

Explorando o pensamento computacional no ensino médio: do *design* à avaliação de jogos digitais

Rozelma Soares de França¹, Patrícia Tedesco¹

¹Centro de Informática – Universidade Federal de Pernambuco (UFPE)
Recife – PE – Brasil

{rsf2, pcart}@cin.ufpe.br

Abstract. *The teaching of computational thinking in basic education is urgently needed. However, learning computational concepts presents several challenges. In this light, this work presents a study aimed at analyzing the impact of peer assessment on the learning of computational thinking and the quality of games produced by students in contrast to a traditional approach to learning assessment, in the context of a game programming course in high school. The results suggest that the adoption of the proposed strategy is beneficial to learning and may enhance the playability of the games produced by students.*

Resumo. *O ensino do pensamento computacional na educação básica é uma necessidade premente. Contudo, a aprendizagem de conceitos computacionais possui diversos desafios. Neste âmbito, este trabalho apresenta um estudo que visou analisar o impacto da avaliação por pares na aprendizagem do pensamento computacional e na qualidade de jogos produzidos pelos educandos em contraste com uma abordagem tradicional de avaliação da aprendizagem, no contexto de um curso de programação de jogos digitais no ensino médio. Os resultados obtidos sugerem que a adoção da estratégia proposta é benéfica ao aprendizado e pode contribuir com a jogabilidade de games produzidos pelos estudantes.*

1. Introdução

Na atualidade é requerido dos estudantes desenvolver habilidades que lhes permitam conviver e prosperar no mundo tecnologicamente rico em que vivemos. Nesse contexto, além do conhecimento instrumental sobre editores de texto e planilha, por exemplo, faz-se necessário também uma compreensão sobre os conceitos fundamentais da Ciência da Computação.

Nesse cenário, é importante diferenciar Informática de Computação, embora em alguns países as duas palavras tenham o mesmo significado. Segundo o *Computing at School Working Group* (2011), a Informática trata da aplicação intencional de sistemas computacionais para resolver problemas do mundo real, incluindo questões como a instalação e uso de *hardware* e *software*. Por outro lado, a Computação está particularmente, mas não exclusivamente, relacionada ao estudo, projeto e implementação de sistemas computacionais e aos princípios subjacentes a estes projetos.

Portanto, ao falar-se em Computação enquanto saber necessário na educação básica deve-se levar em consideração que ela engloba princípios fundamentais, como a teoria da computação, e incorpora técnicas e métodos, como a abstração e raciocínio lógico, que podem ser aplicados na solução de problemas e no avanço do conhecimento (*ibid.*).

A habilidade em questão tem sido chamada de *pensamento computacional* (WING, 2006). Ela é a base da Ciência da Computação, tem uma ampla aplicação em múltiplas disciplinas e deve ser explorada desde a formação básica habilitando futuros sociólogos, músicos, físicos, economistas, dentre outros, a resolver problemas de suas respectivas áreas de atuação.

O pensamento computacional pode ser definido como um processo de resolução de problemas que inclui, mas não se limita, às seguintes características (ISTE; CSTA; NSF, 2011): *i)* formular problemas de modo que seja possível usar o computador e outras ferramentas para ajudar a resolvê-los; *ii)* organizar e analisar dados de forma lógica; *iii)* representar dados através de abstrações, tais como modelos e simulações; *iv)* automatizar soluções através do pensamento algorítmico; *v)* identificar, analisar e implementar as soluções possíveis com o objetivo de conseguir a combinação mais eficiente e eficaz de etapas e recursos; e *vi)* generalizar e transferir esse processo de resolução de problemas para uma grande variedade de problemas.

Como forma de ilustrar que a aplicação do pensamento computacional pode ser relevante para diferentes áreas do conhecimento, Nunes (2011) descreve uma situação passível de ocorrer com advogados. O autor cita que tais profissionais podem ler textos e, usando o pensamento computacional, extrair deles fatos e regras, e assim chegar a conclusões que balizem um parecer irrefutável. Outros procedimentos, como organizar eleições, também podem ser expressos de forma algorítmica.

