

Uso de Jogos em Cursos Introdutórios de Programação no Ensino Superior na Área de Computação: Uma Revisão Sistemática

Alexandre Scaico¹, Pasqueline Dantas Scaico¹

¹Departamento de Ciências Exatas, Centro de Ciências Aplicadas e Educação,
Universidade Federal da Paraíba – Campus IV
Rua da Mangueira, s/n, Rio Tinto, PB, Cep: 58000-000

{alexandre,pasqueline}@dcx.ufpb.br

***Abstract.** Despite the increasing demand for Computer Science professionals in recent years, the number of graduates in this area has been decreasing. Introductory programming courses have been related as a major factor to this issue especially because of high failure rates lead to student demotivation and dropouts. In order to increase interest and improve the learning process some strategies have been tried, including games and teaching methods based on it. This study concerns to a systematic literature review on the use of digital games in teaching introductory programming courses related to Computer Science.*

***Resumo.** Apesar do aumento da demanda por profissionais da área de Computação a quantidade de graduados nesta área tem diminuído. Um dos aspectos que vem contribuindo para isto é o alto índice de reprovação em disciplinas introdutórias de programação, que ocasiona desmotivação e consequentemente abandono do curso. Como forma de aumentar o interesse dos alunos, e com isso melhorar o aprendizado e reduzir a evasão, algumas estratégias vêm sendo testadas. Uma dessas estratégia é o uso de jogos como apoio ao ensino de programação. Este trabalho objetivou realizar uma revisão sistemática da literatura acerca do uso de jogos digitais no ensino introdutório de programação em cursos ligados a área de Computação.*

1. Introdução

A Computação é uma área com bastante potencial para gerar novos campos de atuação e, consequentemente, mais postos de trabalho. A ACM (Association for Computing Machinery) prevê que 120.000 empregos sejam abertos a cada ano até 2020 apenas nos Estados Unidos. Contudo, a escassez de graduados em Ciência da Computação aponta um cenário que pode ser um dos problemas mais graves enfrentados pela indústria de tecnologia e pela economia dos países [“Microsoft Corporate Citizenship: U.S. Talent Strategy” 2007].

No mundo todo vem sendo observado o aumento nos índices de reprovação nas disciplinas introdutórias de programação e no número de estudantes que abandonam seus cursos de Computação ainda no primeiro ano de graduação [De-La-Fuente-Valentín et al. 2013] [Kinnunen 2009]. No Brasil, a média de desistência nessa área é uma das maiores do mundo, estando em torno de 30% [“Matemática e ciências da

computação têm alta taxa de desistência” 2009]. O desinteresse dos estudantes pelas ciências tem feito muitos pesquisadores da Computação investigarem o uso de novos recursos que facilitem a aprendizagem, a exemplo dos trabalhos que exploram o uso de robótica, softwares educativos, sistemas de tutoria e metodologias para o ensino de programação [Major et al. 2012] [Fowler and Cusack 2011] [Mattos and Fábio 2007] [Muratet et al. 2012] [D’Souza et al. 2008] [Kunkle 2010].

Entretanto, conforme relata [Piteira e Haddad 2011], o aumento no desinteresse dos alunos parece estar bastante relacionado aos métodos de ensino, que por vezes ainda se baseiam em aulas expositivas. [Vihavainen et al. 2011] aponta que é comum que uma aula de programação esteja direcionada à leituras, resolução de listas de exercícios e demonstrações de código. [Scaico et al. 2013] destaca, por sua vez, que as dificuldades em aprender programação estão associadas ao uso de estratégias que possam motivar e engajar os estudantes no processo, assim como, ao ajuste de diferentes emoções que estão associadas ao processo de aprender, como é o caso da frustração.

