

# A programação de jogos como um instrumento motivador da aprendizagem

## *Game programming as a motivating instrument of learning*

Sergio Crespo Coelho da Silva Pinto\*

Marcelo Simas Mattos\*\*

### Resumo

Este artigo descreve um modelo de aprendizagem que usa programação de jogos para telefones celulares como uma alternativa no processo de ensino-aprendizagem em disciplinas escolares. A aplicação do modelo tem em vista contribuir para a motivação, o engajamento e a aprendizagem dos estudantes por meio do desenvolvimento do pensamento computacional. Durante a pesquisa, foi desenvolvida uma arquitetura pedagógica deste modelo, nela se propõe que os estudantes realizem melhorias sucessivas em um jogo a partir de uma versão inicial. Aplicou-se a arquitetura por intermédio de experimentos com alunos do ensino médio, nos quais eles desenvolviam jogos digitais que envolviam conceitos de matemática. Este trabalho foi realizado de forma interdisciplinar, envolvendo docentes de Informática e Matemática. Para a programação, escolheu-se uma linguagem de programação visual baseada em blocos do ambiente *MIT App Inventor 2*, de modo a abstrair a complexidade das linguagens de programação tradicionais, como as linguagens C, C++, Pascal, Java e JavaScript. Para concretizar as melhorias, os estudantes foram incentivados a realizarem a programação em atendimento ao que foi especificado pelos docentes. Os resultados obtidos com a aplicação do modelo e a sua investigação indicam o seu uso como recurso didático contextualizado com o cotidiano dos estudantes do ensino médio.

*Palavras-chave:* Pensamento computacional. Programação. Programação em jogos.

### Abstract

This paper describes a learning model that uses game programming for mobile phones as an alternative in the teaching-learning process in school subjects. The application of the model aims to contribute to the motivation, engagement and learning of students through the development of computational thinking. During the research, a pedagogical architecture of this model was developed, in which it is proposed that the students make successive improvements in a game from an initial version. The architecture was applied through experiments with Brazilian High School students in which they developed digital games that involved concepts of Mathematics. This work was carried out in an interdisciplinary way involving teachers of Computer Science and Mathematics. For programming, we chose a block-based visual programming language from the MIT App Inventor 2 environment to abstract the complexity of traditional programming languages such as C, C++, Pascal, Java, and JavaScript. To make improvements, the students were encouraged to carry out the programming in compliance with what was specified by the teachers. The results obtained with the application of the model and its investigation indicate its use with a didactic resource contextualized with the daily life of Brazilian high school students.

*Keywords:* Computational thinking. Programming. Programming in games.

Recebido em 28/09/2018 – Aprovado em 12/12/2018

<http://dx.doi.org/10.5335/rep.v26i2.8692>

\* Doutor em Informática pela Pontifícia Universidade Católica do Rio de Janeiro. Professor convidado no Programa de Mestrado em TICs na Universidade Tecnológica do Panamá. Professor da Universidade Federal Fluminense, Rio de Janeiro, Brasil. E-mail: [screspo@id.uff.br](mailto:screspo@id.uff.br)

\*\* Mestre em Engenharia de Produção e Sistemas Computacionais pela Universidade Federal Fluminense. Professor do Instituto Federal de Educação, Ciência e Tecnologia do Rio de Janeiro, *Campus Arraial do Cabo*, Brasil. E-mail: [marcelo.mattos@ifrj.edu.br](mailto:marcelo.mattos@ifrj.edu.br)

## Contextualização

Alguns dados sobre a educação escolar no Brasil indicam a necessidade de esforços continuados por todos aqueles envolvidos com a educação no país, incluindo governos, universidades e profissionais da educação. Para ilustrar a situação, os dados extraídos do Programme for International Student Assessment (Pisa) – Programa Internacional de Avaliação de Estudantes –, avaliação aplicada em vários países a estudantes na faixa dos 15 anos, desenvolvida e coordenada pela Organização para Cooperação e Desenvolvimento Econômico (OCDE), apresentaram, em 2015, em um *ranking* com 70 países/economias participantes, os seguintes resultados para o Brasil: em matemática, 65º lugar, com 377 pontos; em leitura, 59º lugar, com 407 pontos; e em ciências, 63º lugar, com 401 pontos (PISA, 2015).

Em relação ao ensino médio no Brasil (nível de ensino foco desta pesquisa), as taxas de reprovação apresentam índices considerados altos, se comparados aos de outros países. Dados de 2014, do Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (Inep), autarquia federal vinculada ao Ministério da Educação (MEC), apontaram uma taxa de reprovação total de 12,1% no ensino médio. A rede de ensino pública apresentou uma taxa de 13,1% e a rede particular 5,5%; comparando as redes municipal, estadual e federal, as taxas foram, respectivamente, 11,3%, 13,1% e 13,1%. Desses resultados, pode-se depreender que as taxas de reprovação na rede pública foram significativamente superiores às da rede particular, e comparando as redes municipal, estadual e federal, os valores das taxas foram próximos (INEP, 2014).

Para provocar melhorias dos indicadores da educação básica no país, algumas propostas vêm sendo estudadas e aplicadas, inclusive com uso de ferramentas tecnológicas, com vistas a auxiliar o processo de ensino-aprendizagem. Entre as iniciativas, no nível governamental, podem-se citar o Programa Nacional de Informática na Educação (Proinfo), o Programa um Computador por Aluno (Prouca) e o Programa Banda Larga nas Escolas (PBLE) (FUNDO NACIONAL DE DESENVOLVIMENTO DA EDUCAÇÃO, 2016).

De um modo geral, observa-se que na educação básica esses recursos tecnológicos são trabalhados focando no ensino da utilização da tecnologia da informação (TI). Segundo Wangenheim, Nunes e Santos (2014, p. 116), essa forma de ensino não é mais suficiente, é necessário ensinar a proficiência digital (*IT fluency*), “[...] acrescentando a capacidade de aprender e aplicar as novas tecnologias de forma produtiva ao longo da vida profissional/pessoal”. Nesse sentido, uma forma de pro-

mover fluência digital pode se dar por meio do aprendizado de programação, estimulando-se a aprendizagem por meio do pensamento computacional.

Este artigo apresenta um estudo de caso em uma instituição de ensino pública, seguindo a linha de programar jogos para estimular o aprendizado. É proposto um modelo que faz uso de recursos tecnológicos como uma alternativa para apoiar o ensino de disciplinas escolares por meio de atividades de programação em dispositivos móveis, no caso, em celulares.

## Introdução

O ensino de programação nas escolas de ensino fundamental e ensino médio se apresenta como uma alternativa para motivar e engajar o aluno no processo de ensino-aprendizagem, tornando-o mais criativo e estimulante (SCAICO et al., 2013; SILVA et al., 2015). No entanto, em um sentido mais amplo, há que se considerar o que Wing (2006) descreve como pensamento computacional. Segundo o autor, esse tipo de pensamento é necessário para a aprendizagem de programação e pode se oferecer como uma alternativa para desenvolver habilidades de leitura, escrita e aritmética em jovens. Ele ainda descreve, em linhas gerais, que o pensamento computacional envolve resolver problemas, conceber sistemas e compreender o comportamento humano, que se desenham como conceitos fundamentais para a Ciência da Computação.

Os conceitos de Ciência da Computação vêm influenciando outras áreas, como por exemplo: aprendizagem de máquina na Estatística; conceitos de Ciência da Computação na Biologia; *nanocomputing*<sup>1</sup> na Química; e computação quântica na Física (WING, 2006). Entretanto, a computação, geralmente, não é integrada aos currículos formais escolares (RODRIGUEZ et al., 2015; HINTERHOLZ; CRUZ, 2015).

A área da educação escolar também vem sofrendo mudanças significativas, ocasionadas pela influência da Ciência da Computação. Trabalhos recentes em âmbito nacional vêm propondo a introdução de conceitos e práticas próprios da Ciência da Computação em disciplinas escolares regulares da educação básica (FERREIRA et al., 2015; FRANÇA; AMARAL, 2013; HINTERHOLZ; CRUZ, 2015), incentivando os alunos a desenvolverem o pensamento computacional. Desses estudos, em se tratando especificamente do ensino de programação na educação escolar, há indicação de níveis mais altos de motivação e empenho dos estudantes com essas experiências em suas aprendizagens. Contudo, a revisão sistemática realizada por

Silva et al. (2015) alerta que, para que se tenha uma aplicação efetiva e em escala no ensino básico, ainda são necessárias mais pesquisas, assim como o desenvolvimento de novas abordagens.

