade Estadual de Campinas – Unicamp. Experiência profissional atual: Formador da Central de Formação e Mídias (Ceformi) da Secretaria de Educação da Rede Municipal de Vinhedo. Experiências profissionais anteriores: orientador de área – Matemática na Rede Municipal de Vinhedo; professor efetivo de Matemática da Rede Municipal de Vinhedo; tutor no Núcleo de Informática Aplicada a Educação. Instituição: Secretaria de Educação da Rede Municipal de Vinhedo.

Profa. Eliana Maria Fattori Calza

Possui graduação em Letras – Língua Portuguesa pela Universidade São Francisco (2002). Atualmente é Assessora Técnico Pedagógica da Prefeitura Municipal de Itatiba. Tem experiência na área de Letras, com ênfase em Língua Portuguesa.

Eliei Constantino da Silva

Licenciado e bacharel em Matemática pela Universidade Estadual Paulista “Júlio de Mesquita Filho”, Brasil, e Universidade do Minho, Portugal, respectivamente. Mestre em Educação Matemática pela Universidade Estadual Paulista “Júlio de Mesquita Filho”. Membro do Grupo de Pesquisa em Informática, outras Mídias e Educação Matemática (GPIMEM) e membro do Grupo de Pesquisa Ensino e Aprendizagem como Objeto da Formação de Professores. Atuou como professor bolsista do Departamento de Educação Matemática do Instituto de Geociências e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”. Tem interesse e desenvolve pesquisas nos seguintes temas: Educação Matemática, Pensamento Computacional, Robótica, Programação Computacional, Tecnologias Digitais na Educação, Ensino e Aprendizagem, Teoria Histórico-Cultural e Formação de Professores. Atualmente é douto-

rando em Educação Matemática pela Universidade Estadual Paulista “Júlio de Mesquita Filho”, editor de conteúdo da Geekie, colunista do InfoGeekie, membro do Comitê Técnico Científico da Atena Editora e desenvolve ações de formação de professores relacionadas ao uso de tecnologias e pensamento computacional na Educação.

Número do Orcid: 0000-0003-3555-791X

Érica Luiza de Oliveira

Instrutora de Informática na Prefeitura Municipal de Tapejara – Rio Grande do Sul. Graduada em Análise e Desenvolvimento de Sistemas pelo Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul, Campus Sertão e graduanda em Formação Pedagógica de Docentes para a Educação Básica e Profissional no Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul, Campus Sertão.

Giani Petri

É professor na Universidade Federal de Santa Maria (UFSM), lotado no Colégio Politécnico da UFSM. Possui doutorado em Ciência da Computação pela Universidade Federal de Santa Catarina (UFSC), mestrado em Ciência da Computação pela Universidade Federal de Santa Maria (UFSM) e bacharelado em Sistemas de Informação pela Universidade Luterana do Brasil (ULBRA). Tem experiência na área de análise, projeto e desenvolvimento de sistemas, atuando principalmente nos seguintes temas de pesquisa: ensino de computação, jogos educacionais e engenharia de software.

Número do Orcid: 0000-0002-9884-8151

Irani Bernadete Roani

SME Secretaria Municipal de Educação – Núcleo de Inovações Educacionais – Passo Fundo. Especialista em Psicopedagogia Clínica e Institucional, Especialista em Linguagens e Tecnologias na Educação, Arte educadora. Professora Responsável pela formação em informática educativa na rede municipal; coordenadora pedagógica e

professora no/do Projeto Letramento em Programação Norte Gaúcho na rede municipal; líder GEG PF; instrutora Nuvem Mestra/Google.

Profa Dra. Ivânia Campigotto Aquino

Possui graduação em Curso de Letras pela Universidade de Passo Fundo (1990), mestrado em Teoria da Literatura pela Pontifícia Universidade Católica do Rio Grande do Sul (1999) e doutorado em Letras pela Universidade Federal do Rio Grande do Sul (2007). Atualmente é professor Titular III da Universidade de Passo Fundo e Coordenador da Coordenadoria de Inovações Educacionais da Secretaria Municipal de Educação de Passo Fundo. Tem experiência na área de Letras, com ênfase em Letras, atuando principalmente nos seguintes temas: literatura, romance, história, leitura e aprendizagem.