Por outro lado, a aprendizagem de conteúdos fundamentais da Computação tem apresentado dificuldades de natureza diversa, como a capacidade de abstrair conceitos na resolução de problemas ou a habilidade de gerenciar o conhecimento pelo próprio estudante. Isto tem despertado o interesse de pesquisadores e educadores por métodos e ferramentas que auxiliem os iniciantes no aprendizado de conceitos computacionais. Nesse contexto, os jogos têm sido utilizados em disciplinas de Computação, a exemplo de Introdução à Programação, como forma de motivar e engajar os aprendizes durante a aprendizagem, tendo impacto positivo sobre sua formação (WERNER *et al.*, 2014). Contudo, quando além do uso há o desenvolvimento de jogos pelos estudantes outro aspecto deve ser considerado: a qualidade desses artefatos, uma vez que repercutirá na interação entre os usuários e os jogos produzidos.

Ademais, o envolvimento ativo dos estudantes em atividades de reflexão e análise crítica da própria aprendizagem por meio da autoavaliação e avaliação por pares tem demonstrado ser uma abordagem eficaz. Isto é particularmente verdade em comparação à adoção da avaliação com caráter somativo que certifica o conhecimento dos educandos ao final de um curso. No contexto da Computação, evidências empíricas apontam que o processo de avaliação por pares melhora significativamente os resultados de aprendizagem dos estudantes em diversas áreas incluindo habilidades de programação, aprendizagem colaborativa, a conformidade com os padrões de codificação, gestão de tempo, e capacidade de dar e aceitar críticas (WANG *et al.*, 2012).

Todavia, ao se observar a literatura sobre pensamento computacional no ensino médio não se constata a realização de estudos que versem sobre a promoção dessa habilidade por meio da construção de jogos digitais juntamente com a avaliação da aprendizagem centrada no aprendiz. Desse modo, esta pesquisa vem preencher essa lacuna, analisando a influência dessa abordagem na formação dos estudantes em contraste com uma abordagem tradicional de avaliação da aprendizagem.

O restante do artigo está organizado como segue: a Seção 2 traz estudos que tratam do ensino do pensamento computacional na educação básica; na Seção 3 a abordagem de avaliação da aprendizagem proposta nesta pesquisa é apresentada e um estudo experimental é delineado; na Seção 4 são reportados os resultados obtidos com experimento realizado; e, por fim, na Seção 5 são feitas algumas considerações acerca deste trabalho, incluindo direcionamentos para investigações futuras.

2. Evidências do pensamento computacional na escola

Percebendo a importância em desenvolver desde cedo habilidades como o pensamento computacional, os Estados Unidos e países da Europa têm implantado um currículo mínimo de Computação em suas escolas (CSTA, 2005). No Brasil é possível observar iniciativas em diversos estados como Rio de Janeiro (DE SOUZA *et al.*, 2014), Minas Gerais (CARVALHO *et al.*, 2013), Paraíba (SCAICO *et al.*, 2013), Rio Grande do Sul (ANDRADE *et al.*, 2013), Amazonas (VIEIRA *et al.*, 2013), Pernambuco (FRANÇA *et al.*, 2012) e Bahia (SOUSA *et al.*, 2010).

As abordagens de ensino do pensamento computacional empregadas na educação básica são diversas. Quando se tem como objetivo promover essa habilidade por meio da lógica de programação é recorrente o uso de ambientes visuais de programação no desenvolvimento de projetos. Nestes, a programação é feita arrastando-se blocos de comandos que devem ser encaixados uns nos outros. Os comandos assemelham-se a peças de quebra-cabeça e quando combinados formam programas sintaticamente corretos, tendo o estudante que focar apenas na lógica de funcionamento do seu projeto.

Nesse contexto pode-se citar os ambientes Scratch (www.scratch.mit.edu/), App Inventor (www.appinventor.mit.edu) e AgentSheets (www.agentsheets.com/) utilizados nas pesquisas de Scaico *et al.* (2013), Gomes e Melo (2013) e de Souza *et al.* (2014), respectivamente. Com o uso do Scratch foi possível ensinar as principais estruturas de uma linguagem de programação e praticar técnicas utilizadas na construção de algoritmos, como é o caso do uso da abstração, depuração de erros, testes e melhoria de algoritmos simples. Já o App Inventor possibilitou a criação aplicativos para dispositivos Android por estudantes do ensino médio. Com o AgentSheets os educandos expressaram seus interesses e capacidades de comunicação por meio da programação de jogos e simulações.