O interesse é um aspecto que influencia diretamente o engajamento dos alunos no processo de aprendizagem em cursos introdutórios de programação e que influencia, conseqüentemente, a redução das taxas de evasão e retenção. Uma das estratégias que vem sendo utilizada para aumentar o interesse dos alunos, e conseqüentemente melhorar o desempenho e reduzir a evasão, é o uso de jogos no apoio ao ensino de programação. Conforme menciona [Salen et al. 2011], os jogos são sistemas poderosos de aprendizagem, especialmente porque podem ser um gatilho para despertar a imaginação e o engajamento que ocorrem quando se está engajado em uma experiência que é relevante. Um jogo é um mecanismo que valoriza, dentre outras coisas, o contato com negociações complexas, a investigação de situações imaginárias e a interpretação, processos que culminam na diversão do jogador, mas que também representam um sistema relevante para que seja possível criar autonomamente significado para as coisas.

Este artigo apresenta o resultado de uma revisão sistemática da literatura que visou identificar e analisar os estudos relacionados ao uso de jogos digitais em disciplinas introdutórias de programação de cursos superiores da área de computação. A ênfase da revisão estudo se voltou para trabalhos que tenham aplicado e validado estudos comparativos entre grupos de alunos em que o uso de jogos foi ou não utilizado visando entender como o processo de ensino/aprendizagem de programação foi influenciado pelo uso de jogos digitais. Sendo assim, trabalhos contendo relatos de experiência baseados na percepção do professor ou que apenas apresentavam propostas não foram considerados nesta revisão.

Esta revisão sistemática faz parte de um estudo mais amplo que tem como objetivo entender os fatores que geram, ou inibem, o interesse no aprendizado de programação de alunos de cursos superiores na área de Computação. Este trabalho está organizado da seguinte maneira: na Seção 2 é detalhado o protocolo da revisão sistemática; na seção 3 a execução da revisão é apresentada; na Seção 4 apresenta-se a análise dos artigos primários selecionados. Por fim, a Seção 5 contém as conclusões dos autores.

2. Protocolo da Revisão Sistemática

Uma das primeiras etapas de uma revisão sistemática trata-se da definição do protocolo da revisão, que descreve como ocorrerá o processo de identificação das publicações. Na

criação do protocolo são definidos os objetivos da revisão, as questões de pesquisa que estão sendo perseguidas, as estratégias de busca das publicações, os critérios de seleção dos trabalhos, o procedimento de seleção dos trabalhos, os parâmetros de análise da qualidade dos trabalhos e como será realizada a extração de dados.

2.1. Objetivos

Este trabalho teve como objetivo principal:

- Identificar os estudos que aplicaram e validaram o uso de jogos digitais no processo de ensino/aprendizagem em cursos superiores na área de Computação.

A partir do objetivo principal, os seguintes objetivos específicos foram definidos:

- Identificação do desenho metodológico dos estudos, bem como, os seus achados de pesquisa acerca do fenômeno central então explorado;
- Sintetização dos trabalhos identificados visando entender como o uso de jogos digitais pode beneficiar o ensino de programação em cursos superiores ligados a área de computação.

2.2. Questão de Pesquisa

A partir do objetivo foi definida a seguinte questão de pesquisa:

- Quais são os estudos, validados a partir da comparação entre grupos experimentais e de controle, focados no uso de jogos digitais como elemento auxiliar no processo de ensino/aprendizagem de introdução à programação em cursos superiores ligados a área de Computação e quais são os seus achados?

2.3. Fontes de busca e idiomas

O processo de busca dos artigos foi realizada a partir de bases de dados digitais, a saber: ACM Digital Library¹, Springer Link², IEEE Explorer³, Science Direct⁴ e CiteSeerX⁵. Para a realização desta revisão sistemática foi definido o idioma inglês (idioma padrão para publicações internacionais).

2.4. String de busca

A partir da questão de pesquisa foram definidas as palavras-chave usadas para montar a *string* de busca a ser utilizada nos motores de pesquisa.

