

## Formação Continuada de Professores da Educação Básica para o Ensino de Algoritmos e Programação

Nathalia da Cruz Alves<sup>1</sup>, Fabíola Maria Kretzer<sup>1</sup>, Christiane Gresse von Wangenheim<sup>1</sup>, Miriam Nathalie Fortuna Ferreira<sup>1</sup>, Jean Carlo Rossa Hauck<sup>1</sup>

<sup>1</sup>Departamento de Informática e Estatística

Universidade Federal de Santa Catarina (UFSC) – Florianópolis – SC – Brasil

nathalia.alves@posgrad.ufsc.br, fabiola.kretzer@grad.ufsc.br,  
{c.wangenheim, jean.hauck}@ufsc.br, nathalie.fortuna@posgrad.ufsc.br

**Abstract.** *Introducing computer education into school requires trained teachers. However, as formally trained computing teachers in K-12 are scarce, some initiatives aim at training in-service teachers from other areas. In this context, this article presents the result of a teacher training covering basic computing competencies as well as pedagogical and technological knowledge to enable the teaching of computing in a multidisciplinary way in their respective disciplines following a constructionist approach. Results from a first application demonstrate the viability and positive contribution of such an in-service teacher training for computing education, contributing to the popularization of computing competencies in K-12.*

**Resumo.** *Introduzir o ensino de computação na escola requer professores com competências básicas de computação. No entanto, como professores com licenciatura em computação são escassos, algumas iniciativas visam formar professores de outras áreas. Nesse contexto, este artigo apresenta um curso com professores, abrangendo competências básicas de computação, incluindo conhecimentos pedagógicos e tecnológicos para possibilitar o ensino de computação de maneira multidisciplinar em suas respectivas disciplinas seguindo uma abordagem construcionista. Resultados de uma primeira aplicação demonstram a viabilidade e a contribuição positiva da formação continuada de professores para o ensino de computação na Educação Básica.*

### 1. Introdução

A influência da computação no dia-a-dia é vivenciada em níveis pessoais, sociais e globais [CSTA 2016]. Num mundo rodeado de tecnologia, a formação de cidadãos bem preparados requer o ensino de princípios e práticas da computação, o que vai além do simples uso da Tecnologia da Informação (TI). Isso inclui o uso de conceitos de computação como uma ferramenta para entender e ser um participante ativo de uma sociedade digital na qual a computação é onipresente [Wing 2006]. Consequentemente, é importante que as pessoas aprendam sobre computação desde a infância.

Uma alternativa pode ser por meio de iniciativas que buscam ensinar computação a crianças e adolescentes na Educação Básica, seguindo uma abordagem construcionista [Grover e Pea 2013], tipicamente focando na programação usando ambientes de programação baseada em blocos, como App Inventor (<https://appinventor.mit.edu/>) ou Scratch (<http://scratch.mit.edu/>) para criar animações, jogos ou aplicativos móveis.

Porém, a implementação do ensino de computação nas escolas exige professores motivados e dedicados com competências computacionais, pedagógicas e tecnológicas, a fim de ensinar computação de uma forma envolvente aos alunos [Gal-Ezer e Stephenson 2010]. Idealmente, os professores seriam preparados por meio de um rigoroso programa de formação de professores, normalmente por meio de curso de graduação ou pós-graduação. Embora tais programas ofereçam aos professores uma compreensão profunda da disciplina, é um caminho que requer vários anos para ser concluído [Granor et al. 2016]. A inscrição e o acompanhamento de tais programas é também cada vez mais limitado pelo crescimento contínuo do emprego na indústria, já que o incentivo para seguir carreiras lucrativas na indústria de TI é muito maior do que o incentivo para ensinar. Como resultado, o número de concluintes dos cursos de licenciatura na área da computação é muito escasso [INEP 2019] e os alunos da Educação Básica possuem poucas oportunidades de aprender computação e de compreender como a computação influencia sua vida cotidiana [CSTA 2016].