Os princípios da Computação também têm sido ensinados aos estudantes sem depender do uso de uma tecnologia específica. Nesse contexto, destaca-se o trabalho de Bell *et al.* (2011) que propuseram um conjunto de atividades lúdicas envolvendo fundamentos da Ciência da Computação, publicadas no livro *Computer Science Unplugged*. Tais atividades estimulam o pensamento computacional, sem o uso do computador, e têm despertado o interesse de pesquisadores e professores, bem como empregadas em diversos países como Estados Unidos, França e Japão (SOUSA *et al.*, 2010). No Brasil também há relatos de adoção dessa metodologia de ensino apontados por França *et al.* (2012) e Vieira *et al.* (2013) e seus resultados sugerem que ela pode engajar os estudantes nas atividades bem como promover a aprendizagem de conceitos computacionais.

Considerando-se que a capacidade de resolver problemas com a aplicação do pensamento computacional pode ser aplicada em várias áreas, pesquisas também têm trabalhado tal habilidade articulada com outras disciplinas do currículo. No trabalho de Barcelos e Silveira (2013) os autores investigaram de que forma as competências relacionadas

à Matemática são mobilizadas por estudantes do ensino médio no processo de desenvolvimento do pensamento computacional através da construção de jogos digitais. Nessa direção, Allan *et al.* (2010) apresentam projetos que integram a Computação em disciplinas já existentes no currículo do ensino médio, tais como Física, Química e Matemática. Em um dos projetos envolvendo Física, por exemplo, os autores revelam que o uso de visualização e o favorecimento a atividades criativas tiveram papel fundamental em manter o empenho os estudantes, uma vez que eles tinham diferentes interesses de resolução de problemas e habilidades de abstração distintas.

3. Produção e avaliação de jogos digitais no ensino médio

O estudo apresentado neste artigo tem como objetivo analisar o processo de avaliação por pares com o propósito de caracterizá-lo com respeito à contribuição na aprendizagem do pensamento computacional como também na qualidade de jogos digitais produzidos por estudantes do ensino médio. Tal abordagem será chamada de *alternativa* e contrastada com o método *tradicional* de avaliação da aprendizagem durante um curso de desenvolvimento de games. Para nortear esta investigação, as seguintes questões e métricas foram definidas:

- **Q1:** Houve melhoria na aprendizagem do pensamento computacional?
 - ⊕ *Métrica:* desempenho obtido pelos estudantes no conteúdo Estruturas Condicionais.
- **Q2:** Houve melhoria na qualidade dos jogos digitais produzidos pelos estudantes?
 - ⊕ *Métrica:* resultado médio do índice de jogabilidade dos games produzidos.

Duas hipóteses nulas e alternativas foram definidas para as duas questões de pesquisa norteadoras deste estudo experimental.

Para a Q1:

- *Hipótese Nula ($H0_1$):* O desempenho médio dos aprendizes que estudaram com a abordagem alternativa é igual ao obtido por aqueles que estudaram com a abordagem tradicional.
- *Hipótese Alternativa ($H1_1$):* O desempenho médio dos aprendizes que estudaram com a abordagem alternativa é melhor que o obtido por aqueles que estudaram com a abordagem tradicional.

Para a Q2:

- *Hipótese Nula ($H0_2$):* O resultado médio do índice de jogabilidade dos jogos produzidos pelos aprendizes que estudaram com a abordagem alternativa é igual ao obtido por aqueles que estudaram com a abordagem tradicional.
- *Hipótese Alternativa ($H1_2$):* O resultado médio do índice de jogabilidade dos games produzidos pelos aprendizes que estudaram com a abordagem alternativa é melhor que o obtido por aqueles que estudaram com a abordagem tradicional.

3.1. Contexto e participantes

O estudo foi realizado em curso de desenvolvimento de jogos digitais oferecido pelo Espaço Ciência de Pernambuco em janeiro de 2015. O curso possuía duas turmas, cada uma com 6h de aula por semana; era direcionado a adolescentes, com conhecimentos em informática básica; e objetivava introduzir conceitos básicos de programação para o público em questão. O referido curso foi escolhido por ser constituído de conteúdos definidos em currículos para o ensino do pensamento computacional na escola e ser formado por estudantes da educação básica.

A população do estudo foi composta pelos vinte e dois estudantes do curso de games. Não foi possível a atribuição aleatória dos participantes aos grupos controle e experimental. Desse modo, um quasi-experimento foi realizado, ao invés de um experimento, caso frequente de ocorrer no contexto educacional devido a questões éticas e operacionais envolvidas na intervenção didática (SHADISH *et al.*, 2002).