Leila Ribeiro

Instituição: UFRGS. Professora titular do Instituto de Informática (INF) da Universidade Federal do Rio Grande do Sul (UFRGS). Diretora de Ensino de Computação na Educação Básica da Sociedade Brasileira de Computação (SBC). É membro do IFIP Technical Committee 1 (Foundations of Computer Science) e membro efetivo da SBC. Doutora em Computação pela Universidade Técnica de Berlim/Alemanha (1996). Atua nas áreas de Fundamentos da Computação e Educação em Computação, principalmente em Modelos de Computação e Pensamento Computacional.

Número do Orcid: 0000-0001-7039-092X.

Lisandro Zambenedetti Granville

Instituição: UFRGS; Professor do Instituto de Informática (INF) da Universidade Federal do Rio Grande do Sul (UFRGS). Membro do Conselho da Sociedade Brasileira de Computação (SBC). Diretor de Centro de Pesquisa e Desenvolvimento em Tecnologias Digitais para Informação e Comunicação (CTIC) da Rede Nacional de Ensino e Pesquisa (RNP)..

Número do Orcid: 0000-0001-8956-8660.

Márcia Lucia Andrade dos Anjos Naves

Diretora geral da Fundação Torino e Superintendente do Isvor. Graduação na Pontifícia Universidade Católica de Minas Gerais em Ciências Econômicas, Economia (1987). Master of Business Administration (M.B.A.), e-business, na Fundação Getúlio Vargas – FGV (1986). Pós-graduação em Informática Aplicada à Educação na Pontifícia Universidade Católica de Minas Gerais (2014). Coach certified by Instituto Ecosocial – ICF, Coaching (2014). Especializações: Implantação e Gestão de Universidade Corporativa. Desenvolvimento, implantação e gestão de e-learning. Introdução de Design Thinking em planejamentos estratégicos e modelo educacional. Instituição que representa: Fundação Torino

Márcia Regina Kaminski

Doutoranda em Educação em Ciências e Educação Matemática na Universidade Estadual do Oeste do Paraná, campus de Cascavel. Instrutora de Informática desde 2004 no município de Cascavel, atuando na Escola Municipal Aloys João Mann. Suas áreas de interesse envolvem o uso de Jogos Digitais e Ambientes Virtuais para o ensino e aprendizagem, o desenvolvimento do Pensamento Computacional, o Ensino da Matemática, Educomunicação e Sustentabilidade. Membro do GTIE (Grupo de Pesquisa em Tecnologia, Inovação e Ensino). Número do Orcid: 0000-0001-5705-0322

Maria Teresa Zampieri

Licenciada em Matemática pela Universidade Federal de São Carlos (2003), mestre em Educação Matemática pela Universidade Estadual Paulista “Júlio de Mesquita Filho” (2013) e doutora em Educação Matemática por essa mesma instituição (2018). Pós-doutora pelo Programa Nacional de Pós-Doutorado (PNPD/CAPES), vinculada ao Programa de Pós-Graduação em Educação Matemática da UNESP, campus de Rio Claro. É pesquisadora associada ao Grupo de Pesquisa em Informática, outras Mídias e Educação Matemática (GPIMEM). No doutorado, sua pesquisa foi financiada pela FAPESP, processo #2014/27166-9. Como parte das atividades do doutorado, realizou um estágio no exterior por um período de dois meses, também finan-

ciado pela FAPESP, processo #2015/10536-0. O estágio foi realizado no Departamento de Matemática da Universidade de Coimbra, Portugal. Tem experiência na área de Educação Matemática, atuando principalmente nos seguintes temas: Produção de Conhecimento, Comunicação, Colaboração, Tecnologias Digitais, Formação Inicial e Continuada de Professores de Matemática. Atualmente é professora substituta no Departamento de Metodologia de Ensino, da Universidade Federal de São Carlos (UFSCar).

Número do Orcid: 0000-0002-6656-2538

Maurício Braga Julio

Aluno do ensino médio integrado ao Curso Técnico de Informática no Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS), do Campus Osório (2019). Ao decorrer do curso interessou-se pela área de Ciência da Computação, adquirindo experiências relacionadas à metodologia, à educação e às técnicas de computação. Atualmente é bolsista do Programa Institucional de bolsas de Extensão (Pibex) do projeto Programando Fácil: Conhecendo a Computação; trabalhando com oficinas lúdicas que envolvem pensamento computacional, raciocínio lógico e programação básica, oferecidas para alunos das séries finais do ensino fundamental de escolas da região do Litoral Norte Gaúcho.