Outra possibilidade para implementar o ensino de computação em maior escala na Educação Básica, pode ser de forma multidisciplinar [Alves et al. 2016]. Deste modo, os alunos podem aprender computação no contexto da disciplina de formação e atuação do professor. Nesse cenário, os professores já possuem conhecimento de conteúdo da área de estudo que ensinam (por exemplo, História, Artes, Ciências, etc.), conhecimento pedagógico geral, mas tipicamente não possuem competências de computação [Alves et al. 2016]. A fim de ensinar computação de maneira eficaz e envolvente, os professores precisam ter conhecimento de conteúdo em computação (especialmente algoritmos e programação), conhecimento de conteúdo pedagógico (como ajudar os alunos a aprender computação), e também conhecimento tecnológico (para instalar e manter a infraestrutura de TI necessária) [Liu et al. 2011]. Desta forma, é importante recrutar e ensinar computação a professores *in-service* que possuem credenciais em outras áreas de estudo [Goode 2008]. Consequentemente, é importante oferecer a formação continuada para criar condições a curto prazo para a popularização da computação nas escolas [Kretzer et al. 2020].

Porém, atualmente existem muito poucas unidades instrucionais visando à formação continuada de professores *in-service* referentes ao ensino de competências de computação para os Anos Finais do Ensino Fundamental. Em um mapeamento sistemático recente foram encontradas apenas 16 unidades instrucionais para a formação continuada voltadas ao ensino de computação para professores *in-service* da Educação Básica [Kretzer et al. 2020]. Foi observada também a falta de abordagens mais completas, incluindo não só competências relacionadas ao conteúdo, mas também competências pedagógicas referentes ao ensino de computação bem como competências tecnológicas para lidar com a infraestrutura necessária nesse tipo de ensino. Assim, este artigo tem o objetivo de apresentar uma unidade instrucional (UI) e sua aplicação na prática com professores da Educação Básica (atuantes em outras disciplinas, como Artes, História, Geografia e outras).

## 2. Trabalhos relacionados

Resultados de um mapeamento sistemático [Kretzer et al. 2020] demonstram que atualmente existem muito poucas unidades instrucionais visando a formação de professores *in-service* referentes ao ensino de competências de computação para o Ensino Fundamental Anos Finais (apenas 16 no total). A maioria dessas unidades instrucionais foca em conteúdos relacionados a algoritmos e programação. As unidades instrucionais geralmente também abordam o ensino de competências pedagógicas aos professores, enquanto competências tecnológicas são abordadas somente em poucas UI.

Vários métodos são utilizados para ensinar esse conteúdo, mas a maioria emprega abordagens de aprendizagem ativa. Tipicamente, é feita a introdução de um conceito com demonstrações de exemplos e em seguida os participantes desenvolvem algum artefato. Em outras UI, os professores realizam a aula com base em tutoriais e/ou resolução de problemas. O conteúdo é explicado aos professores utilizando diferentes materiais instrucionais, como vídeos, slides, tutoriais e exemplos, proporcionando caminhos de aprendizagem flexíveis e mistura de conteúdo. A aprendizagem e o desempenho dos professores nessas aulas são medidos por meio de questionários, observações e artefatos criados.

Os diversos ambientes e plataformas de programação usados também chamam a atenção, abrangendo desde robótica utilizando o *Lego NXT-G* até ambientes de programação baseada em blocos, como App Inventor e Scratch [Kretzer et al. 2020]. Algumas UI utilizam múltiplas linguagens/plataformas, supondo que ensinar programação como parte da computação não é sobre o uso de uma plataforma ou o aprendizado de uma linguagem de programação específica, mas sobre como aprender conceitos de computação em geral [Martinez et al., 2016].

De forma geral, observa-se uma falta de informações sobre a base teórica e o método de desenvolvimento das UI. Apenas três UI apresentam informações sobre a metodologia utilizada para desenvolvê-las sistematicamente [Kretzer et al. 2020]. A maioria das UIs também foi avaliada somente por estudos de caso com pequenas amostras e/ou de forma *ad-hoc*. Isso mostra a escassez de trabalhos nessa área e a necessidade de desenvolvimento de mais unidades instrucionais para esse fim.