As duas turmas que formavam os grupos controle e experimental possuíam professores distintos. No entanto, de acordo com o levantamento de dados com os educadores, o planejamento pedagógico era definido conjuntamente. As aulas ministradas eram expositivas com a realização de exercícios práticos. Em tais exercícios, havia o desenvolvimento de jogos digitais com o uso do Stencyl (www.stencyl.com), um *software* que possibilita a criação de jogos para dispositivos móveis, web e desktop através do encaixe de blocos de comando, tendo o usuário que preocupar-se apenas com a lógica de funcionamento do projeto.

Antes da realização do quasi-experimento, componentes do Stencyl (atores, cenários, fontes e outros) haviam sido apresentados às turmas. Além disso, conceitos fundamentais de programação haviam sido ensinados como operadores lógicos, eventos e variáveis. Os estudantes eram avaliados ao fim dos exercícios propostos e realizados no curso, sendo cada professor responsável por corrigir as atividades de sua turma.

3.2. Execução

O quasi-experimento ocorreu no período de 21 a 26 de janeiro de 2015. Inicialmente, aos estudantes de ambas as turmas, foi apresentado o objetivo da pesquisa, como também os riscos e benefícios de sua participação no estudo, seguida da assinatura do Termo de Consentimento Livre e Esclarecido por aqueles que demonstraram interesse em participar do experimento. Depois, os estudantes preencheram um questionário de perfil seguido da realização do pré-teste. Este visou identificar o nível de conhecimento dos participantes sobre o conteúdo Estruturas Condicionais e foi constituído de três questões que exploravam as habilidades de depuração e criação de código por meio dos blocos de comando do Stencyl. A primeira questão do pré-teste explorava a capacidade dos estudantes de detectar e corrigir erros; a segunda requeria a análise de um jogo e a implementação das funcionalidades que estavam faltando; e a terceira solicitava a criação de um game. Em todos os casos, era necessário que os estudantes mobilizassem seus conhecimentos sobre estruturas condicionais e utilizassem os blocos de comandos do Stencyl para responder às questões.

Na etapa de intervenção, ocorrida no encontro seguinte, os professores de ambas as turmas apresentaram Estruturas Condicionais aos estudantes seguido da proposição de exercícios, em formato de jogos, que exploravam o conteúdo abordado. Ao grupo controle, cabia responder os exercícios propostos no Stencyl e apresentar a solução para avaliação pelo professor. Ao grupo experimental, as atividades também foram realizadas em formato de jogos, no Stencyl, contudo a avaliação da produção foi realizada pelos estudantes. Assim, a cada um deles foi atribuído de dois a três jogos desenvolvidos por seus colegas que deveriam ser analisados de acordo com critérios definidos previamente.

Considerando-se a importância da qualidade da interação do usuário com jogos digitais, heurísticas de jogabilidade foram apresentadas para ambos os grupos, incentivando que os estudantes atentassem para tais aspectos no momento da construção de seus jogos. Para o grupo controle, as heurísticas propostas por Barcelos *et al.* (2011), exibidas no Quadro 1, foram explanadas oralmente e entregues em formato impresso a cada um dos participantes. Para o grupo experimental, as heurísticas além de discutidas e entregues à turma, foram também

utilizadas para avaliação de jogos produzidos pelos estudantes. Desse modo, a avaliação por pares contemplou critérios para análise de aspectos de programação, especificamente Estruturas Condicionais, bem como àqueles relacionados à jogabilidade.

No encontro seguinte, os estudantes de ambos os grupos responderam a um pós-teste que foi estruturado nos moldes do pré-teste, visando identificar o desempenho dos participantes no conteúdo Estruturas Condicionais. Na primeira questão do pós-teste era requisitado detectar e corrigir erros em um jogo; na segunda era necessário analisar outro game e implementar as funcionalidades que estavam faltando; e a terceira questão requeria que os participantes usassem a criatividade para implementar um jogo. Novamente, para solucionar os problemas propostos os estudantes deveriam mobilizar seus conhecimentos sobre Estruturas Condicionais e codificar os jogos, arrastando blocos de comando no Stencyl.

