# Relação estudante-professor:

# Educação Baseada na Construção de Jogos

Carlo Emmanoel Tolla de Oliveira<sup>1</sup>, Érica Calil Nogueira<sup>1</sup>, Claudia Lage Rebello da Motta<sup>1</sup>, Ludmila Barros Meireles<sup>1</sup>

<sup>1</sup>Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais – Universidade Federal do Rio de Janeiro (UFRJ) Rio de Janeiro – RJ – Brazil

(carlo, erica.noqueira, claudiam, ludmila.meireles)@nce.ufrj.br

Abstract. Games are fun and can become the best form of education. This article aims to show the process of teaching students and teachers to create their own games. Programming games, students and teachers can release your creativity and innovate their own education. In this process is used a particular game platform training for the Internet, together with a tutorial introducing the games construction. Students and teachers discover this innovative capability and plan new ideas that exploit programming as an educational tool.

Resumo. Jogos são divertidos e podem se tornar a melhor forma de educação. Este trabalho mostra o processo de ensinar estudantes e professores a criar seus próprios jogos. Programando jogos, estudante e professores libertam sua criatividade e podem inovar sua própria educação. Neste processo é usada uma plataforma específica para o ensino de jogos para a internet juntamente com um tutorial introduzindo à construção de jogos. Estudantes e professores descobrem esta potencialidade inovadora e planejam novas ideias que exploram a programação como meio educacional.

## 1. Introdução

Nas últimas décadas, a explosão do desenvolvimento tecnológico tem contribuído para um acesso rápido e fácil à informação. Novas formas de mídia, como a internet e os games, são massivamente utilizadas para o entretenimento e diversão. Em contra partida, há uma grande busca por meios inovadores e efetivos na educação - em todos os seus níveis. De que forma pode-se evitar o método repetitivo e exaustivo que é o sistema educacional atual, de forma a se ter certeza de que o estudante terá prazer e entendimento do que aprende? Tendo como motivação encontrar uma forma de tornar a educação divertida, resolveu-se utilizar os jogos como uma ferramenta para ensinar conceitos e idéias interdisciplinares.

Assim, este artigo descreve o trabalho que está sendo feito para ensinar crianças e professores a criar jogos educativos e divertidos. Os estudantes desenvolvem diferentes áreas do conhecimento ao mesmo tempo, aprendem a linguagem do século XXI - a programação - aprendem a ensinar - pois criam jogos educativos - e aprendem com prazer.

Porém, apesar de todas essas qualidades, o maior êxito deste projeto está no fato de que toda essa atividade quebra barreiras e estimula uma das características humanas

DOI: 10.5753/cbie.sbie.2015.637

mais em falta no mundo: a criatividade. Dessa forma, os estudantes que utilizarem este método serão pessoas mais conscientes de si e do mundo, e buscarão maneiras inovadoras de resolver os problemas da sociedade.

# 2. Proposta

O ensino de linguagem de programação no ensino básico tem sido discutido em diversos países, inclusive sendo já incluído na legislação de alguns. A importância da programação de computadores na educação pode ser vista a partir da competência de formalizar as ideias que advêm do seu uso. Ao transformar ideias em um programa de computador elas tem que passar por um processo de ordenamento onde se analisa cada parte e considera o papel de cada uma no todo. A ideia passa por uma fatoração de suas partes constituintes, e cada uma ganha corpo e significado individual e entrelaçado com o todo. Esta atribuição de significado se apoia e se integra em toda uma rede preexistente de conceitos já formalizados pelo indivíduo pensante. Desta forma esta rede forma vínculos com estas novas ideias, fixando-as como conceitos pertinentes e coesos com toda a formalização já alcançada pelo indivíduo. Todo este processo é muito relevante para a educação pois a rede de significados resultante é um conhecimento mais operante e durável do que apenas uma aquisição memorizada de um assunto. Quando a programação está ligada à construção de jogos, a aprendizagem é motivada por um prazer lúdico. Este prazer leva o aprendiz a perseverar na busca de seu intento, e o faz avançar superando as frustrações advindas dos erros.

O modelo lúdico aqui proposto é o jogo de fazer o jogo. A base da metodologia pedagógica é uma adaptação da base neurocientífica cognitiva que utiliza a técnica de grupos operativos (Pichon-Reviére, 1998) e elaboração dirigida (Seminério, 1987) com aplicação do Fio Condutor (Marques, 2009). O ensino baseado em problemas é o patamar inicial do Fio Condutor que se expande em um modelo de neuropedagogia computacional. Nesta pedagogia o computador é o paramediador onde ele é o prórpio estímulo e desafio que encaminha o desenvolvimento do aprendizado. É importante que logo nos primeiros passos se tenha a percepção de que o jogo já está sendo criado. Quando o estudante percebe que o jogo já se descortina, ele sente que é capaz de criá-lo. Este sentimento de autoeficácia (Bandura, 1977) se transforma em um estímulo interno onde o estudante sente que tem a competência de ir adiante e construir algo que não havia tentado antes. Mesmo estudantes dispersos ou com fragilidades em sua base teórica irão concentrar sua atenção no projeto e tentar resolver os desafios. Essa é a mesma mecânica dos jogos virtuais que oferecem desafios aparentemente insuperáveis, mas que ao serem superados oferecem confiança e a experiência de se ter conquistado algo. O resultado é o jogador seguir obstinadamente por todas as provas oferecidas pelo jogo até que supere todos os desafios.