### 3. Metodologia de Pesquisa

O objetivo desta pesquisa é o desenvolvimento, a aplicação e a avaliação de uma unidade instrucional. Para atingir este objetivo é realizado um estudo de caso exploratório para compreender os fenômenos observados durante as aplicações da unidade instrucional em um contexto particular e identificar direcionamentos para trabalhos futuros. O estudo de caso é realizado conforme os procedimentos propostos por Yin (2009):

**Definição do estudo:** O estudo é definido em termos do objetivo e perguntas de pesquisa e o *design* de pesquisa. A partir do objetivo e das perguntas de pesquisa, as medidas para a coleta de dados são sistematicamente derivadas utilizando o método GQM [Basili et al. 1994]. Para a operacionalização da coleta de dados são definidos instrumentos de coleta de dados para as medidas definidas.

**Execução do estudo:** A execução do estudo é realizada adotando-se o modelo ADDIE [Branch 2009] de *design* instrucional. Em uma primeira etapa, a unidade instrucional é desenvolvida. Para isso, são caracterizados os aprendizes e o ambiente em que a unidade instrucional é aplicada. São levantadas as necessidades de aprendizagem e, com base nessas informações, os objetivos de aprendizagem são definidos. De acordo com a análise de contexto, é projetada a unidade instrucional, definindo o seu conteúdo, a sequência e os métodos instrucionais a serem adotados. Em seguida, o material instrucional é desenvolvido. Durante a segunda etapa da execução do estudo, a unidade instrucional é aplicada na prática e avaliada, coletando-se os dados conforme a definição do estudo.

**Análise e Interpretação do estudo:** Nesta etapa são analisados os dados em relação às perguntas de pesquisa, usando métodos quantitativos e qualitativos. Ao final, os resultados são interpretados e discutidos.

#### 4. Unidade instrucional para Formação Continuada de Professores

A unidade instrucional é voltada ao ensino de conceitos básicos da computação principalmente com enfoque em algoritmos e programação. A UI aborda conhecimento tecnológico, pedagógico e de conteúdo, conforme sugerido por Mishra e Koehler (2006). A unidade instrucional é alinhada ao *framework* da CSTA (2016) e às diretrizes de ensino da SBC (2018) buscando habilitar professores *in-service* a ensinar algoritmos e programação por meio de atividades de programação, projetos de jogos com Scratch e aplicativos com App Inventor.

O público-alvo da unidade instrucional consiste em professores do Ensino Fundamental - Anos Finais. A maioria dos professores possui idade entre 35 e 54 anos, sendo que 90% dos professores possuem nível superior completo em cursos de licenciatura em suas respectivas áreas [INEP 2016]. A maioria dos professores utiliza programas e aplicativos educacionais durante suas aulas [INEP 2017]. No entanto, os professores geralmente não possuem conhecimento aprofundado em computação [Alves et al. 2016].

##### 4.1. Objetivos de aprendizagem

O objetivo geral da UI é formar professores *in-service* para o ensino de computação nos Anos Finais do Ensino Fundamental. Considerando que esses professores, em geral, têm conhecimento de conteúdo e pedagógico da área que ensinam, mas normalmente não possuem competências em computação, torna-se necessário ensinar conhecimento de conteúdo, pedagógico e tecnológico referentes ao ensino de computação (Tabela 1).

**Tabela 1. Extrato de objetivos de aprendizagem da UI [Kretzer 2020]**

|                          | ID   | Objetivo de aprendizagem   | Referência              |
|--------------------------|------|--|-------------------------|
| Conhecimento do conteúdo | OA1  | Compreender algoritmos como um conjunto de instruções passo-a-passo para realizar tarefas  | [CSTA 2016]             |
|                          | OA2  | Conhecer diferentes ambientes de programação   | [Ferreira et al. 2020]  |
|                          | ...  | ...  | ...                     |
| Conhecimento pedagógico  | OA22 | Conhecer e entender guias e diretrizes de currículo para o ensino de computação na Educação Básica                                       | [Mishra e Koehler 2006] |
|                          | OA23 | Conhecer métodos de ensino e aprendizagem de computação e como ele se aplica em sala de aula   | [Mishra e Koehler 2006] |
|                          | ...  | ...  | ...                     |
| Conhecimento tecnológico | OA26 | Compreender habilidades para tecnologias específicas e como lidar com novas tecnologias  | [Mishra e Koehler 2006] |
|                          | OA27 | Revisar/criar a infraestrutura técnica necessária para realizar aulas de implementação de atividades de programação, jogos e aplicativos | [Mishra e Koehler 2006] |
|                          | ...  | ...  | ...                     |