Uma abordagem que vem sendo usada, nestes ambientes, é a programação de jogos. Entre os motivos desse enfoque estão o interesse dos jovens nos *videogames* (AL-BOW et al., 2009; MURATET et al., 2009) e o fato de a criação de jogos envolver muitos aspectos de computação, incluindo computação gráfica, inteligência artificial, interação homem-máquina, segurança, programação distribuída, simulação e engenharia de *software* (OVERMARS, 2004).

O uso de dispositivos móveis, presente na investigação do modelo proposto, justifica-se pelo fato de que o público-alvo desta pesquisa são alunos do ensino médio, portanto se buscou utilizar algum equipamento tecnológico que estivesse presente no dia a dia desses estudantes. Dados estatísticos, desde 2011, comprovam essa presença, segundo a Pesquisa Nacional por Amostra de Domicílios (PNAD), do Instituto Brasileiro de Geografia e Estatística (IBGE), tem-se também que, desde 2013, os dados mostram que o consumo de eletrônicos, basicamente computador e telefone celular, teve crescimento em todas as faixas de idades e em todas as regiões do país. Ainda, 41,9% das crianças e jovens entre 10 e 14 anos tinham celular, o que totalizava 10,9 milhões, com crescimento a uma taxa de 43%. Os jovens adultos, entre 25 e 29 anos, eram os maiores consumidores de celulares (83,1%) (SCHWARTZ, 2013). Atualmente, dados da Anatel, segundo Teleco (2018), indicam que o Brasil terminou julho de 2018 com 234,7 milhões de celulares e densidade de 112,21 cel/100 hab (celulares por 100 habitantes).

## Revisão da literatura

A revisão de literatura realizada em fontes nacionais teve como principal objetivo descobrir o que vem sendo pesquisado, no país, sobre o uso de programação como apoio ao aprendizado. Além disso, os resultados da pesquisa auxiliaram na proposição do modelo pedagógico.

Foram pesquisadas fontes nacionais ligadas à área de informática na educação, para tanto, foram aplicados filtros de pesquisa de modo a buscar publicações recentes que abordavam o assunto: uso de programação como apoio ao aprendizado. Dessa forma, buscaram-se publicações dos últimos seis anos (2011 a 2017), na *Revista Brasileira de Informática na Educação* (RBIE) e na *Revista Novas Tecno-*

*logias na Educação* (Renote), assim como nos três últimos anos no Workshop de Informática na Escola (WIE).

No levantamento bibliográfico, foram selecionados estudos que propunham ensinar programação como recurso para o aprendizado de conceitos de disciplinas escolares ou para o aprendizado de programação no ensino superior. Com esse levantamento, esperava-se conhecer as iniciativas nacionais que vêm sendo adotadas como alternativas às formas tradicionalmente usadas para o ensino de programação.

Nas fontes Renote e RBIE, foram usados os mesmos termos de busca: (educação OR ensino) AND (“programação de computadores” OR “linguagem de programação” OR programação). Para escopo de busca, escolheu-se “todos”, configurando-se para buscar artigos do período de 2011 a 2017. A ferramenta de busca da Renote retornou 21 artigos e a da RBIE, 16 artigos. Fazendo a leitura do título e do resumo, nos resultados destas duas revistas, foram selecionados, respectivamente, 18 e 14 artigos. Seguiu-se para a fase de inclusão ou exclusão pelo critério já relatado, de modo que foram excluídos: um artigo da Renote e dois artigos da RBIE. Sendo assim, a pesquisa teve um total de 29 artigos selecionados.

Na fonte WIE, foram selecionados artigos de 2012 a 2016. Primeiramente, realizou-se a seleção por meio da leitura dos títulos e dos resumos. Na sequência, foi usado o mesmo critério de inclusão e exclusão adotado nas revistas. Com isso, foram extraídos 19 artigos na fonte WIE.

## Questões a serem verificadas

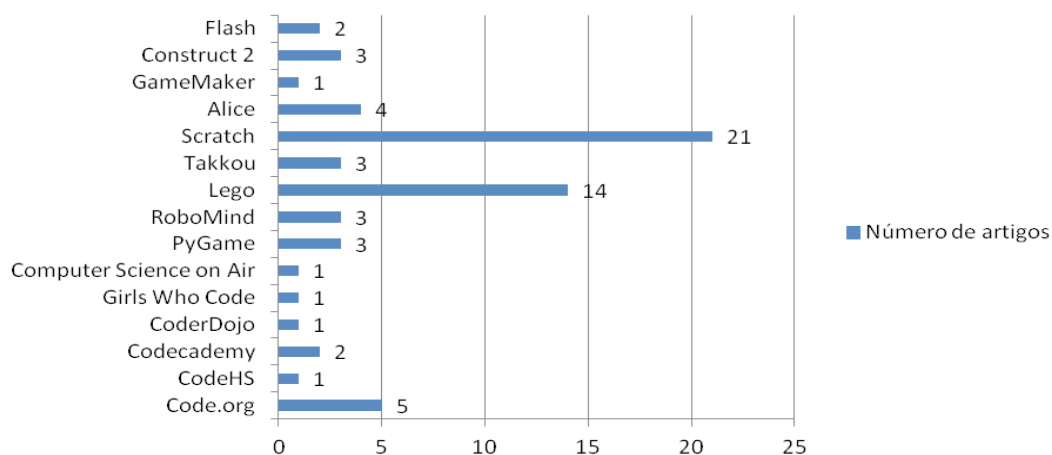
A extração de informações foi voltada para se descobrir quais recursos de programação foram citados e em que se aplicaram, resultando nas seguintes perguntas:

- Questão 1 – Quais recursos de programação foram citados nos textos pesquisados?
- Questão 2 – Quantos artigos investigaram a aplicação em cursos, oficinas ou projetos para cada recurso extraído da questão 1? E quais foram os recursos usados nesses trabalhos?
- Questão 3 – Em quais disciplinas do currículo escolar os recursos de programação foram aplicados?

## Resultados da pesquisa bibliográfica

Alguns recursos e iniciativas citados foram: Code.org, CodeHS, Codecademy, CoderDojo, Girls Who Code, Black Girls Code, Computer Science on Air, PyGame, RoboMind, Lego Mindstorms, Takkou, Scratch, Alice, GameMaker, Construct 2 e Flash. Realizando a contagem de cada artigo em que se mencionou os recursos, obteve-se os números ilustrados na Figura 1.

Figura 1 – Número de artigos que fazem menção a cada um dos recursos

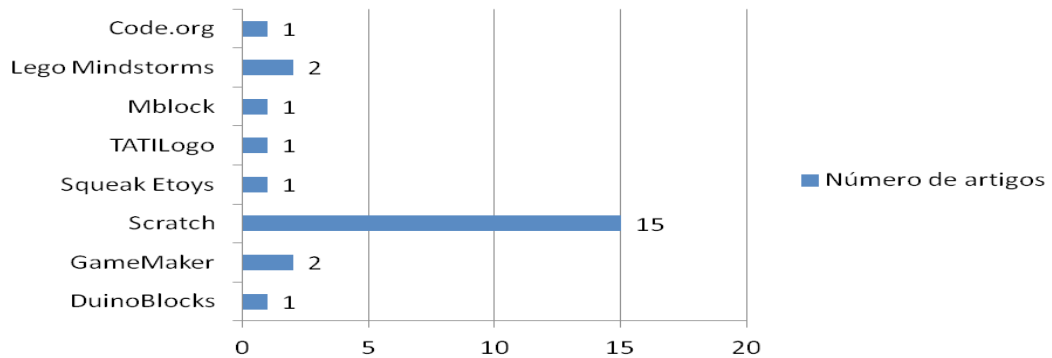


Fonte: elaboração dos autores.