- ("game") AND ("learn" OR "learning" OR "teach" OR "teaching") AND ("introductory programming" OR "CS0" OR "CS1") AND ("programming" OR "programmer" OR "code" OR "coding") AND ("undergraduate" OR "higher education" OR "university" OR "college") AND ("computer" OR "computing")

¹ <http://dl.acm.org>

² <http://link.springer.com>

³ <http://ieeexplore.ieee.org/Xplore/home.jsp>

⁴ <http://www.sciencedirect.com>

⁵ <http://citeseerx.ist.psu.edu/index>

2.5. Critérios de seleção dos trabalhos

Para definir se os trabalhos provenientes do resultado da busca seriam selecionados foram definidos os critérios de inclusão e exclusão, contidos na Tabela 1:

Tabela 1 – Critérios de seleção dos artigos primários

Critérios de inclusão:	Critérios de exclusão:
<ul style="list-style-type: none">- O artigo deve estar em inglês ou português- O artigo deve estar disponível integralmente- O artigo deve ter sido publicado a partir do ano 2000- O artigo deve ter no resumo alguma relação com o objetivo dessa pesquisa- Se encontrados vários artigos relacionados ao mesmo trabalho apenas o mais recente será considerado	<ul style="list-style-type: none">- O artigo não possui uma análise comparativa de sua aplicação entre grupos de controle e experimental- O artigo apresenta apenas uma proposta sem ter sido aplicada a um grupo de alunos- O artigo tem sua condução focada em um grupo que não corresponde a alunos introdutórios de programação no ensino superior ligado a área de Computação

2.6. Procedimento de seleção e validação dos artigos

Para a seleção dos artigos realizou-se primeiramente uma busca automatizada nas bases de dados digitais a partir da *string* de busca, documentando, posteriormente, os resultados obtidos. A partir do resultado da busca automatizada foi realizada uma nova seleção baseada nos critérios de inclusão e exclusão, definindo assim os artigos primários desta revisão sistemática.

Para validar o processo de seleção um dos autores realizou as buscas em todas as bases digitais enquanto o outro conduziu a busca em duas bases digitais escolhidas aleatoriamente para fins de comparação e validação do processo.

2.7. Parâmetros de análise de qualidade dos trabalhos selecionados

Para analisar a qualidade dos trabalhos selecionados foi utilizada uma adaptação das 11 questões presentes em [Dybå and Dingsøy 2008]. A escala de pontuação dessas questões é baseada na resposta ser SIM (1 ponto), TALVEZ (0,5 ponto) ou NÃO (0 ponto). Com esta análise, tem-se uma avaliação da qualidade dos trabalhos para fins de comparação, não de exclusão. As questões utilizadas nessa revisão foram:

1. Os objetivos da pesquisa são definidos claramente?
2. Existe uma descrição clara do contexto no qual a pesquisa foi realizada?
3. A condução da pesquisa foi adequada aos objetivos da mesma?
4. A estratégia de recrutamento foi adequada aos objetivos da pesquisa?
5. Existiu um grupo de controle para comparar os resultados da aplicação da estratégia apresentada?
6. A análise dos dados foi suficientemente rigorosa?
7. Os resultados são apresentados claramente?
8. O estudo tem valor para a pesquisa ou prática?

3. Execução da revisão sistemática

Esta revisão sistemática ocorreu durante o mês de novembro do ano de 2015. Inicialmente foram realizadas as buscas automatizadas nas bases de dados, seguida da análise dos mesmos baseados nos critérios de inclusão e exclusão. A Tabela 2 sintetiza os resultados desta etapa. Vale salientar que foram identificados muitos trabalhos que se propõem a usar jogos no ensino introdutório de programação, mas poucos estavam efetivamente direcionados para disciplinas introdutórias de programação de cursos superiores na área de Ciência da Computação. Dentre estes, apenas uma pequena parcela realizou um estudo comparativo da aplicação de jogos, sendo a maioria baseada apenas na percepção do professor da disciplina ou em propostas de metodologia que ainda não havia sido aplicadas e validadas.