##### 4.2. Estratégia instrucional e ambientes de ensino programação

A UI utiliza diferentes métodos instrucionais, os quais variam desde instrução direta (instrucionista), até atividades práticas por meio de uma abordagem construcionista de aprendizagem baseada em projetos adotando a estratégia de ação computacional [Tissenbaum et al. 2019]. Assim, a unidade instrucional prevê que professores desenvolvam seus próprios projetos e criem soluções computacionais de modo a construir o conhecimento de como podem utilizá-las em suas aulas, motivando a multidisciplinaridade da aplicação.

Com o intuito de atingir os objetivos de aprendizagem, são utilizados três ambientes de programação baseados em blocos em diferentes etapas, tendo em vista que o objetivo não é ensinar o uso de uma plataforma ou uma linguagem de programação específica, mas sim como ensinar conceitos de computação [Martinez et al. 2016]. Na introdução da UI é utilizado o Code.org por ser um ambiente mais simples com atividades predefinidas. Em seguida, o ambiente Scratch é introduzido proporcionando uma maior liberdade na criação de projetos multidisciplinares. Por fim, é utilizado o ambiente App Inventor por conter diversos conceitos considerados avançados e possibilitar o uso de componentes de um *smartphone* em projetos multidisciplinares adotando a ação computacional.

Considerando o contexto, a unidade é dividida em 4 módulos como mostra a Tabela 2. Um plano de ensino é desenvolvido com a duração de cada aula, conteúdo, objetivo de aprendizagem, método instrucional e avaliação.

**Tabela 2. Módulos da unidade instrucional**

| Módulo | Descrição   | Duração |
|--------|---|---------|
| M1     | Motivação e conceitos básicos por meio de uma atividade desplugada e Code.org   | 4h/a    |
| M2A    | Aprendizagem baseada em projetos para conhecer e desenvolver jogos multidisciplinares com Scratch   | 10h/a   |
| M2B1   | Abordagem instrucionista para conhecer e desenvolver um aplicativo multidisciplinar com App Inventor  | 8h/a    |
| M2B2   | Aprendizagem baseada em projetos usando <i>design thinking</i> para conceber e desenvolver um aplicativo multidisciplinar com App Inventor            | 20h/a   |
| M3     | Aspectos pedagógicos no ensino de computação: currículos e diretrizes, estratégias de ensino, integração multidisciplinar e avaliação da aprendizagem | 4h/a    |
| M4     | Aspectos tecnológicos no ensino de computação: pré-requisitos de infraestrutura, dicas e métodos de resolução de problemas técnicos recorrentes.      | 2h/a    |

Diversos materiais didáticos são desenvolvidos para cada módulo conforme ilustrado na Tabela 3.

**Tabela 3. Exemplos de materiais didáticos desenvolvidos para a unidade instrucional**

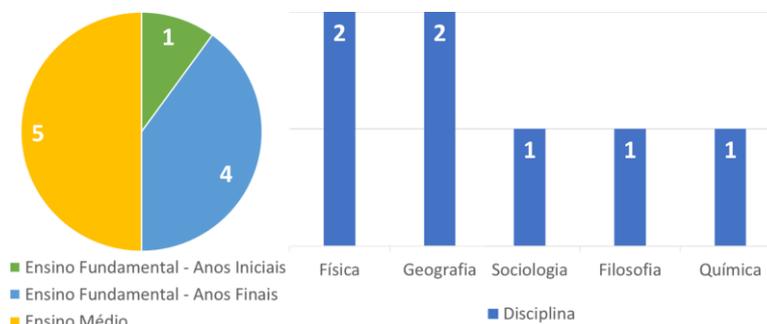
| Material   | Descrição  | Ilustração  |
|--|--|---|
| Jogo <i>SplashCode</i> [Gresse von Wangenheim et al. 2019] | Jogo de tabuleiro para a aplicação dos conceitos de algoritmos. Os jogadores utilizam as cartas de movimento para fazer os personagens chegarem até o campo <i>casa</i> , desviando dos obstáculos.                              |  |
| Slides com conteúdo  | Slides com o conteúdo de todos os conceitos abordados na UI.   |  |
| Folha de atividades  | Guia para a produção dos artefatos propostos no processo de desenvolvimento de apps e de jogos.  |  |
| Tutoriais  | Tutoriais para desenvolvimento de funcionalidades características de diversos apps no App Inventor, como banco de dados, menu lateral, <i>login</i> e mapas, bem como utilização de diversos conceitos de computação no Scratch. |  |