Considerando os 48 artigos analisados, verificou-se que a maioria faz referência ao Scratch, identificado em 21 (43,8%), seguido por Lego, em 14 (29,2%), e Code.org, em 5 (10,4%). Em Lessa et al. (2015), foram encontrados resultados semelhantes, contudo, os autores pesquisaram apenas na fonte WIE e buscaram quais ferramentas estavam sendo usadas em “programação de computadores” e em “robótica educativa” na escola.

Respondendo à Questão 2, a maioria dos artigos que investigaram a aplicação dos recursos de programação fizeram uso do Scratch, foram 15 artigos (31,3%), seguido do Lego Mindstorms e do GameMaker, com 2 artigos cada (4,2%). Pode-se identificar, na Figura 2, que, entre ambientes de programação ligados à robótica educacional, os números estão diluídos entre a ferramenta comercial Lego Mindstorms e as não comerciais: Mblock, Squeak Etoys e DuinoBlocks (ALVES; SAMPAIO; ELIA, 2014).

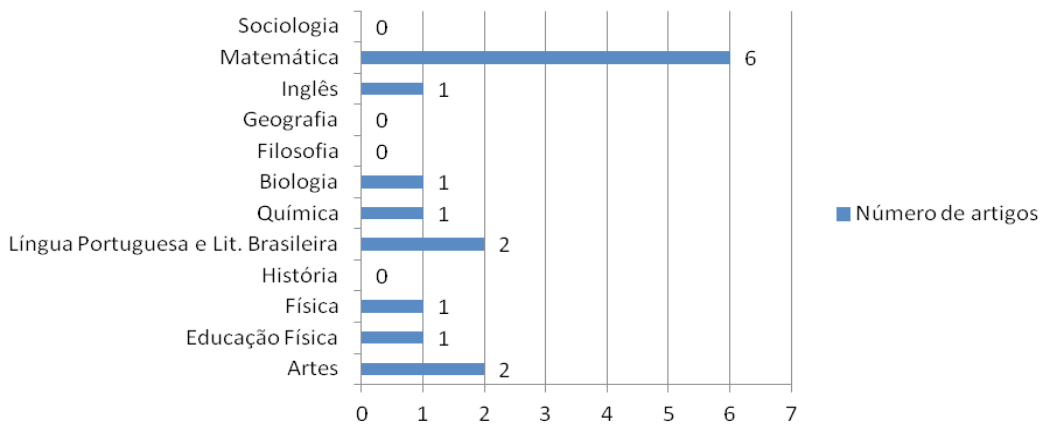
Figura 2 – Artigos que investigaram a aplicação dos recursos de programação



Fonte: elaboração dos autores.

As informações sobre os artigos em que os autores definiram claramente a disciplina escolar em que estava inserido o recurso estão ilustradas na Figura 3. Pode-se constatar que a maioria foi aplicada em Matemática (6 artigos), seguida de Artes e Língua Portuguesa e Literatura Brasileira (2 artigos cada), e Inglês, Biologia, Química, Física e Educação Física (1 artigo cada). As disciplinas Sociologia, Geografia, Filosofia e História não foram identificadas nos artigos.

Figura 3 – Aplicação de recursos em disciplinas escolares



Fonte: elaboração dos autores.

O resultado da Figura 3 indica que houve pesquisas em várias disciplinas escolares nos últimos anos, porém, em pequena quantidade em algumas áreas ou inexistente em outras. Esses resultados reforçam a necessidade de que mais pesquisas

sejam realizadas nessa área. Os resultados confirmam, também, que as iniciativas de uso de recursos de computação para apoio ao aprendizado são escassas no país, o que indica a relevância do presente estudo como contribuição para a difusão de conhecimentos sobre o ensino de programação na educação básica no Brasil.

Pelos estudos realizados, optou-se pela escolha do recurso de programação MIT App Inventor 2, sobretudo, porque o ambiente usa a mesma metáfora de linguagem gráfica baseada em blocos do Scratch (recurso mais citado na presente revisão de literatura). Contudo, poder-se-ia pensar em utilizar o próprio Scratch, mas esse ambiente não é voltado à criação de aplicativos ou jogos para dispositivos móveis, ao contrário do MIT App Inventor 2, que é inteiramente voltado para a criação de aplicativos para esses dispositivos.

Durante a pesquisa, na busca de trabalhos que se utilizavam de novos recursos de ensino de programação voltados especificamente para dispositivos móveis, chegou-se ao MIT App Inventor 2. É importante destacar que, na época em que a revisão de literatura foi realizada, não foram encontrados artigos que citavam este ambiente utilizando-se os termos de busca da revisão de literatura. Entretanto, em pesquisa posterior, especificando-se o termo “App Inventor”, nas três revistas pesquisadas e para o mesmo período, chegou-se ao artigo publicado por Finizola et al. (2014), que relata uma experiência com um curso de ensino de programação utilizando a plataforma MIT App Inventor com alunos do ensino médio.

As leituras dos artigos selecionados nesta revisão de literatura e das publicações obtidas por buscas diretas em fontes nacionais e internacionais, realizadas durante os estudos dos referenciais teóricos, ajudaram na proposição do modelo de aprendizagem (Figura 4), o qual se baseia na fundamentação teórica sobre aprendizagem apresentada na próxima seção.

## Fundamentação teórica sobre aprendizagem

As teorias e metodologias previstas para esta pesquisa são: teoria de aprendizagem socioconstrutivista, *problem-based learning* (PBL) – aprendizagem baseada em problemas – e programação de jogos (ou programação com temática de *games*). A união delas serviu para instigar, engajar e motivar os alunos, além de contribuir para suas aprendizagens. Sem a intenção de explorar de modo aprofundado estas teorias e metodologias, seguem, nos próximos parágrafos, algumas definições importantes para a pesquisa.



O socioconstrutivismo é derivado de duas teorias, o Construtivismo, de Jean Piaget, e a Teoria de Aprendizagem, de Lev Vygotski. Para Piaget, o indivíduo constrói ativamente o conhecimento, pela interação com o meio. Ele defende que as crianças devem ter a oportunidade de descobrir e inventar as coisas por conta própria, para, então, entendê-las. Enquanto Vygotski propõe que, além da interação com o meio, a interação com os semelhantes também é essencial para a aprendizagem. Em sala de aula, o socioconstrutivismo se configura como uma teoria que considera o potencial do indivíduo na construção de seu próprio conhecimento, valorizando a sua interação com o ambiente e também com outros indivíduos, no caso, professores e colegas (BISSOLOTTI; NOGUEIRA; PEREIRA, 2014).

As ideias de Piaget chegaram ao Brasil na década de 1920, vinculadas ao contexto do Movimento da Escola Nova (NIEMANN; BRANDOLI, 2012; SANCHIS; MAHFOUD, 2010). Sanchis e Mahfoud (2010) descrevem que, ao longo das décadas, houve várias etapas quanto ao tipo de apropriação das ideias de Piaget, em vários lugares do país, até chegar à década de 1990, quando despontou maior interesse pelo estudo de natureza epistemológica, destacando-se pesquisas com temas como: desenvolvimento moral, cognição e informática, afetividade e inteligência, linguagem e pensamento, cultura e cognição. Os autores destacam que as transformações da teoria do construtivismo de Piaget para o campo da pedagogia são muitos e variados, e que isso se deve, em parte, à escolha de partes ou conceitos da teoria, como: os estágios de desenvolvimento, a capacidade de estruturação operatória, as provas operatórias ou o princípio da atividade autoestruturante.

Entre os estudos que investigam a programação em cursos introdutórios, existem abordagens que se utilizam da criação de *games*. Esses estudos têm indicado que essa investida tem se mostrado promissora em relação ao forte engajamento dos alunos e ao aumento do interesse em campos de estudos relacionados à computação. Essas investigações, de uma forma geral, relatam experiências teórico-empíricas, nas quais os estudantes criam programas que implementam as características dos jogos digitais (AL-BOW et al., 2009; LEUTENEGGER; EDGINGTON, 2007; KELLEHER; PAUSH, 2007; MURATET et al., 2009). Com essa abordagem, o aluno passa a ser um sujeito ativo na construção de seu conhecimento. Essa característica alude ao uso de ideias e aspectos pedagógicos construtivistas na utilização da programação de jogos.