Quadro 1. Heurísticas de Jogabilidade

| IDENTIFICADOR | HEURÍSTICA |
|---------------|--|
| H1 | Os controles devem ser claros, customizáveis e fisicamente confortáveis; suas respectivas ações de resposta devem ser imediatas. |
| H2 | O jogador deve poder customizar o áudio e o vídeo do jogo de acordo com suas necessidades. |
| H3 | O jogador deve conseguir obter com facilidade informações sobre seu status e pontuação. |
| H4 | O jogo deve possibilitar que o jogador desenvolva habilidades que serão necessárias futuramente. |
| H5 | O jogador deve encontrar um tutorial claro de treinamento e familiarização com o jogo. |
| H6 | Todas as representações visuais devem ser de fácil compreensão pelo jogador. |
| H7 | O jogador deve ser capaz de salvar o estado atual para retomar o jogo posteriormente. |
| H8 | O layout e os menus devem ser intuitivos e organizados de forma que o jogador possa manter o seu foco na partida. |
| H9 | A história deve ser rica e envolvente criando um laço com o jogador e seu universo. |
| H10 | Os gráficos e a trilha sonora devem despertar o interesse do jogador. |
| H11 | Os atores digitais e o mundo do jogo devem parecer realistas e consistentes. |
| H12 | O objetivo principal do jogo deve ser apresentado ao jogador desde o início. |
| H13 | O jogo deve propor objetivos secundários e menores, paralelos ao objetivo principal. |
| H14 | O jogo deve possuir vários desafios e permitir diferentes estratégias. |
| H15 | O ritmo do jogo deve levar em consideração a fadiga e a manutenção dos níveis de atenção. |
| H16 | O desafio do jogo pode ser ajustado de acordo com a habilidade do jogador. |
| H17 | O jogador deve ser recompensado pelas suas conquistas de forma clara e imediata. |
| H18 | A inteligência artificial deve representar desafios e surpresas inesperadas para o jogador. |
| H19 | O jogo deve fornecer dicas, mas não muitas. |

Fonte: Barcelos *et al.* (2011)

4. Resultados

Nesta seção os resultados serão analisados e interpretados oferecendo as conclusões sobre a possibilidade de rejeição das hipóteses nulas.

4.1. Perfil dos participantes

A identificação do perfil dos participantes foi realizada por meio de um questionário, respondido pelos vinte e dois estudantes do curso de games. O grupo experimental era formado pela turma da manhã, composta por nove educandos dos quais apenas um era do sexo feminino. Eles tinham idade variando entre 14 e 17 anos e cinco deles informaram já ter cursado alguma disciplina de programação antes de ingressar no curso de games. A turma da tarde constituiu o grupo controle e possuía treze estudantes, sendo três deles do sexo feminino. Eles tinham idade variando entre 11 e 32 anos, porém a maioria estava da faixa de 15 a 18 anos. No que se refere a conhecimentos em programação, cinco deles afirmaram ter cursado anteriormente alguma disciplina da área.

Sobre a experiência com jogos digitais, a maioria de ambos os grupos informou dedicar mais de 12h por semana ao uso de games. No que diz respeito ao gênero dos jogos preferido

dos participantes, identificou-se que no grupo experimental os três gêneros preferidos são: estratégia, ação/aventura e RPG. No grupo controle, os gêneros de jogos preferidos são: ação/aventura, RPG e simulação, nessa ordem.

4.2. Frequência dos participantes no estudo

O quasi-experimento ocorreu no período regular das aulas do curso de games. Assim, caso um estudante faltasse à aula, sua ausência também era registrada nesta pesquisa. Todos os educandos de ambas as turmas participaram do experimento em algum momento de sua execução. Todavia, nem todos fizeram o pré e o pós-teste.

Tendo em vista que os testes foram utilizados como instrumentos de coleta para identificação de possíveis variações na aprendizagem do pensamento computacional e também na qualidade dos jogos produzidos, esta análise levará em consideração apenas os dados dos estudantes que fizeram o pré e o pós-teste. No grupo experimental, isto corresponde a uma amostra de seis participantes. Esta amostra é representativa ao se considerar a frequência dos estudantes neste estudo: 7,67 participantes por encontro (min: 6; max: 9). No grupo controle, a amostra de participantes do pré e pós-teste é igual a sete e também é considerada representativa tendo em vista a frequência dos estudantes nas atividades: média de 9 por encontro (min: 7, max: 13)¹.