## 5. Aplicação e Avaliação

### 5.1. Aplicação da Unidade Instrucional

Participaram da formação continuada sete professores de diferentes disciplinas e anos escolares (Figura 1) de uma escola privada da cidade Florianópolis/SC durante aproximadamente 4 meses em 2019-2020, em encontros semanais de quatro horas na própria escola. A sala utilizada não era informatizada, então os professores utilizaram *laptops* próprios ou da escola, com conexão à internet por meio de uma rede Wi-Fi.

Os professores foram convidados e selecionados pela escola, os quais participaram de maneira voluntária no período que não lecionavam. Alguns dos professores lecionavam em mais de uma disciplina ou em mais de um nível escolar (Figura 1), por isso, a soma do número de professores no gráfico é maior que o número de professores que participaram da aplicação.



**Figura 1. Nível escolar de ensino e disciplinas lecionadas pelos professores**

O primeiro e segundo módulo foram lecionados presencialmente, as aulas envolveram a apresentação de slides, introdução de conceitos básicos de computação usando Code.org, Scratch e App Inventor. Foram também apresentados vídeos ilustrativos, tutoriais de criação de aplicativos e jogos, atividades e exercícios de fixação (Figura 2).



**Figura 2. Encontros presenciais na escola**

Como parte de uma estratégia de atividade prática, no módulo 1, os professores tiveram como tarefa final a lição Labirinto Clássico da plataforma Code.org. Todos os professores concluíram os 20 níveis da lição, conforme indicado na Figura 3. No módulo 2A, foi utilizado o ambiente de programação Scratch seguindo a aprendizagem baseada em projetos cuja tarefa final incluiu a criação de um projeto multidisciplinar com um tema relacionado à disciplina lecionada por cada professor. A Figura 3 apresenta alguns exemplos de projetos desenvolvidos pelos professores de diferentes disciplinas.



Figura 3. Exemplos de resultados das tarefas finais dos módulos 1 e 2A

No módulo M2B (1 e 2) foi utilizado o ambiente de programação App Inventor. Na primeira parte utilizou-se uma abordagem instrucionista para ensinar alguns componentes complexos do ambiente. Na segunda parte adotou-se a estratégia de ação computacional com *design thinking*. A tarefa final envolveu a criação de um aplicativo com um tema de acordo com a disciplina lecionada por cada professor (Figura 4).



Figura 4. Exemplos de resultado da tarefa final do módulo 2B criado pelos professores

No final do curso em 2020, por causa da proibição de aulas presenciais em consequência da pandemia, as aulas do terceiro e quarto módulo foram lecionadas a distância via a plataforma Google Meet. Tendo em vista as limitações de aulas online, as aulas envolveram somente apresentação de slides e exercícios simples de fixação.

## 5.2. Avaliação da unidade instrucional

Com o objetivo de avaliar qualidade da unidade, foi utilizado o modelo dTECT [Gresse von Wangenheim et al. 2017] completado pela análise do desempenho dos participantes da formação continuada. A coleta de dados foi feita durante todo o curso, sempre ao final dos módulos [Kretzer 2020].