No processo de programação para a construção dos jogos, os estudantes necessitam desempenhar algumas tarefas associadas à Ciência da Computação, para realizar suas funcionalidades, por exemplo: identificar e decompor o problema; de-

finir padrões para tarefas repetitivas, na criação de uma solução algorítmica que atenda a todos os níveis do jogo, que pode se dar pela criação de funções para tarefas repetitivas; praticar o *mode-debug*, para verificar e detectar eventuais erros de lógica no jogo; praticar o modo *run-time*, para verificar movimentos durante o modo de execução; *brainstorming*, na atividade de examinar a estratégia de pensamento de outros jogadores, comparando a sua solução com a de outros, e isso pode se dar por meio de debates sobre o funcionamento dos jogos (KAZIMOGLU et al., 2012).

Estudos recentes sobre o ensino de programação se utilizam dos conceitos do socioconstrutivismo. Entre eles estão os que seguem a abordagem da aprendizagem baseada em problemas. Na definição de PBL, Burguillo (2010) descreve que essa abordagem é uma estratégia instrucional centrada no estudante, na qual há trabalho colaborativo para resolução de problemas, o que se reflete em suas experiências. A aprendizagem é dirigida por fornecimento de problemas abertos, em que os alunos trabalham em pequenos grupos colaborativos e são encorajados a serem responsáveis por organizar o seu grupo, gerenciando o processo de aprendizagem com suporte de um tutor ou instrutor que desempenha o papel de mediador da aprendizagem (BURGUILLO, 2010).

Em relação ao método da PBL, Mohorovičić e Strčić (2011) trazem que ele se centra no próprio envolvimento dos alunos na resolução de problemas. Os autores descrevem que a abordagem gira em torno de problemas que os profissionais encontram em suas áreas diariamente e, ainda, pontuam que o desenvolvimento de pensamento de ordem superior, o conhecimento disciplinar e as habilidades práticas que os alunos enfrentam com a situação problema atribuem a eles um papel ativo de solucionador destes. Os autores indicam que eles variam de uma abordagem simples baseada em problemas, em que, normalmente, palestras são apresentadas, e os problemas são introduzidos para motivar os alunos e para demonstrar uma teoria, até modelos completos de PBL, em que os problemas orientam todo o processo de aprendizagem.

## Arquitetura pedagógica desenvolvida

Neste trabalho, aplica-se um modelo alicerçado em teorias pedagógicas e de aprendizagem descritas na seção 4, o que se pesquisou na revisão de literatura da seção 3, e explicitamente o uso de programação de jogos como apoio ao aprendizado de estudantes do ensino médio. Dessa forma, nesta seção, cada parte do modelo da arquitetura pedagógica é apresentada com as devidas justificativas. É explicada a

ideia do modelo, de modo a servir de referencial para a sua aplicação e, também, para servir como ponto de apoio enquanto proposta pedagógica.

As duas teorias de aprendizagens que alicerçam o modelo são: o socioconstrutivismo e a PBL. Seguindo essas linhas de pensamento e as ideias do presente modelo, ajuizou-se que, como ponto inicial para a aplicação do modelo, o professor deve solicitar que os alunos se dividam em grupos. A finalidade é que eles se organizem para decidirem sobre a programação dos jogos e sobre a codificação dos programas. Com isso, cria-se um ambiente propício à invenção e à construção do conhecimento a respeito de conceitos de programação e de disciplinas escolares.

Trabalhou-se a construção de jogos por meio de codificação de versões, partindo de programações simples nas versões iniciais até versões com mais complexidades, sempre se trabalhando no mesmo jogo.

Na programação, para a solução das versões, os alunos precisam interagir entre si nos respectivos grupos, a fim de chegarem à melhor solução, dentro de um tempo pré-estabelecido pelo docente. Neste ponto, também a ação do docente é primordial, pois, além das explicações e do acompanhamento dos trabalhos, ele pode fazer o papel de mediador dos grupos.

Especificamente, em relação às versões para a criação dos jogos, a ideia é partir de um problema inicial (PI) simples, e, a cada nova versão, novos requisitos são apresentados aos alunos, aumentando assim as funcionalidades do jogo. Sugere-se que o professor pense em como abranger a disciplina escolar alvo desta abordagem no momento de criar o PI e os requisitos.

Mas, para se pensar tanto na versão inicial do jogo quanto nas posteriores, há que se pensar que, além de abranger a disciplina, ele deve possuir características e elementos peculiares aos jogos, que estão diretamente relacionadas a desejos humanos – por exemplo, pontos, níveis, desafios, placares (*rankings*) e presentes. Pontos seriam recompensas cumulativas recebidas por atividades desempenhadas que poderiam servir como moeda de troca para receber premiações; níveis corresponderiam ao progresso do usuário, em geral, em atividades com grau de dificuldade crescente; desafios seriam instruções ao usuário sobre realizações estipuladas dentro de um sistema; placares teriam o propósito de comparar o progresso dos usuários envolvidos e estabelecer classificações para promover competição; presentes corresponderiam a recompensas por realizações ou conquistas do usuário (KLOCK et al., 2014). Sendo assim, é imperioso a este modelo que o jogo concebido para aplicar aos alunos tenha todos esses elementos ou parte deles. Como sugestão, ainda pensando em comportamento humano e jogos, o professor pode pedir,

nos requisitos, além do atendimento a estes elementos, a alteração da aparência do jogo, de modo que o estudante possa personalizá-lo.

Ainda em se tratando das versões, há que se destacar que se utilizou a ideia da PBL, porém, como diferencial, o problema é acompanhado de uma solução inicial fornecida pelo professor. Para tanto, indica-se que o docente deve fazer uma apresentação e uma explicação deste jogo inicial, o qual é denominado de Versão 0 (V0). Então, para a aplicação deste modelo, o docente deve criar a programação da versão inicial (V0), e aconselha-se a elaboração da descrição do problema. Sugere-se, ainda, que V0 tenha itens em aberto, como descrito em Burguillo (2010). Esses itens em aberto dizem respeito a apresentar a programação do jogo inicial sem alguns elementos essenciais, como, por exemplo, placares e vidas. Sendo assim, na versão V0, devem constar apenas algumas programações de características relativas aos jogos, como pontuações, por exemplo, deixando o jogo incompleto, sem algumas outras características, como vidas do jogador, por exemplo. As características faltantes (como no exemplo, as vidas do jogador), devem ser aperfeiçoadas e incrementadas pelos estudantes nas versões sucessivas, denominadas na apresentação deste modelo de V1, V2, Vn.

Os itens em aberto dão subsídios para a definição de requisitos que são aqui nomeados como R1, R2, Rn, estes devem ser solucionados pelos alunos em cada nova versão pretendida.

Como forma de acompanhar e avaliar os estudantes, idealizou-se que, a cada subproblema representado pelos requisitos R1, R2, Rn, para chegar a uma solução V1, V2, Vn, os estudantes devem trabalhar colaborativamente para fazer a programação e, uma vez atendido o requisito, submeter ao professor. Se uma solução não for satisfatória dentro do tempo estimado, o professor deve intervir para melhorar o entendimento dos alunos sobre o requisito, a fim de que eles melhorem o trabalho e submetam nova versão para avaliação. Agindo desta forma, o professor, além de exercer o papel de avaliador, também exerce a mediação durante os trabalhos dos alunos, esclarecendo dúvidas e propondo alternativas.

Em relação à aprendizagem, quando o grupo resolve todos os subproblemas, eles têm concluído o seu produto final e, assim, realizado uma série de processos mentais com conteúdos de programação de computadores e de disciplinas escolares. Isso foi constatado durante o desenvolvimento das versões.