Tabela 2 – Síntese dos resultados da busca de artigos através dos motores de busca

Motor de Busca	ACM Digital Library	Springer Link	IEEEEXplorer	Science Direct	CiteSeerX
Resultado da busca	231	261	21	176	170
Artigos primários	12	2	1	0	2

Uma vez selecionados os artigos, realizou-se a extração dos dados e sua análise de qualidade. Na Tabela 3 são apresentados os trabalhos primários selecionados para esta revisão sistemática.

Tabela 3 – Artigos primários selecionados

Ref	Título	Autores	Motor de Busca	Análise da qualidade
[1]	A case of computer science principles with traditional text-based programming languages	Kelvin Sung Lawrence Snyder	ACM	7,0
[2]	Microworlds, Games and Collaboration: three effective approaches to support novices in learning programming	Stelios Xinogalos Christos Malliarakis Despina Tsompanoudi Maya Satratzemi	ACM Springer	6,5
[3]	Game programming in CS0: a scaffolded approach	Michael Panitz Kelvin Sung Rebecca Rosenberg	ACM	5,5
[4]	Learning and practicing decision structures in a game	Jinghua Zhang Emanuel Smith Elvira R. Caldwell Mathew Perkins	ACM	7,5
[5]	Games, stories, or something more traditional: the types of assignments college students prefer	Daniel C. Cliburn Susan Miller	ACM	6,5
[6]	A games first approach to teaching introductory programming	Scott Leutenegger Jeffrey Edgington	ACM CiteSeerX	5,0
[7]	Game-themed instructional modules: a video case study	Robin Angotti Cinnamon Hillyard Michael Panitz Kelvin Sung Keri Marino	ACM	7,0
[8]	Serious Game and Students' Learning Motivation: Effect of Context Using Prog&Play	Mathieu Muratet Elisabeth Delozanne Patrice Torguet	ACM Springer	5,0

		Fabienne Viallet		
[9]	CS1, arcade games and the free Java book	Daniel L. Schuster	ACM	5,5
[10]	Using games in introductory courses: tips from the trenches	Jessica D. Bayliss	ACM	6,0
[11]	Wu's castle: teaching arrays and loops in a game	Michael Eagle Tiffany Barnes	ACM	8,0
[12]	Game-themed programming assignments for faculty: a case study	Cinnamon Hillyard Robin Angotti Michael Panitz Kelvin Sung John Nordlinger David Goldstein	ACM	8,0
[13]	Teaching Introductory Programming as an Online Game	Ben Leong Zi Han Koh Ali Razeen	CiteSeerX	7,0
[14]	The Effectiveness of Games as Assignments in an Introductory Programming Course	Daniel C. Cliburn	IEEEExplorer	6,5

4. Análise dos artigos selecionados

Ao analisar os artigos primários foram identificadas três estratégias no uso de jogos no apoio ao ensino introdutório de programação:

- **Uso de jogos no ensino de programação:** Nesta categoria um jogo é desenvolvido e utilizado como suporte ao ensino de programação. O aluno então irá jogar e terá desafios ligados a programação através do jogo. Dentre os artigos primários desta revisão sistemática se enquadram nesta categoria os seguintes trabalhos: [2], [4] [8] e [11].
- **Ensino de programação baseado na programação de jogos:** Nesta categoria os alunos realizam tarefas de programação com foco na concepção de um determinado jogo (ou parte dele). Os conceitos de programação são então ensinados com foco no desenvolvimento de um jogo. Se enquadram nessa categoria os seguintes trabalhos: [1], [3], [5], [6], [7]. [9], [10], [12] e [14].
- **Gamificação do ensino de programação:** Nesta categoria as tarefas de programação são no estilo tradicional, porém a dinâmica da disciplina é *gamificada* a partir da criação de *leaderboards* (placares), conquistas e medalhas. Se enquadra nesta categoria o artigo [13].