**Qual a qualidade da unidade instrucional?** De forma geral, os professores apontaram que suas experiências de aprendizagem foram positivas. Eles consideram a complexidade da formação aceitável na maioria dos módulos, considerando os primeiros módulos M1/M2A fáceis. Porém, o desenvolvimento de apps com App Inventor (M2B1 e M2B2) foi percebido como conteúdo mais difícil. Isto também se dá pelo fato de que esses módulos abordam conceitos mais avançados, incluindo estruturas de dados mais complexas, como funções de ordem superior, conceitos de banco de dados e uso de componentes de sensores do App Inventor. Uma forma de superar essa dificuldade indicada pelos professores pode ser por meio de atividades de casa constantes, de forma que os professores consigam identificar melhor as dificuldades encontradas dentro do ambiente e trazê-las para a sala de aula ou ainda adotando programação em pares durante as aulas.

Os conteúdos referentes ao conhecimento pedagógico (M3) e conhecimento tecnológico (M4) foram também considerados fáceis pelos professores apesar de lecionados a distância. Isso é esperado tendo em vista que o M3 aborda aspectos pedagógicos no ensino de computação, os quais podem ter uma semelhança com o conhecimento pedagógico a priori dos professores. Já a facilidade do M4 foi indicada devido a simplicidade e clareza como foram apresentados aspectos técnicos de infraestrutura e sugestões de como enfrentar problemas de conexão, contorno de possíveis bugs e erros dos ambientes de programação.

**Qual o desempenho de aprendizagem dos participantes?** O desempenho dos participantes foi bom ou excelente nos primeiros módulos (M1 e M2A). Eles demonstraram capacidade de aplicação dos conteúdos ensinados, observando também um ótimo desempenho nos projetos multidisciplinares desenvolvidos com Scratch por todos. Contudo, no módulo com App Inventor (M2B1 e M2B2), alguns participantes apresentaram um desempenho suficiente ou abaixo do esperado. Um dos motivos pode ser a dificuldade observada do módulo como sendo difícil e/ou muito difícil e a necessidade de estar em constante contato com o ambiente App Inventor de modo a maturar o conteúdo para iniciantes na área. Mesmo tendo contato já com outros ambientes baseados em blocos, o ambiente App Inventor requer um maior tempo de dedicação fora da sala de aula, o que, segundo os participantes, em muitos momentos não era possível devido a outras responsabilidades pessoais e profissionais.

Em relação aos módulos 3 e 4 todos os professores que participaram das aulas apresentaram desempenho ótimo ou excelente. No entanto, devido a pandemia e substituição de aulas presenciais por aulas a distância em todos os níveis da escola, houve um aumento da carga de trabalho dos participantes impossibilitando três professores de estarem presentes durante a avaliação do desempenho, principalmente aqueles com responsabilidades de coordenação na escola. Assim, não foi possível avaliar o desempenho de todos.

**Qual a aplicabilidade nas disciplinas dos participantes?** Em relação a aplicabilidade dos conteúdos vistos, os professores foram unânimes em relação ao conteúdo dos módulos M2A, M3 e M4. Além disso, nos demais módulos, a maioria também respondeu que irá ensinar os conteúdos vistos durante a unidade instrucional em suas disciplinas. Essas percepções também se refletem em comentários feitos pelos professores (Tabela 4).

**Tabela 4. Comentários dos participantes da formação continuada**

| Prof. | Comentários  |
|-------|--|
| P1    | "O M1 foi muito divertido e lúdico; as atividades introdutórias foram bem concretas para ilustrar o que é programação"; "Foi interessante ter contato com a ferramenta App Inventor no M2B1 e M2B2, mas foi extenso no calendário e dificultou o meu aprendizado."   |
| P2    | "O que mais gostei no M1 foi a atividade do labirinto com Code.org."; "Achei o M3 muito completo. Abordou todas as questões necessárias."  |
| P3    | "Sinceramente gostei de toda a abordagem do M1, dinâmica e interativa! Chamou minha atenção a possibilidade de aplicação com diferentes faixas etárias."; "O M4 foi interessante e aplicado."  |
| P4    | "O que mais me chamou atenção no M2B2 é sobre a importância do planejamento e a necessidade de revisitar as ideias iniciais e fazer as devidas reflexões acerca da importância e viabilidade das mudanças em relação ao plano inicial."; "Gostei do layout da ferramenta Scratch no M2B1, porém não gostei muito dos exercícios de fixação." |
| P5    | "Achei o M1 uma excelente introdução, o que mais gostei foram as questões práticas."   |
| P6    | "Achei o M2A muito interessante e me deu uma ideia de como programar"; "O M2B1 é um módulo mais complexo, precisando ter muito mais atenção nos detalhes de programação."  |
| P7    | "Gostei muito da parte prática do M1"; "O que mais gostei no M2A foi a possibilidade de testar os comandos principais da plataforma."; "O que mais gostei no M2B1 foi a configuração de mapas e localizações pelo sensor do celular usando App Inventor."  |