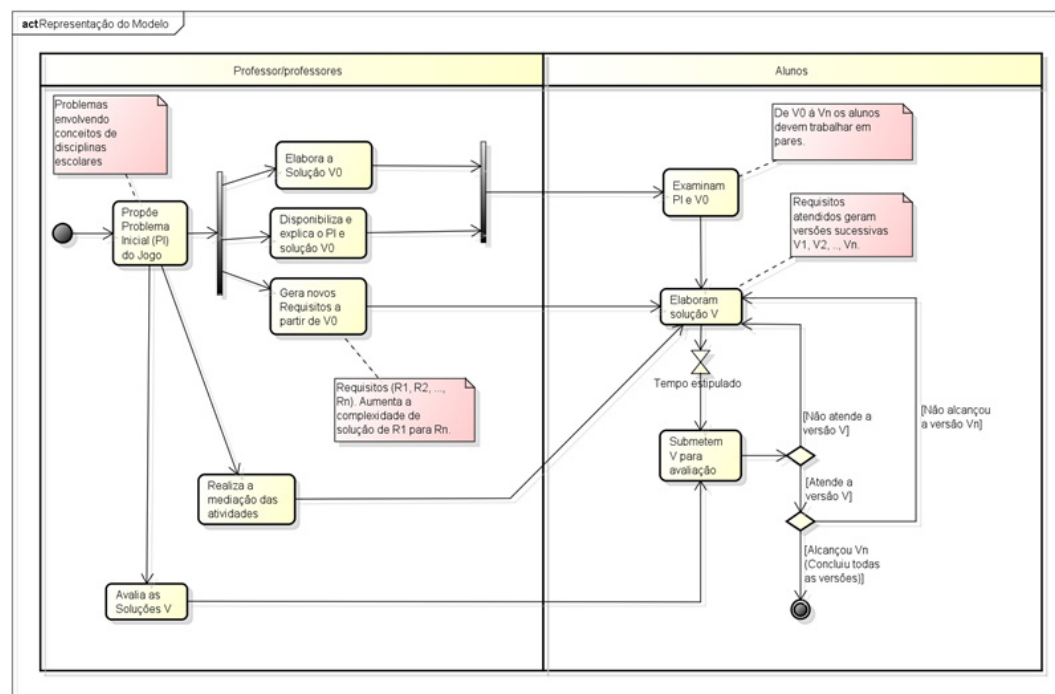
A Figura 4 representa graficamente – em um diagrama de atividades da UML (Unified Modeling Language) – todos os pensamentos descritos nos parágrafos anteriores. Essa representação se apresenta como o esquema do modelo pedagógico

investigado. Na Figura 4, do lado esquerdo, têm-se as atividades que devem ser realizadas pelo professor/professores (raia de natação Professor/professores); e, do lado direito, as atividades que cabem aos alunos (raia de natação Alunos).

Nesse diagrama, as atividades se iniciam no círculo presente na raia Professor/professores, que possui uma seta apontada para a atividade “Propõe problema inicial (PI) do jogo”. A ordem de leitura da sucessão de atividades é realizada de acordo com a orientação das setas. As duas barras verticais na raia Professor/professores são, da esquerda para a direita, os símbolos de bifurcação e união, respectivamente. A bifurcação representa que um único fluxo foi dividido em atividades simultâneas, no caso da Figura 4: “Elabora a solução V0”; “Disponibiliza e explica o PI e a solução V0”; e “Gera novos requisitos a partir de V0”.

A união junta atividades simultâneas e as reintroduz em um único fluxo de atividade. Na raia Alunos, o símbolo que está entre a atividade “Elaboram solução V” e “Submetem V para avaliação” representa o prazo que o professor deve estimar para que os alunos submetam uma solução para cada versão que estejam elaborando. Ainda na raia Aluno, os losangos são nós de decisão, os quais representam uma escolha entre dois ou mais fluxos. No caso do nó de decisão que recebeu o fluxo de “Submetem V para avaliação”, é decidido se os alunos devem refazer a versão que submeteram à avaliação, no caso do não cumprimento dos requisitos, ou se seguem para o próximo nó de decisão, para o caso de terem cumprido os requisitos. No outro nó de decisão, o que recebeu o fluxo “Atende a versão V”, é avaliado se a versão V é a última versão (Vn), se for, os alunos cumpriram todos os requisitos e o fluxo segue para o símbolo que representa o fim das atividades (símbolo com dois círculos concêntricos), caso contrário, os alunos devem elaborar a próxima versão, que no diagrama é a atividade que recebeu o fluxo “Não alcançou a versão Vn”.

Por fim, nesta apresentação do modelo pedagógico, indica-se que ele deva ser usado por um docente com conhecimentos de programação de computadores, ou ainda de forma interdisciplinar, envolvendo pelo menos um docente com conhecimentos de programação.



## Projeto do estudo de caso

O principal público da investigação do modelo consistiu nos alunos de 1º período do Curso Técnico em Informática Integrado ao Ensino Médio (TIEM) do *Campus Araraial do Cabo* do Instituto Federal de Educação, Ciência e Tecnologia do Rio de Janeiro (IFRJ/CAC). Especificamente, planejou-se a utilização de recursos para construção de jogos para dispositivos móveis nas aulas de Matemática I desses estudantes.

Foi usado o método experimental para verificação do modelo. Segundo Gil (2008, p. 16), este método “[...] consiste essencialmente em submeter objetos de estudo à influência de certas variáveis, em condições controladas e conhecidas pelo investigador, para observar os resultados que a variável produz no objeto”. Especificamente entre as condições controladas para o experimento: foram realizadas aulas com estudantes do ensino médio, preferencialmente com alunos não iniciados em cursos ou disciplinas de programação; foi escolhida uma disciplina, a Matemática; foi usado um ambiente de desenvolvimento de aplicativos para dispositivos móveis que não exige conhecimento prévio de programação; realizou-se um trabalho interdisciplinar. De modo a sistematizar o experimento, ele foi classifica-

do em dois tipos. O primeiro se refere às aulas nas quais os alunos participavam de modo voluntário. O segundo se refere à aplicação na disciplina de Matemática, diferenciando-se do primeiro pela obrigatoriedade da participação dos alunos.

Antes de submeter o experimento às aulas regulares da disciplina de Matemática I, primeiramente se realizaram estudos bibliográficos, presentes neste documento, que instigou a cunhar o modelo, e posteriormente foi criado um projeto interdisciplinar entre Matemática e Informática, para organizar a sua aplicação. Este foi denominado Desenvolvimento de Aplicativos como Ferramenta para o Ensino de Matemática (DAFEMat), projeto que envolveu dois professores da escola, um professor de Matemática e o outro de Informática.

## Limitações do experimento

Entre os fatores positivos da investigação do modelo, destaca-se que ele pôde ser verificado em três ocasiões, dando inclusive a possibilidade de validá-lo antes de aplicar em sala de aula e incluí-lo no processo de ensino-aprendizagem dos estudantes.

O experimento teve um percurso de investigação por meio de pesquisa bibliográfica e planejamento até a sua aplicação em sala de aula. Esta aplicação ocorreu durante um semestre na disciplina escolar de duas turmas – período de 08 de abril de 2017 até 08 de julho de 2017. Mas ela se limitou à aplicação em apenas um semestre letivo, uma investigação mais ampla poderia acompanhar o desempenho dos estudantes dessas turmas no seu percurso escolar, além de aplicar em outras turmas nos semestres seguintes. Outra limitação refere-se ao fato de que o presente experimento buscou identificar e aludir pontos com base nos estímulos e nas respostas dos estudantes pela aplicação do modelo. Contudo, as turmas investigadas não foram comparadas com uma turma controle.

## Experimentação do modelo da arquitetura pedagógica projetada

### Oficina

A validação do modelo se deu por meio de oficina e minicurso. A oficina foi um curso piloto de 15 horas, intitulado “Oficina de App Inventor 2”, que ocorreu entre 26 de janeiro e 23 fevereiro de 2017. A participação na oficina foi aberta aos alunos de 1º e 2º períodos, ficando definido que as aulas aconteceriam nas terças e

quintas-feiras, das 13h30 às 15h10, esse horário era o contraturno dos estudantes. A participação não era obrigatória e para formalizar a participação o estudante deveria fazer a inscrição.

Foi escolhido esse público-alvo na oficina por ele ser composto de alunos que não tiveram aulas de programação na instituição, que é o caso dos alunos de 1º período, e alunos que estavam iniciando em disciplina de programação naquele semestre, que é o caso dos alunos de 2º período.