4.3. Pensamento computacional

As questões sobre estruturas condicionais que compuseram o pré e pós-teste foram analisadas como forma de avaliar a aprendizagem dos estudantes no experimento. Para as questões que requeriam a identificação e correção de erros foi verificado se os códigos criados pelos estudantes atendiam às condições exigidas, bem como se eles utilizam apenas os comandos necessários. Os exercícios que solicitavam que fossem implementadas funcionalidades faltantes também foram avaliados sob o ponto de vista do atendimento às condições exigidas e à qualidade dos códigos produzidos. Já as questões que demandavam a criação de jogos foram avaliadas do ponto de vista da aprendizagem e da jogabilidade dos games produzidos. Em relação à aprendizagem do pensamento computacional, verificou-se se os jogos atendiam aos requisitos mínimos solicitados, como também se os códigos produzidos apresentavam boa qualidade. Os aspectos inerentes à jogabilidade serão apresentados em outra seção, adiante.

Três questões integraram cada teste e a pontuação máxima considerada para cada uma delas foi dez. Desse modo, a nota máxima que os estudantes poderiam obter em cada teste era trinta. A Tabela 1 apresenta as médias obtidas pelos participantes deste estudo em sua avaliação da aprendizagem. Como pode ser observado, o grupo experimental apresentava inicialmente uma média superior ao grupo controle. Após as intervenções didáticas em que o conteúdo Estruturas Condicionais foi apresentado, ambos os grupos demonstraram acréscimo de seu conhecimento sobre o assunto. Contudo, o grupo experimental demonstrou um ganho mais expressivo, se comparado ao controle.

Objetivando-se averiguar a significância estatística dos resultados apresentados, recorreu-se ao teste *t* de Student para as hipóteses H_{01} e H_{11} . A hipótese nula afirma que os resultados obtidos pelo grupo experimental são iguais aos do grupo controle no que diz respeito

¹ Considerando-se para os grupos experimental e controle, respectivamente, tamanho da população igual a 9 e 13, com média de 7,67 e 9, desvio médio igual a 1,53 e 2,65 e adotando-se o nível de confiança de 95% e erro máximo da estimativa equivale a 2, a amostra corrigida pela população é igual a 3 e 5, nessa ordem.

à aprendizagem do pensamento computacional. Todavia, a hipótese alternativa declara que tais resultados são melhores.

Tabela 1. Resultado da aprendizagem do pensamento computacional

| ESCALA | PRÉ-TESTE | | PÓS-TESTE | |
|--------------------|-----------|---------------|-----------|---------------|
| | Média | Desvio Padrão | Média | Desvio Padrão |
| Grupo Experimental | 13,75 | 7,71 | 24,17 | 3,42 |
| Grupo Controle | 3,93 | 6,43 | 7,86 | 10,04 |

Tendo em vista as médias obtidas pelos estudantes no pós-teste utilizou-se o teste *t* de Student sobre as duas amostras presumindo variâncias diferentes e nível de significância de 5%. O valor-P resultante é igual a 0,0038 ou 0,38%. Como este valor é inferior a 5%, a hipótese nula é rejeitada e há aceitação da hipótese alternativa. Isto implica dizer que existe relação entre a abordagem alternativa e a melhora da aprendizagem do pensamento computacional.

É importante ressaltar que estudantes dos grupos controle e experimental declararam ter cursado alguma disciplina introdutória de programação antes de ingressar no curso de games e isto pode ter impactado nos resultados obtidos. Nesse contexto, a pesquisa de Byrne e Lyons (2001) sugere que experiência anterior em programação pode favorecer o aprendizado na disciplina. Nesta pesquisa, a partir do questionário de perfil foi identificado que estudantes já haviam cursado alguma disciplina na área. Contudo, cinco de cada grupo afirmaram isto e, mesmo assim, o grupo experimental apresentou melhor desempenho, o que sugere que a abordagem alternativa de avaliação da aprendizagem proposta tem relação com esse resultado e não apenas o conhecimento prévio em programação.