Em geral, o curso de formação continuada demonstrou bons resultados, com algumas dificuldades somente no módulo do App Inventor. Como oportunidade de melhoria, sugere-se usar a abordagem Use-Modifique-Crie no módulo do App Inventor para que os participantes tenham mais oportunidades de compreender e aplicar o conteúdo.

**Ameaças à validade.** O estudo de caso desta pesquisa apresenta algumas limitações em relação a quantidade de dados, pois participaram apenas sete professores. No entanto, considerando a natureza exploratória da pesquisa neste momento, considera-se essa quantidade aceitável. Para mitigar ameaças em relação a validade de conclusão foi conduzido um estudo de caso seguindo a abordagem GQM [Basili et al. 1994] utilizando o modelo de avaliação dTECT [Gresse von Wangenheim 2017] já validado. Assim, sendo esta avaliação um estudo exploratório, considera-se que os resultados podem ser utilizados para obter uma primeira indicação da qualidade da unidade instrucional. Para aumentar a validade externa também se prevê a aplicação de um estudo envolvendo professores de outras escolas/locais.

#### 4. Conclusão

Como resultado do presente trabalho foi relatada uma experiência com uma unidade instrucional para ensinar conceitos básicos de computação, bem como aspectos pedagógicos e tecnológicos aos professores da Educação Básica. A formação de professores de outras disciplinas mostra-se como uma alternativa para formar professores e assim permitir a utilização desse conhecimento de forma interdisciplinar inserida em conteúdo programado do Ensino Fundamental. Com base nesses resultados positivos, visa-se a ampliação da aplicação e avaliação dessa unidade instrucional e o acompanhamento da aplicação das competências pelos professores nas suas disciplinas contribuindo na democratização da competência de computação no nível da Educação Básica.

#### Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001 e do Conselho Nacional de Desenvolvimento Científico e Tecnológico – Brasil (CNPq).

#### Referências

- Alves, N. da C., Gresse von Wangenheim, C., Rodrigues, P., Hauck, J., Borgatto, A. (2016) “Ensino de Computação de Forma Multidisciplinar em Disciplinas de História no Ensino Fundamental – Um Estudo de Caso”. *Revista Brasileira de Informática na Educação*, 24(3).
- Basili, V. R., Caldiera, G., Rombach, H. D. (1994) “Goal Question Metric Paradigm”. *Enc. of Software Engineering*, John Wiley & Sons.
- Branch, R. M. (2009) “Instructional Design: The ADDIE Approach”. New York: Springer.
- CSTA. ACM. (2016) “CSTA K –12 Computer Science Framework”. Disponível em: <https://csteachers.org/>. Acesso em: maio de 2020.
- De Kereki, I. F., Manataki, A. (2016) “Code Yourself and A Programar: a bilingual MOOC for teaching Computer Science to teenagers”. In: *Proc. of the Frontiers in Education Conference*, Erie, PA, EUA.
- Ferreira, M. N. F., Pinheiro, F. da C., Gresse von Wangenheim, C., Filho, R. M., Hauck, J. C. R. (2020). Ensinando design de interface de usuário de aplicativos móveis no ensino fundamental. *Revista Brasileira de Informática na Educação*, 28.

Gal-Ezer, J., Stephenson, C. (2010) “Computer science teacher preparation is critical”. *ACM Inroads*, 1(1), 61-66.