## Minicurso

O minicurso ocorreu nos dias 3 e 4 de abril de 2017, sendo realizado de modo concomitante à Semana Acadêmica da instituição pesquisada. Ele foi aberto à participação de alunos e professores. Foi planejado trabalhar com aplicativos que pudessem ser realizados no período de tempo de 5 horas, duração prevista para o curso. Dessa forma, foi planejada a criação de dois aplicativos, o Talk to Me, disponível no *site* oficial do App Inventor 2, e um jogo pensado pelos professores e nomeado “Salve o banhista”, que tinha como propósito realizar salvamentos de um personagem representando um banhista em apuros no mar.

## Programando jogos com matemática e *matematicando* para jogar

Os jogos utilizados tiveram como tema conceitos de matemática do ensino médio. Para concebê-los, os professores de Informática e Matemática se reuniram, na fase de projeto do DAFEMat, para planejar como seriam os jogos a serem criados pelos alunos e como seriam as abordagens nas aulas. Foram três os jogos previstos para os alunos criarem: “Salve o banhista”, “Dory e os ângulos” e “Bola matemática”.

O planejamento dos três jogos foi pensado em consonância com a ementa da disciplina Matemática, do 1º período do curso TIEM. O Quadro 1 apresenta a ementa prevista para esses alunos. Em termos de organização da matriz curricular do curso, ela é nomeada como Matemática I. Ao longo dos sete períodos previstos na matriz curricular, os estudantes têm disciplinas de Matemática até o 6º período, e é prevista uma disciplina de matemática a cada período do curso, até o 6º. No Quadro 1, identifica-se que, na linha 2, identificada como “Ementa”, há separação de “Matemática I”, em que é previsto um resgate de conteúdos de matemática do ensino fundamental e de conteúdos de álgebra; já em “Matemática II”, há conteúdos de geometria.



## Quadro 1 – Ementa de Matemática I

Disciplina: Matemática I
EMENTA: Matemática I. Revisão e nivelamento. Transformação de decimal em fração. Operações com frações. Potenciação. Produtos notáveis. Radiciação. Operações com radicais. Racionalização de denominadores. Bases Numéricas. Ênfase em bases binárias. Lógica Matemática. Proposição Sentenças. Quantificadores. Conectivos. Negação lógica. Condicional e bicondicional. Tautologia. Conjuntos Numéricos. Matemática II. Lei Angular de Tales. Teorema de Tales. Teorema das Bissetrizes de um Triângulo. Semelhança (Homotetia). Teorema de Pitágoras. Problemas envolvendo o Teorema de Pitágoras. Triângulos Notáveis.

Fonte: quadro extraído do ementário do Curso Técnico em Informática Integrado ao Ensino Médio do IFRJ/CAC.

Destaca-se que, dentre os testes, na oficina e no minicurso, atingiram-se outros públicos (alunos do 2º período e um docente), mas, na aplicação em sala de aula, todos os estudantes eram do 1º período do curso TIEM. As duas aplicações iniciais (oficina e minicurso) serviram para se fazer ajustes para depois aplicar em um ambiente real de sala de aula, por isso, desde o início, pensou-se em conteúdos dirigidos aos alunos de 1º período.

## Jogos desenvolvidos


Dentre os jogos desenvolvidos, destaca-se um deles para exemplificar os conceitos. O jogo descrito a seguir foi concebido pelos professores especificamente para ser usado no DAFEMat. Ele foi pensado para abranger alguns conteúdos de Matemática I (Tabela 1), ao mesmo tempo em que visavam ao desenvolvimento do pensamento computacional durante a programação do jogo.

### A oficina do jogo “Bola matemática”

As principais características do jogo “Bola matemática” podem ser verificadas no Quadro 2, em que é mostrada a imagem inicial do jogo em sua Versão 0, a descrição e os principais conteúdos trabalhados. Na descrição, há o nome do jogo, o objetivo e a forma de se jogar. Na imagem da tela do jogo, as bolas identificadas como N, Z, Q e Ir representam as bolas que podem ser arremessadas pelo jogador por meio da ação arrastar na tela do dispositivo móvel, elas são correspondentes aos conjuntos numéricos dos Naturais, Inteiros, Racionais e Irracionais, respectivamente. A bola identificada como  $\sqrt{2}$  representa um número sorteado que aparecerá na área reservada para ele na tela do jogo. Ainda na imagem, além do nome (Bola matemática), consta a pontuação, que é incrementada em um a cada correspondência

certa entre o conjunto e o número sorteado (ela é conquistada quando há a colisão entre esses elementos), e um botão para sortear um número com a finalidade de se conquistar mais pontos.

Quadro 2 – Jogo “Bola matemática”

<p>Descrição do jogo</p> <p>Nome: Bola matemática</p> <p>Objetivo: arremessar e acertar bolas dos conjuntos numéricos para colidir no número que pertence à sua coleção.</p> <p>Como jogar: um número é sorteado aleatoriamente e surge na tela do jogo. O jogador arremessa a bola do conjunto para colidir com o número. A direção é definida pelo arrastar na tela. Ele ganhará pontos se o número estiver em sua coleção e perde se não houver correspondência. Para acertar o número, o jogador pode usar o recurso de “tabelar” nas bordas.</p>	
	<p>Principais conteúdos abordados:</p> <ul style="list-style-type: none"><li>- noções de programação;</li><li>- noções de lógica matemática;</li><li>- noções de trigonometria;</li><li>- definição e classificação de ângulos;</li><li>- porcentagem;</li><li>- equação de 1º grau;</li><li>- conjuntos numéricos</li></ul>

Fonte: elaboração dos autores.

Essas características e descrições foram pensadas, discutidas e elaboradas pelos dois professores da escola envolvidos no DAFEMat. Os conceitos estavam de acordo com a ementa da disciplina de Matemática dos alunos de 1º período do curso TIEM. Podem-se identificar aspectos interdisciplinares na descrição do Quadro 2, percebe-se que não há separação do que é conteúdo de matemática e o que é de computação. Todavia, os conteúdos de cada uma das disciplinas estão inseridos na construção do jogo.

Na oficina, foram 17 alunos inscritos, porém houve desistências, motivadas principalmente porque os alunos estavam em um período de provas, além disso, foi verificado que o contraturno pode ter dificultado a participação dos alunos. No primeiro dia, apenas 9 alunos se apresentaram (Tabela 1), e ao longo da oficina houve certa flutuação do número de cursistas a cada dia; no final, apenas 5 alunos

concluíram o curso. Contudo, mesmo com essas dificuldades, a realização do piloto serviu para validação e ajustes da proposta.

Tabela 1 – Flutuação do número de alunos na oficina de App Inventor

Data (ano 2017)	26/01	31/01	02/02	07/02	09/02	14/02	16/02	21/02	23/02
Número de alunos	9	11	7	6	4	6	2	5	5

Fonte: elaboração dos autores.

Os professores planejaram requisitos que deveriam ser cumpridos pelos estudantes após a criação da Versão 0, eles são apresentados a seguir, separados em versões de V1 até V6:

- a) V1 – o jogo deve possuir uma contagem de tempo, determine 120 segundos como limite para essa fase do jogo; quando uma bola estiver em movimento e for tocada, ela deve parar (esse requisito possibilita ao jogador corrigir um lançamento errado);
- b) V2 – aumente a quantidade de exemplos de números para cada conjunto, para isso, crie pelo menos mais 5 números por conjunto; altere a codificação para que a imagem que contém o número apareça em posições aleatórias na tela de pintura do App Inventor;
- c) V3 – ajuste o botão sortear para que ele se torne invisível logo após ser pressionado; essa medida é para forçar o jogador a realizar pelo menos um arremesso para o número sorteado; quando houver pelo menos uma colisão com o número, o botão deve se tornar visível novamente;
- d) V4 – acrescentar música de fundo e áudio para as colisões entre as Sprites; crie uma figura que represente a quantidade de vidas que o jogador possui; o jogo começa com 4 vidas, o jogador perde uma vida a cada erro; se zerar as vidas, deve ser exibida a mensagem “Game Over” (ou “Fim do Jogo”); introduza um áudio correspondente para essa situação; se o tempo se esgotar e o jogador atingir 25 acertos ou mais, deve ser exibida uma mensagem de que ele venceu a fase, caso contrário, deve ser exibida a mensagem de “Game Over” (ou “Fim de Jogo”); introduza um áudio correspondente para cada uma das situações;
- e) V5 – nesta versão, introduza mais dois números que devem surgir na tela em posições aleatórias; garanta que as imagens desses números não apareçam

sobrepostas umas às outras; crie barreiras que devem surgir em posições aleatórias na tela, para dificultar os acertos aos alvos (pode-se estipular que essas barreiras surjam depois de decorridos 60 segundos do jogo);

- f) V6 – crie uma tela para o nível 2 do jogo; elabore essa fase com características semelhantes ao do nível anterior, porém aumentando a dificuldade; como sugestão, acrescente outros conceitos sobre conjuntos.