A análise de cada estratégia é apresentada nas Subseções 4.1, 4.2 e 4.3.

4.1. Uso de jogos no ensino de programação

Foi observado que esta estratégia é utilizada como um apoio ao ensino de alguns conceitos de programação. Normalmente os alunos têm o jogo a sua disposição para utilizar em casa. Com isso, os autores buscam que os alunos vejam o jogo como uma diversão e não como uma obrigação. O jogo apresentado em [2] é utilizado para reforçar o ensino de *arrays*, em [4] estruturas de decisão, [8] é utilizado durante todo o curso de programação abordando os conceitos vistos nas aulas, e [13] aborda *arrays* e *loops*.

Como os alunos podem escolher utilizar o jogo ou não, naturalmente, são criados os grupos de controle e experimental dos estudos. A exceção é o trabalho

apresentado em [8] onde existiam turmas em que jogo era obrigatório, turmas em que ele era opcional e turmas sem acesso ao jogo.

A partir da comparação das notas entre os grupos experimentais e de controle, aliados a questionários e entrevistas durante o decorrer das disciplinas, pôde-se avaliar a eficácia da aplicabilidade desta estratégia. O que foi observado pelos pesquisadores é que tanto o desempenho na disciplina quanto a motivação dos alunos que jogaram aumentou em relação aos alunos que escolheram não jogar. No geral, a avaliação dos alunos quanto ao uso do jogo é positiva com a manifestação de que deveriam existir mais jogos para auxiliar o aprendizado durante o curso.

Uma ressalva em relação a estes trabalhos refere-se às conclusões que eles apresentam, que devem ser entendidas como indícios de que o uso de jogos melhora o processo de aprendizagem, especialmente porque os relatos tratam-se da experiência em apenas um semestre letivo. Fazem-se necessárias mais experimentações para que se possa de fato concluir se que o aprendizado e o interesse dos participantes pela programação realmente aumentam com o uso de jogos digitais.

Um fato apontado como negativo é que o uso de jogos apresenta a dificuldade relacionada ao projeto e à implementação do jogo, que demandam tempo e esforço por parte dos professores.

4.2. Ensino de programação baseada na programação de jogos

Esta estratégia é na realidade uma adaptação do ensino tradicional de programação. Os professores buscam aumentar o interesse nos alunos trocando as tarefas de programação tradicional por tarefas que gerem artefatos ligados ao desenvolvimento de um jogo. O que se tem então é a adaptação do programa da disciplina para que as tarefas sejam todas associadas ao desenvolvimento de um jogo. Alguns trabalhos definiam qual o jogo que seria desenvolvido, enquanto outros permitiam aos alunos a escolha o tipo de jogo a ser desenvolvido.

A maioria dos trabalhos encontrados que se valem desta estratégia não adotaram um grupo de controle e um experimental em paralelo. Aqui, a estratégia de programação de jogos foi adotada na disciplina, comparando o seu resultado com os resultados de semestres anteriores, nos quais o ensino de programação não abordava programação de jogos. Os trabalhos apresentam a aplicação desta estratégia em mais de um semestre letivo, o que torna o resultado da comparação mais efetivo.

No geral, o que se pode concluir é que o uso desta estratégia engajou e melhorou o aprendizado dos alunos. Isso foi verificado a partir da comparação das notas dos alunos e através de entrevistas e questionários ao longo das disciplinas. Mas também levantou o questionamento de até que ponto isso ocorre dado que nem todos os alunos se sentem motivados em desenvolver um jogo (no artigo [3] vários alunos disseram preferir que o curso fosse ministrado da forma tradicional). Uma solução apresentada para este problema é permitir que o aluno escolha se deseja ou não que as tarefas de programação sejam orientadas à programação de jogos (como ocorre em [5] e [14]).