Granor, N., Delyser, L. A., Wang, K. (2016) “TEALS: Teacher Professional Development Using Industry Volunteers”. In: *Proc. of the 47th ACM Technical Symposium on Computing Science Education*, Memphis, EUA, 60-65.

Gresse von Wangenheim, C., Petri, G., Zibetti, A. W., Borgatto, A. F., Hauck, J. C. R., Pacheco, F. S., Missfeldt Filho, R. (2017) “dTECT: A Model for the Evaluation of Instructional Units for Teaching Computing in Middle School”. *Informatics in Education*, 16(2), 301-318.

Gresse von Wangenheim, C., Medeiros, G., Filho, R., Petri, G., Pinheiro, F., Ferreira, M., Hauck, J. C. R. (2019) “Desenvolvimento e Avaliação de um Jogo de Tabuleiro para Ensinar o Conceito de Algoritmos na Educação Básica”. *Revista Brasileira de Informática na Educação*, 27(3).

Grover, S., Pea, R. (2013) “Computational thinking in K–12: A review of the state of the field”. *Educational Researcher*, 42(1), 38–43.

Goode, J. (2008) “Increasing Diversity in K-12 Computer Science: Strategies from the Field”. In: *Proc. of the 39th SIGCSE Technical Symposium on Computer Science Education*, Portland, OR, EUA.

INEP. (2019) “Sinopse Estatística da Educação Superior 2018”. Disponível em: <http://inep.gov.br/sinopses-estatisticas-da-educacao-superior>. Acesso em: maio de 2020

INEP. (2016) “Notas Estatísticas Censo Escolar da Educação Básica”. Disponível em: [http://download.inep.gov.br/educacao\\_basica/censo\\_escolar/notas\\_estatisticas/2017/notas\\_estatisticas\\_censo\\_escolar\\_da\\_educacao\\_basica\\_2016.pdf](http://download.inep.gov.br/educacao_basica/censo_escolar/notas_estatisticas/2017/notas_estatisticas_censo_escolar_da_educacao_basica_2016.pdf). Acesso em: maio de 2020

INEP. (2017) “Microdados da Prova Brasil”. Disponível em: <https://www.qedu.org.br/brasil/pessoas/professor>. Acesso em: maio de 2020

Kretzer, F. M. (2020). “Desenvolvimento de uma Unidade Instrucional para Formação de Professores da Educação Básica para o Ensino de Computação”. TCC, CCO/INE/UFSC, Florianópolis, Brasil.

Kretzer, F. M., Gresse von Wangenheim, C., Hauck, J. C. R., Pacheco, F. S. (2020) “Formação Continuada de Professores para o Ensino de Algoritmos e Programação na Educação Básica: Um Estudo de Mapeamento Sistemático”. *Revista Brasileira de Informática na Educação*, 28, 389-419.

Liu, J., Lin, C. H., Hasson, E. P., Barnett, Z., Xu, Y. I. (2011) “Introducing computer science to K-12 through a summer computing workshop for teachers”. In: *Proc. of the 42nd ACM Technical Symposium on Computer Science Education*, Dallas, EUA, 389-394.

Martinez, M., Gomez, M. J., Moresi, M., Benotti, L. (2016) “Lessons Learned on Computer Science Teachers Professional Development. In: *Proc. of the ACM Conference on Innovation and Technology in Computer Science Education*, Arequipa, Peru, 77-82.

Mishra, P., Koehler, M. J. (2006) “Technological Pedagogical Content Knowledge: A Framework for Teacher Knowledge”. *Teachers College Record*, 108(6), 017-1054.

SBC (2018) “Diretrizes para ensino de computação na Educação Básica”. Disponível em: <http://www.sbc.org.br/documentos-da-sbc/send/131-curriculos-de-referencia/1177-diretrizes-para-ensino-de-computacao-na-educacao-basica>.

Tissenbaum, M., Sheldon, J., Abelson, H. (2019) “From Computational Thinking to Computational Action”. *ACM Interactions*, 62(3), 34-36.

Wing, J. M. (2006) “Computational thinking”. *Communications of the ACM*, 49(3), 33.

Yin, R. K. (2009) “Case Study Research: Design and Methods”. SAGE, 4 ed.