Ao longo da aplicação, foi verificado que, se atingissem os quatro requisitos, os alunos teriam um jogo completo com funcionalidades principais, como: vidas, pontuação, aspectos aleatórios, tempo e conquista (“Venceu a Fase”) ou derrota (“Game Over”). Dessa forma, por questões de prazo para a criação do jogo “Bola matemática”, foi decidido que os estudantes terminariam o jogo quando cumprissem todos os requisitos até a versão 4.

As duas versões adicionais (V5 e V6) poderiam ser trabalhadas em um curso com duração maior e serviriam para estimular, ainda mais, a criatividade dos estudantes. Mas os docentes verificaram que, de acordo com a proposta, se os alunos atingissem a versão 4, eles já teriam construído um jogo com as principais características (como as mencionadas anteriormente), abrangeriam os conteúdos de matemática e também as noções de computação esperadas.

Próximo ao término da oficina, os cinco alunos que chegaram até o fim do curso responderam a um questionário sobre seus hábitos em relação ao uso de dispositivos móveis, sobre o curso e sobre o App Inventor 2.

Em relação aos dados gerais dos sujeitos pesquisados, 3 eram do sexo feminino e 2 do sexo masculino, e tinham em média 16 anos de idade. Nenhum deles havia concluído algum curso de programação anterior à oficina de App Inventor.

Sobre o uso de dispositivos móveis, todos responderam que possuíam *tablet* ou *smartphone*, com sistema operacional Android, e usavam mais frequentemente o seu dispositivo em casa. A média diária de uso dos dispositivos entre os pesquisados foi de 6,1 horas. Todos afirmaram usar o seu dispositivo móvel para estudar, com a seguinte frequência de uso: sempre (40%); frequentemente (40%); e eventualmente (20%); nenhum deles marcou raramente e nunca.

Entre os pesquisados, apenas um já conhecia o App Inventor 2 antes de realizar a oficina. Quanto à satisfação com a oficina, 60% ficaram muito satisfeitos e 40% satisfeitos (as opções não marcadas foram: pouco satisfeito, insatisfeito e muito insatisfeito). Quando questionados sobre o que gostavam no ambiente MIT App Inventor 2, as respostas foram: “Praticidade”; “A forma de codificar através de blocos e a praticidade na criação do design”; “A simplicidade”; “É simples e fácil de

usar”; “O fato de ser prático e versátil”. Pode-se depreender, com essas respostas, que os alunos destacaram os principais pontos fortes do App Inventor 2, sendo essas características importantes para um ambiente de desenvolvimento ser usado em cursos introdutórios de programação.

Quando perguntados sobre o que não gostavam no ambiente MIT App Inventor 2, as respostas foram: “O fato dele não ter muitos recursos como um programa superior”; “O emulador”; “Apesar de eu ter professores para me auxiliar, a falta de um ‘tutorial’ ou ‘guia de uso’ é ruim”; “Eu gosto de tudo. Ainda não identifiquei falhas”; “Nada”. Quanto à resposta que faz referência a tutorial ou guia de estudos, pode-se identificar a importância dos materiais didáticos para esse pesquisado. Contudo, possivelmente, o estudante sentiu falta desse material em língua portuguesa ou não se atentou aos materiais disponíveis no *site* do App Inventor 2, que estão em língua inglesa. Todos responderam ser muito importante para a aprendizagem deles o acompanhamento dos professores da escola durante o estudo no MIT App Inventor 2 (desde “É ruim para o meu aprendizado” até “É muito importante para o meu aprendizado”).

Quando foi solicitado para classificar os recursos de programação do App Inventor 2, todos classificaram como bom, numa escala na ordem do pior para o melhor, na seguinte sequência: inadequado, ruim, regular, bom e excelente. Por fim, quando perguntado se, na opinião dos alunos, a “Oficina de MIT App Inventor”, do projeto interdisciplinar de Informática e Matemática, estaria conseguindo integrar programação/informática com conceitos de Matemática, todos responderam que sim.

## Conclusões e trabalhos futuros

O artigo apresentou o uso de uma arquitetura com base em teorias pedagógicas e de aprendizagem que foi experimentado em aplicações com estudantes de um Curso Técnico em Informática Integrado ao Ensino Médio (TIEM). Os testes do experimento do modelo foram voltados à aplicação na disciplina de Matemática desses alunos e ocorreram em cursos livres e no contexto da sala de aula. Para tanto, dois professores da escola, um de Informática e o outro de Matemática, realizaram um trabalho interdisciplinar que, em síntese, constituiu-se em: estudos prévios (com base no modelo), aplicação e acompanhamento dos estudantes na realização dos testes.

As aplicações nos cursos livres serviram, principalmente, para realizar a validação do modelo antes de sua inserção no contexto da sala de aula. Nas aplicações inseridas na disciplina Matemática I, o modelo pôde ser experimentado em duas turmas do 1º período do curso TIEM. Essas aplicações foram inseridas dentro do projeto Desenvolvimento de Aplicativos como Ferramenta para o Ensino de Matemática (DAFEMat), concebido pelos professores envolvidos. Os docentes puderam vincular as atividades relativas à investigação do modelo ao DAFEMat.

Todos os testes indicaram a adequação do modelo enquanto recurso didático de ensino-aprendizagem contextualizado com o cotidiano dos estudantes de ensino médio. Fatos constatados, principalmente, pelo nível de satisfação dos pesquisados na Oficina de App Inventor 2, que alcançou 60% de “muito satisfeitos” e 40% de “satisfeitos”; também, pela resposta à pergunta sobre satisfação com as atividades interdisciplinares entre Informática e Matemática do DAFEMat (respostas ao questionário ligado à aplicação em sala de aula), os alunos se sentiram: satisfeitos (24, 70,6%), muito satisfeitos (8, 23,5%) e pouco satisfeitos (2, 5,9%). Além disso, ainda em se tratando da adequação enquanto recurso didático, os estudantes identificaram a interdisciplinaridade entre programação/informática com conceitos de matemática (32, 94,1%).

O modelo se apresentou como proposta pedagógica favorável para servir de apoio ao aprendizado em disciplinas escolares. Isso porque se verificou que ele pode ser adotado como recurso para trabalhar de maneira interdisciplinar, inserido nas atividades de disciplinas escolares. Índícios disso estão nos resultados da aplicação do DAFEMat, nos questionários e na aplicação de uma prova na disciplina de Matemática I.

Foi verificado que a maioria dos pesquisados, no questionário aplicado aos alunos da disciplina Matemática I, acreditava que as atividades que envolviam a aplicação do modelo foram úteis para o aprendizado de matemática e informática. Em resposta aos dois questionamentos que tratam desta questão, a maioria dos pesquisados afirmou que as atividades de se melhorar o jogo a cada versão e o trabalho em duplas – previstos no modelo – foram úteis para o aprendizado de matemática e informática. Nas respostas sobre o desenvolvimento dos jogos em duplas, destacam-se alguns relatos negativos, em que as justificativas remetem a dificuldades em circunstâncias relativas à negociação e ao comprometimento no trabalho em equipe. Portanto, uma atenção especial deve ser dada a essas circunstâncias, para que a aplicação do modelo realmente traga benefícios aos participantes.

A interdisciplinaridade foi um aspecto importante na aplicação do modelo. Os relatos das aulas, as respostas dos alunos aos questionários e a própria aplicação de uma prova da disciplina Matemática I sugerem a possibilidade de utilização do modelo de forma interdisciplinar, envolvendo outras disciplinas escolares.

Como trabalhos futuros, sugere-se o planejamento de jogos com temas ligados a outras disciplinas escolares e com a adoção do modelo apresentado, para verificação de sua adequação como recurso de apoio à aprendizagem no ensino médio para diferentes áreas do conhecimento: Linguagens, Códigos e suas Tecnologias; Ciências da Natureza e suas Tecnologias; Ciências Humanas e suas Tecnologias.