4.4. Jogabilidade dos games produzidos

Para apoiar a análise da jogabilidade, os jogos produzidos pelos estudantes na questão 3 do pré e do pós-teste foram avaliados a partir das heurísticas de jogabilidade de Barcelos *et al.* (2011). Cada heurística foi pontuada por meio da escala *Likert*, sendo: 1) Discordo Totalmente, 2) Discordo, 3) Neutro, 4) Concordo e 5) Concordo Totalmente. A partir disso, o índice de jogabilidade foi calculado por meio do somatório dos pontos atribuídos a cada heurística. Tendo em vista que há dezenove heurísticas e a pontuação máxima que cada uma pode obter é cinco, o escore máximo que cada jogo pode ter é noventa e cinco. A Tabela 2 apresenta o resultado médio do índice de jogabilidade dos games produzidos no pré e pós-teste pelos grupos controle e experimental.

Tabela 2. Resultado da jogabilidade dos games produzidos

| ESCALA | PRÉ-TESTE | | PÓS-TESTE | |
|--------------------|-----------|---------------|-----------|---------------|
| | Média | Desvio Padrão | Média | Desvio Padrão |
| Grupo Experimental | 48 | 2,65 | 58,60 | 16,83 |
| Grupo Controle | 34 | 6 | 42,17 | 9,45 |

Como pode ser observado, o grupo experimental apresentou no pré-teste um resultado médio superior ao grupo controle. Após serem apresentados às heurísticas de jogabilidade, na etapa de intervenção, ambos os grupos tiveram seus escores ampliados havendo, contudo, um maior ganho no grupo experimental, apesar de ter a maior dispersão dos valores apontada pelo desvio padrão.

Visando-se verificar se a diferença observada nos resultados é estatisticamente significativa, o teste t de Student foi novamente utilizado para as hipóteses H_0 e H_1 . A hipótese nula afirma que os resultados obtidos no grupo experimental são iguais aos do grupo controle no que se refere à jogabilidade dos jogos produzidos. Por outro lado, a hipótese alternativa afirma que tais resultados são diferentes. Desse modo, levando-se em consideração a média de cada estudante obtida na análise da jogabilidade dos games produzidos na terceira questão do pós-teste, aplicou-se o teste t sobre as duas amostras presumindo variâncias diferentes e nível de significância de 5%. O valor-P resultante é igual a 0,1001 ou 10,01%. Este valor é superior ao nível de significância adotado, então hipótese nula não foi rejeitada.

Apesar disso, pelos resultados obtidos no pré e pós-teste é possível perceber que o grupo experimental teve maior ganho no que se refere à qualidade de seus jogos, se comparado ao grupo controle. Isto pode ter ocorrido devido à forma com que as heurísticas de jogabilidade foram apresentadas aos grupos na etapa de intervenção. Enquanto que o grupo controle conheceu as heurísticas por meio de uma apresentação oral e material impresso, o grupo experimental, além de ter essas atividades, utilizou as heurísticas para avaliar jogos criados por seus colegas. Esta atividade pode ter oportunizado aos estudantes refletir sobre os jogos produzidos, aumentando a consciência da qualidade do próprio trabalho a partir da avaliação dos jogos de seus pares. Como consequência, o grupo experimental pôde incorporar em seus jogos, com mais frequência e qualidade, as heurísticas ora apresentadas e utilizadas como critério de avaliação. Assim, tal resultado sugere que a longo prazo a avaliação por pares pode contribuir com a qualidade dos jogos produzidos por estudantes durante a aprendizagem do pensamento computacional.

5. Considerações finais

A lista de habilidades e conhecimentos necessários para o pleno exercício da cidadania no século XXI é extensa, incluindo o pensamento computacional. Ele é necessário por permitir que os estudantes melhor conceituem, analisem e resolvam problemas complexos e pode ser aplicado às diversas áreas do saber. Por outro lado, a educação em Computação é considerada um desafio, especialmente o ensino introdutório de programação.

Assim, neste trabalho foi apresentado um estudo que analisou o impacto da avaliação por pares na formação em Computação de estudantes do ensino médio. Supôs-se que a partir do contato com essa abordagem alternativa de avaliação da aprendizagem os educandos melhorariam a aprendizagem do pensamento computacional, como também a qualidade de jogos produzidos durante sua formação. Entretanto, apenas uma hipótese foi confirmada com os dados obtidos neste estudo.

Do ponto de vista da aprendizagem do pensamento computacional foi possível constatar o impacto positivo da avaliação por pares na formação dos estudantes. Por outro lado, a hipótese que tratava da melhoria da qualidade dos jogos produzidos não foi confirmada. Apesar disso, os resultados demonstram melhora nos aspectos inerentes à jogabilidade dos games criados após ambos os grupos terem tido contato com heurísticas da área, sendo o grupo que experimentou a avaliação por pares o que obteve maior rendimento. Tais resultados fornecem *insights* para realização de futuras pesquisas que poderão considerar um maior número de participantes e iterações, como também uma diversificação de conteúdos de Computação e variação de ferramentas de suporte ao desenvolvimento de jogos digitais por estudantes do ensino médio.