E-mail: mauriciobragajulio@gmail.com

Nariana de Oliveira Tavares

Graduada em Filosofia pela Faculdade de Filosofia, Ciência e Letras de Caruaru, pós-graduanda em Educação, Diversidade e Cidadania pela Fael. Servidora pública municipal desde 2009. Atualmente, professora do Letramento em Programação na Escola de Tempo Integral Álvaro Lins na cidade de Caruaru/PE com turmas de 4º e 5º anos.

Nizam Omar

Professor da Universidade Presbiteriana Mackenzie, possui graduação em Engenharia Mecânica pelo Instituto Tecnológico de Aeronáutica (1974), mestrado em Matemática Aplicada – Informática

pelo Instituto Tecnológico de Aeronáutica (1979) e doutorado em Informática pela Pontifícia Universidade Católica do Rio de Janeiro (1989). Professor titular do ITA (aposentado), atualmente é professor adjunto III da Universidade Presbiteriana Mackenzie. Tem experiência na área de Ciência da Computação, com ênfase em Teoria da Computação, Inteligência Artificial e Linguagem Formais e Autômatos, atuando principalmente nos seguintes temas: tutores inteligentes, adaptatividade didática, sistemas especialistas, redes neurais e ambiente de aprendizagem.

Número do Orcid: 0000-0001-8361-2165

Peter Rodrigo Trento

Secretaria de Educação de Vinhedo-SP. Vencedor do Prêmio Professores do Brasil (MEC) 2017, finalista do prêmio Educador Nota 10 de 2017, finalista do Desafio Aprendizagem Criativa Brasil 2018 do MIT MEDIA LAB, palestrante TEDx, Google Innovator, Google Educator Group Leader, mestre na Faculdade de Educação da UNICAMP pelo Programa de Pós-Graduação Multiunidades em Ensino de Ciências e Matemática (PECIM). Possui bacharelado e licenciatura em Geografia pela Universidade de São Paulo – USP. Tem experiência nas áreas de ensino, aprendizagem e formação de professores na área de Ensino de Ciências e Matemática e tecnologia educacional, formando professores da Universidade Paranaense (UNIPAR). Trabalhei com a produção de material didático e planos de aula para a Google, no projeto “Eu sou Amazônia”. Produziu também sequências didáticas e planos de aulas para as editoras FTD em 2017 e para a Editora Moderna em 2018, para livros que irão compor o PNLD 2019 e PNLD 2020. Atualmente, é revisor de conteúdo de Geografia na Google/Youtube Edu, mentor do Time de Autores da Nova Escola, juiz na Technovation e professor Innovator Certificado pela Google em uso de tecnologias em sala de aula, além de professor de Geografia e coordenador de inovação e tecnologia educacional da rede municipal de Vinhedo-SP.

Sabrina Moreira de Oliveira

Possui graduação em Administração pelo Centro Universitário UNA (2004) e mestrado em Modelagem Matemática e Computacional pelo Centro Federal de Educação Tecnológica de Minas Gerais (2007). É estudante de doutorado pela Université Libre de Bruxelles – ULB. Tem experiência na área de Engenharia de Transportes, com ênfase em Engenharia de Transportes, atuando principalmente nos seguintes temas: modelagem, programação linear, programação não linear, otimização de transportes por meio de métodos exatos, heurísticas e metaheurísticas. É professora de Programação de Computadores, Simulação e Pesquisa Operacional no curso de Engenharia de Produção da Faculdade Ibmec-MG.

Simone André da Costa Cavalheiro

Instituição UFPel, É Engenheira Civil pela Universidade Católica de Pelotas (1998), Licenciada em Matemática pela Universidade Federal de Pelotas (1998), possui mestrado (2000) e doutorado (2010) em Ciência da Computação pela Universidade Federal do Rio Grande do Sul. Atualmente é professora associada da Universidade Federal de Pelotas (UFPel) e membro do Programa de Pós-Graduação em Computação da UFPel. Atua na área de Fundamentos da Computação e Educação em Computação, principalmente em Métodos Formais e Pensamento Computacional. Desde 2013 é coordenadora do projeto “Explorando o Pensamento Computacional para Qualificação do Ensino Fundamental”, o qual integra ensino, pesquisa e extensão, com o apoio da Secretaria Municipal de Educação e Desporto de Pelotas. Desde 2017, é membro da Comissão de Educação Básica da Sociedade Brasileira de Computação (SBC) e vem contribuindo com a proposta de Diretrizes para ensino de Computação na Educação Básica da SBC. Número do Orcid: 0000-0002-7442-7379.