## Nota

<sup>1</sup> *Nanocomputing* (nanocomputação) – descreve a computação que usa dispositivos extremamente pequenos ou em nanoescala, nesta se tem que um nanômetro (nm) equivale a um bilionésimo de metro.

## Referências

AL-BOW, M. et al. Using game creation for teaching computer programming to high school students and teachers. In: ITICSE '09; ANNUAL ACM SIGCSE CONFERENCE ON INNOVATION AND TECHNOLOGY IN COMPUTER SCIENCE EDUCATION, 14. *Proceedings...* New York: ACM, 2009. v. 41. n. 3. p. 104-108.

ALVES, R. M.; SAMPAIO, F. F.; ELIA, M. F. DuinoBlocks: desenho e implementação de um ambiente de programação visual para robótica educacional. *Revista Brasileira de Informática na Educação*, Rio de Janeiro, v. 22, n. 3, p. 126-140, 2014.

BISSOLOTTI, K.; NOGUEIRA, H.; PEREIRA, A. T. C. Potencialidades das mídias sociais e da gamificação na educação a distância. *Revista Novas Tecnologias na Educação*, Joinville, v. 12, n. 2, p. 1-11, dez. 2014.

BURGUILLO, J. Using game theory and Competition-based Learning to stimulate student motivation and performance. *Computers & Education*, [S. l.], v. 15, p. 566-575, 2010.

FERREIRA, A. C. et al. Experiência prática interdisciplinar do raciocínio computacional em atividades de computação desplugada na educação básica. In: WORKSHOP DE INFORMÁTICA NA ESCOLA, 21, 2015. *Anais...* Maceió: Sociedade Brasileira de Computação, 2015. p. 256-265.

FINIZOLA, A. B. et al. O ensino de programação para dispositivos móveis utilizando o MIT-App Inventor com alunos do ensino médio. In: WORKSHOP DE INFORMÁTICA NA ESCOLA, 20. *Anais eletrônicos...* Dourados: Sociedade Brasileira de Computação, 2014. p. 337-341.

FUNDO NACIONAL DE DESENVOLVIMENTO DA EDUCAÇÃO. *Proinfo*. 2016. Disponível em: <<http://www.fnede.gov.br/programas/programa-nacional-de-tecnologia-educacional-proinfo>>. Acesso em: 25 jul. 2016.

FRANÇA, R. S.; AMARAL, H. J. C. Proposta Metodológica de Ensino e Avaliação para o Desenvolvimento do Pensamento Computacional com o Uso do Scratch. In: Workshop de Informática na Escola, 19, 2013. *Anais eletrônicos...* Campinas: Sociedade Brasileira de Computação, 2013. p. 179-188.

GIL, A. C. *Métodos e técnicas de pesquisa social*. 6. ed. São Paulo: Atlas, 2008.

INSTITUTO NACIONAL DE ESTUDOS E PESQUISAS EDUCACIONAIS ANÍSIO TEIXEIRA. *Indicadores Educacionais: taxas de rendimento*. 2014. Disponível em: <<http://portal.inep.gov.br/indicadores-educacionais>>. Acesso em: 20 jul. 2016.

KAZIMOGLU, C. et al. Learning programming at the computational thinking level via digital game-play. *Procedia Computer Science*, Omaha, Nebraska, USA, v. 9, n. 0, p. 522-531, 2012.

KELLEHER, C.; PAUSH. Using Storytrlling to motivate programming. *Communication of the ACM*, New York, v. 50, n. 7, p. 58-64, 2007.

KLOCK, A.C.T. et al. Análise das técnicas de gamificação em ambientes virtuais de aprendizagem. *Revista Novas Tecnologias na Educação*, Joinville, v. 12, n. 2, p. 1-10, dez. 2014.

HINTERHOLZ, L. T.; CRUZ, M. E. K. Desenvolvimento do pensamento computacional: um relato de atividade junto ao ensino médio, através do Estágio Supervisionado em Computação III. In: WORKSHOP DE INFORMÁTICA NA ESCOLA, 21, 2015. *Anais eletrônicos...* Maceió: Sociedade Brasileira de Computação, 2015. p. 137-146.

LESSA, V. et al. Programação de computadores e robótica educativa na escola: tendências evidenciadas nas produções do Workshop de Informática na Escola. In: WORKSHOP DE INFORMÁTICA NA ESCOLA, 21, 2015. *Anais eletrônicos...* Maceió: Sociedade Brasileira de Computação, 2015. p. 93-101.

LEUTENEGGER, S.; EDGINGTON, J. A Games First Approach to Teaching Introductory Programming. *ACM Inroads*, Covington, KY, USA, v. 39, n. 1, p. 115-118, 2007.

MOHOROVIČIĆ, S.; STRČIĆ, V. An overview of computer programming teaching methods. In: CENTRAL EUROPEAN CONFERENCE ON INFORMATION AND INTELLIGENT SYSTEMS, 22, 2011, Varazdin. *Proceedings...* Varazdin, Croatia: University of Zagreb, 2011. p. 47-52. Disponível em: <<http://www.ceciis.foi.hr/app/index.php/ceciis/2011/paper/view/431>>. Acesso em: 2 nov. 2015.

MURATET, M. et al. Towards a serious game to help students learn computer programming. *International Journal of Computer Games Technology*, New York, NY, USA, v. 2009, n. 1, p. 1-13, 2009.

OVERMARS, M. Teaching computer science through game design. *Computer*, [S. l.], v. 37, n. 4, p. 81-83, 2004.

NIEMANN, F. A.; BRANDOLI, F. Jean Piaget: um aporte teórico para o construtivismo e suas contribuições para o processo de ensino e aprendizagem da Língua Portuguesa e da Matemática. In: SEMINÁRIO DE PESQUISA EM EDUCAÇÃO DA REGIÃO SUL. *Anais...* Caxias do Sul: Universidade de Caxias do Sul, 2012. p. 1-14.



PROGRAMME FOR INTERNATIONAL STUDENT ASSESSMENT. *Pisa 2015: Pisa results in focus*. 2015. Disponível em: <<http://www.oecd.org/pisa/pisa-2015-results-in-focus.pdf>>. Acesso em: 03 nov. 2018.

RODRIGUEZ, C. et al. Pensamento computacional: transformando ideias em jogos digitais usando o Scratch. In: WORKSHOP DE INFORMÁTICA NA ESCOLA, 21, 2015. *Anais eletrônicos...* Maceió: Sociedade Brasileira de Computação, 2015. p. 62.

SANCHIS, I. P.; MAHFOUD, M. Construtivismo: desdobramentos teóricos e no campo da educação. *Revista Eletrônica de Educação*, São Carlos, v. 4, n. 1, p. 18-33, maio 2010.

SCAICO, P. D. et al. Ensino de programação no ensino médio: uma abordagem orientada ao design com a linguagem Scratch. *Revista Brasileira de Informática na Educação*, [S. l.], v. 21, n. 2, p. 92-103, 2013.

SCHWARTZ, G. *Os videogames e a morte dos professores*. 2013. Pesquisa sobre o uso das tecnologias de informação e comunicação nas escolas brasileiras: TIC Educação 2013 [livro eletrônico]. São Paulo: Comitê Gestor da Internet no Brasil, 2014.

SILVA, T. R. et al. Ensino-aprendizagem de programação: uma revisão sistemática da literatura. *Revista Brasileira de Informática na Educação*, [S. l.], v. 23, n. 1, p. 182-196, 2015.

TELECO. *Estatísticas de Celulares no Brasil*. 2018. Disponível em: <<http://www.teleco.com.br/ncel.asp>>. Acesso em: 27 set. 2018.

WANGENHEIM, C. G. V.; NUNES, V. R.; SANTOS, G. D. D. Ensino de computação com Scratch no ensino fundamental: um estudo de caso. *Revista Brasileira de Informática na Educação*, [S. l.], v. 22, n. 3, p. 115, 2014.

WING, J. M. Computational thinking. *Communications of the ACM*, New York, v. 49, n. 3, p. 33-35, 2006.