6. Agradecimentos

Rozelma Soares de França agradece à CAPES pela concessão de bolsa de mestrado no período de realização desta pesquisa.

Referências

- Allan, V. *et al.* (2010). “Computational thinking in high school courses”. In: *Proceedings of the 41st SIGCSE*, p. 390-391.
- Andrade, D. *et al.* (2013). “Proposta de Atividades para o Desenvolvimento do Pensamento Computacional no Ensino Fundamental”. In: *Anais do XVI WIE*.
- Barcelos, T. S.; Silveira, I. F. (2013). “Relações entre o Pensamento Computacional e a Matemática através da construção de Jogos Digitais”. In: *Anais do XII SBGames*.
- Barcelos, T. S. *et al.* (2011). “Análise comparativa de heurísticas para avaliação de jogos digitais”. In: *Proceedings of the 10th IHC and the 5th CLIHC*, p. 187-196.
- Bell, T.; Witten, I, H.; Fellows, M. (2011). *Computer Science Unplugged: Ensinando Ciência da Computação sem o uso do computador*. Tradução coord. por Luciano Porto Barreto.
- Byrne, P.; Lyons, G. (2001). “The effect of student attributes on success in programming”. In: *Proceedings of the SIGCSE*, p. 49-52.
- Carvalho, M. L. B.; Chaimowicz, L.; Moro, M. M. (2013) “Pensamento Computacional no Ensino Médio Mineiro”. In: *Anais do XXI WEI*, p. 640-649.
- Computing at School Working Group. *Computing: A curriculum for schools*. Dezembro, 2011.
- CSTA. (2005). *The New Educational Imperative: Improving High School Computer Science Education*. Final Report of the CSTA. Curriculum Improvement Task Force. ACM.
- De Souza, C. S. *et al.* (2014). “Cultural appropriation of computational thinking acquisition research: seeding fields of diversity”. In: *Proceedings of the 19th ITiCSE*, p. 117-122.
- França, R. S; Silva, W. C.; Amaral, H. J. C. (2012). “Ensino de ciência da computação na educação básica: experiências, desafios e possibilidades”. In: *Anais do XX WEI*.
- Gomes, T. C. S.; Melo, J. C. B. (2013). “O Pensamento Computacional no Ensino Médio: Uma Abordagem Blended Learning”. In: *Anais do XXI WEI*, p. 640-649.
- ISTE; CSTA; NSF. (2011). *Computational thinking: leadership toolkit*. First Edition.
- Nunes, D. J. (2011). “Ciência da Computação na Educação Básica”. *Jornal da Ciência*. 09 de setembro.
- Scaico, P. D. *et al.* (2013). “Ensino de Programação no Ensino Médio: Uma Abordagem Orientada ao Design com a linguagem Scratch”. *Revista Brasileira de Informática na Educação*, v. 21, n. 02, p. 92.
- Shadish, W. R.; Cook, T. D.; Campbell, D. T. (2002). *Experimental and quasi-experimental designs for generalized causal inference*. Boston: Houghton Mifflin Company.
- Sousa, R. V. *et al.* (2010). “Ensinando e aprendendo conceitos sobre a ciência da computação sem o uso do computador: Computação Unplugged!”. *Práticas em Informática na Educação: Minicursos do CBIE*, v.1, n.1.
- Vieira, A.; Passos, O. ; Barreto, R. (2013). “Um Relato de Experiência do Uso da Técnica Computação Desplugada”. In: *Anais do XXI WEI*, p. 670-679.
- Wang, Y. *et al.* (2012). “Assessment of programming language learning based on peer code review model: Implementation and experience report”. *Computers & Education*, v. 59, n. 2, p. 412-422.
- Werner, L., Denner, J., Campe, S. (2014). “Children Programming Games: A Strategy for Measuring Computational Learning”. *ACM Transactions on Computing Education*, v.14, n., p. 24.
- Wing, J. M. (2006). “Computational thinking”. *Communications of the ACM*, v. 49, n. 3, p. 33-35.