Profa. Sonia Mara Martins de Mattos Souza

Possui graduação em Educação Física pela Universidade de Passo Fundo (1989) e especialização em Tecnologias da Informação e da Comunicação na Promoção da Aprendizagem pela Universidade

Federal do Rio Grande do Sul (2008). Tem experiência na área de Educação, com ênfase em Ensino-Aprendizagem.

Sueli Liberatti Javaroni

Bacharel em Matemática pela Universidade de São Paulo (1989), mestre em Matemática pela Universidade Federal de São Carlos (1993) e doutora em Educação Matemática pela Universidade Estadual Paulista (2007). Professora pesquisadora em regime de dedicação exclusiva do Departamento de Matemática da Faculdade de Ciências (FC) da UNESP, campus de Bauru. Atua nos cursos de graduação nas disciplinas de Cálculo Diferencial e Integral, Cálculo Numérico Computacional, Tecnologias Digitais em Educação Matemática e Educação Financeira. Credenciada no Programa de Pós-Graduação em Educação Matemática (PPGEM) do Instituto de Geociências e Ciências Exatas (IGCE) da UNESP, campus de Rio Claro, onde orienta mestrandos e doutorandos na linha de pesquisa em Tecnologias Digitais e Educação Matemática. Membro do Grupo de Pesquisa em Informática, outras Mídias e Educação Matemática (GPIMEM). Coordenou o projeto de pesquisa “Mapeamento do uso de tecnologias da informação nas aulas de Matemática no Estado de São Paulo” (projeto nº 16429 – Edital 049/2012/CAPES/INEP, Programa Observatório da Educação – OBEDUC). Principais áreas de interesse de pesquisa: Formação de Professores de Matemática, Tecnologias Digitais em Educação Matemática, Modelagem Matemática, Pensamento Computacional e o Ensino e Aprendizagem de Matemática. Número do Orcid: 0000-0002-1948-4346

Valéria Farinazzo Martins

É bacharel em Ciência da Computação pela Universidade Estadual Paulista Júlio de Mesquita Filho (1995), mestre em Ciência da Computação pela Universidade Federal de São Carlos (2000) e doutora em Engenharia Elétrica pela Escola Politécnica da Universidade de São Paulo (2011). Tem estágio pós-doutoral pela Universidade Federal de Itajubá (2014). É professora nos cursos de Ciência da Computação, Sistemas de Informação e Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade Presbiteriana Mackenzie. Atua como

professora colaboradora no Programa de Pós-Graduação em Distúrbios do Desenvolvimento. Atua, principalmente, em temas relacionados à Interação Humano-Computador, Informática na Educação e Informática na Saúde da Universidade Presbiteriana Mackenzie. Número do Orcid: 0000-0002-5058-6017

Vanessa Polli

Possui graduação em Letras Português/ Espanhol e Respectivas Literaturas pela URI – Campus de Erechim (2009), pós-graduação em Educação Interdisciplinar com ênfase em Língua Espanhola pela Faculdade IDEAU de Getúlio Vargas (2011) e mestrado em Estudos Linguísticos pela UFFS – Campus Chapecó (2014). É professora de Língua Portuguesa da Rede Municipal de Educação de Getúlio Vargas/ RS desde 2012. Em 2018 iniciou no Programa Letramento em Programação.

Vitória de Souza Fabricio

Aluna do ensino médio integrado ao Ensino Técnico em Informática e diretora de Relações da Diretoria Executiva do Grêmio Estudantil (DGEÇO) dos Cursos Técnicos Integrados ao ensino médio do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS), do Campus Osório (2019). Atualmente é bolsista do Programa Institucional de Bolsas de Extensão (PIBEX) do IFRS, trabalhando com projetos na área de educação cultural, desenvolvendo atividades culturais no Campus Osório do IFRS, atuando na formação complementar dos estudantes e com a comunidade externa. Desde 2018 atua como estudante voluntária em outras ações de extensão, voltadas para a Educação Computacional a partir do raciocínio lógico e da programação básica, aplicadas em oficinas com alunos das séries finais do ensino fundamental de escolas da região do Litoral Norte Gaúcho.

E-mail: vitoria.souza.fabricio@gmail.com