A estratégia apresentada nesta seção tem a vantagem de não requerer do professor o desenvolvimento de um artefato (jogo) para o apoio ao ensino, sendo mais fácil a adaptação das tarefas de programação para tarefas orientadas a programação de jogos. Há também a vantagem de permitir, caso seja do interesse do professor, a escolha

dos alunos pela execução de tarefas orientadas a programação de jogo ou tradicionais de programação.

4.3. Gamificação do ensino de programação

A estratégia de *gamificação* é muito citada na literatura mas, nesta revisão só foi encontrado um artigo que fez uma análise que fosse além da simples percepção do professor sobre sua aplicação.

Neste trabalho o curso em si não foi alterado, as aulas e as tarefas de programação continuaram as mesmas, só que a disciplina foi *gamificada* a partir da criação de uma história baseada no universo de Star Wars, com as tarefas sendo vistas como missões, e com a inserção de recompensas típicas da *gamificação* (medalhas, placar de líderes, etc.) Mesmo com a disciplina tendo sido *gamificada*, foi facultado aos alunos cursar a disciplina da maneira tradicional. Com isso, os autores criaram de forma espontânea os grupos experimental e de controle.

A avaliação ocorreu a partir da comparação do desempenho entre os grupos experimental e de controle, da comparação com o desempenho das turmas de semestres anteriores (últimos 4 anos) onde a disciplina não era *gamificada*, e a partir de questionários e entrevistas aplicados com os alunos. O que foi observado é que o desempenho dos alunos melhorou e que o tempo de realização das tarefas de programação foi reduzido.

Uma ressalva é que, embora a comparação tenha sido relacionada a várias turmas anteriores, a aplicação da *gamificação* ocorreu em apenas um semestre da disciplina, sendo necessária a aplicação em mais turmas para realmente validar a estratégia.

5. Considerações finais

Embora vários estudos apontem que o uso de jogos como auxílio no ensino introdutório de programação melhora o engajamento e o aprendizado, a execução desta revisão sistemática da literatura apontou que ainda existe a escassez de evidências científicas que comprovem as alegações assumidas por alguns autores, sendo necessário estudos com avaliações com mais rigor científico e metodológico. Pode-se afirmar que existem indícios dessa melhora no aprendizado, mas estes precisam ser comprovados a partir de mais aplicações das estratégias apresentadas no apoio ao ensino introdutório de programação.

O que se pode concluir é que, embora já se discuta amplamente se jogos auxiliam ou não o ensino de programação, sua aplicação e avaliação ainda se situa em fase embrionária. Faz-se necessário que as avaliações sejam críticas, ponderando os pontos positivos e negativos, para que se possa verificar até que ponto o uso de jogos auxilia e a partir de que ponto ele se torna inócuo. O artigo [10] desta revisão seguiu esta linha e, ao invés de advogar de que o uso de jogo é a “bala de prata” para o engajamento e melhoria do aprendizado, apresenta até que ponto o uso de jogos melhorou o processo de aprendizagem e suas possíveis desvantagens.

Foi observado que existem vários trabalhos que apresentam o uso de jogos para o ensino de programação em contextos diferentes do foco desta revisão: ensino fundamental e básico, disciplinas não-introdutórias de programação e disciplinas de

introdução à programação aplicadas a cursos que não são da área de computação. Um dos trabalhos futuros que serão realizados trata-se de um levantamento de como as estratégias identificadas têm sido exploradas em cursos introdutórios de programação, considerando outros níveis de educação, como é o caso do ensino básico. Tais resultados serão relevantes para gerar mais conhecimento sobre o uso de jogos no ensino de programação, bem como, se figurar com um recurso para que professores de programação adaptem as experiências de outros nos seus próprios cursos.

Concluindo, acredita-se que o uso de jogos no ensino introdutório de programação traz resultados positivos para o aprendizado e engajamento. Os autores desta revisão pretendem desenvolver uma metodologia de uso de programação de jogos a ser aplicada e avaliada junto aos alunos introdutório de sua universidade.

Como propostas de trabalhos futuros podemos destacar:

- Realizar uma revisão sistemática da literatura visando levantar todos os trabalhos já realizados e cientificamente avaliados sobre o uso de jogos no ensino introdutório de programação em outros níveis de ensino;
- Realizar um estudo sobre os aspectos pedagógicos do uso de jogos no ensino de programação;
- Desenvolver uma metodologia para o ensino de programação apoiada por jogos, que será experimentada junto à disciplina de introdução a programação na Universidade Federal da Paraíba, auxiliando com isto, o processo de validação dos efeitos do uso de jogos digitais no ensino introdutório de programação.

Referencias

- D'Souza, D., Hamilton, M., Harland, J., et al. (2008). Transforming learning of programming: A mentoring project. *Conferences in Research and Practice in Information Technology Series*, v. 78, p. 75–84.
- De-La-Fuente-Valentín, L., Pardo, A. and Delgado Kloos, C. (2013). Addressing drop-out and sustained effort issues with large practical groups using an automated delivery and assessment system. *Computers and Education*, v. 61, n. 1, p. 33–42.
- Dybå, T. and Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology*, v. 50, n. 9-10, p. 833–859.
- Fowler, A. and Cusack, B. (2011). Kodu Game Lab: Improving the motivation for learning programming concepts. [M. Cavazza, K. Isbister, & C. Rich, Eds.]In *Foundations of Digital Games*. . ACM.
- Kinnunen, P. (2009). Challenges of Teaching and Studying Programming at a University of Technology – Viewpoints of Students, Teachers and the University. Helsinki University of Technology.
- Kunkle, W. M. (2010). The impact of different teaching approaches and languages on student learning of introductory programming concepts. Drexel University.
- Major, L., Kyriacou, T. and Brereton, O. P. (2012). Systematic literature review: teaching novices programming using robots. *IET Software*. IET. <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6353335>.

- Matemática e ciências da computação têm alta taxa de desistência (2009). *Folha de São Paulo*,
- Mattos, M. M. and Fábio, F. J. (2007). Qualifica: Uma Ferramenta para Apoio a Construção de Algoritmos Estruturados. In *Anais do XVI Seminário de Computação Furb*. . http://www.inf.furb.br/seminco/2007/artigos/07_34048.pdf.
- Microsoft Corporate Citizenship: U.S. Talent Strategy (2007). . <http://news.microsoft.com/download/presskits/citizenship/MSNTS.pdf>.
- Muratet, M., Delozanne, E., Torguet, P. and Viallet, F. (2012). Serious game and students' learning motivation: Effect of context using Prog&Play. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11th International Conference on Intelligent Tutoring Systems, ITS 2012. v. 7315 LNCS, p. 123–128.
- Piteira, M. and Haddad, S. (2011). Innovate in your program computer class: an approach based on a serious game. In *Workshop on Open Source and Design of Communication*. . <http://dl.acm.org/citation.cfm?id=2016730>.
- Salen, K. T., Torres, R., Wolozin, L., Rufo-Tepper, R. and Shapiro, A. (2011). *Quest to Learn: Developing the School for Digital Kids*. London: The MIT Press.
- Scaico, P. D., Dantas, V. F., Farias, A. B., et al. (2013). Atividades Virtuais de Apoio Aplicadas ao Ensino Introdutório de Programação. In *XXXIX Latin American Computing Conference, Puerto azul/Naiguata/Venezuela. XXI Iberoamerican Congress on Higher Education in Computing (CIESC-2013)*.
- Vihavainen, A., Paksula, M. and Luukkainen, M. (2011). Extreme apprenticeship method in teaching programming for beginners. *Proceedings of the 42nd ACM technical symposium on Computer science education - SIGCSE '11*, p. 93.