O princípio que leva o jogador a enfrentar situações muitas vezes frustrantes e continuar tentando pode se aplicar em qualquer ambiente ou problema que o jogo lhe ofereça. Por isso a gamificação é um processo adotado em diversas áreas da atuação humana nos dias de hoje (Medina, 2014). A gamificação é o processo de converter tarefas do dia a dia, em especial as ligadas com educação, em uma forma de jogo. A gamificação toma como base um conteúdo ou uma tarefa que precisa ser assimilada ou executada e encapsula no ambiente e na mecânica de um jogo virtual. Na proposta de se capacitar na construção de jogos, a gamificação é uma opção natural de metodologia. Em um jogo virtual, o jogador não recebe instruções detalhadas, ele apenas é colocado

diante da situação problema e tem que pensar a sua própria solução. Esta premissa instiga o jogador a usar o seu raciocínio e sua criatividade para inferir e construir a sua própria resposta. Com o raciocínio o jogador pondera os fatos a que tem acesso, analisa e contrapõe todos os detalhes no contexto em que se encontra. O jogo se lhe apresenta como uma ideia incompleta que precisa ser fechada com uma intervenção criativa sua. O jogador faz uso do seu imaginário e busca nele informações que possam se encaixar e completar o quebra-cabeça que o jogo apresenta. Isto consiste em um exercício de criatividade delimitada pelas condições de contorno que o jogo oferece. E quando o problema é solucionado o resultado é o indivíduo que a partir de seus próprios esforços e recursos se capacitou em resolver este tipo de problema.

Em um jogo o problema se apresenta sempre como uma situação real e concreta onde o jogador está envolvido completamente. A concretude, mesmo que virtual, da situação urge ao jogador que tome uma atitude para solucioná-la. No seu imaginário, cada segmento da situação é confrontada com episódios dos mais variados que já presenciou ou conheceu. Esta diversidade não tem limites e todo o tipo de conhecimento pode ser confrontado com a situação problema. Esta abertura do imaginário concede que diversas áreas do conhecimento confluam e se entrelacem nas tentativas de analisar e resolver o problema. Esta oportunidade igual para todos os conhecimentos é uma vantagem para um processo de educação gamificado. Ela facilita com que o estudante tome contato com novas áreas, mesmo que o estudante não tenha muita familiaridade com o assunto. A situação concreta oferecida pelo jogo justifica o acesso a essas novas áreas e não causa um estranhamento que crie uma aversão inicial à novidade. Esta vantagem permite que se ensine assuntos que sejam completamente novos aos estudantes sem que seja formado um preconceito prejudicial à assimilação de novos conceitos. No caso particular da programação de computadores, o assunto já goza de uma reputação mística de ser difícil e inacessível a maioria das pessoas. No entanto, jogos, pelo contrário, são conhecidos e aceitos pela maioria das pessoas. Ao se ensinar a programação em forma de um jogo as barreiras vão se diluindo e o estudante penetra no assunto sem maiores problemas, pois confia na sua capacidade de jogar um jogo.

Um jogo, em sua essência, envolve uma gama de conhecimentos e habilidades para se construir. Ou seja, construir um jogo é essencialmente uma atividade interdisciplinar. Este é uma outra vantagem deste modelo de educação gamificada. A exclusividade disciplinar, apesar de ser um modelo didático plenamente empregado, é menos eficaz em formar competências necessárias ao mundo do século XXI. Ao se propor a construção de um game, o desafio se estende a todas as áreas envolvidas no processo. Um jogo virtual requer minimamente o envolvimento com a arte, a linguagem e a lógica. O estudante então terá que dominar simultaneamente estes três conhecimentos que se entrelaçam a cada momento na construção do jogo.

Na construção de um jogo sempre existe uma narrativa que vai conectar as cenas e episódios dentro de uma sequência e interação lógica entre as partes. Na nossa proposta, a primeira instância da narrativa é história da construção do jogo. O jogo inicia com um cenários estático que constitui o fundo da narrativa. Este cenário vai sendo descrito a medida que o programador decide as decorações que vão sendo acrescentadas, uma árvore, uma rocha, um muro e todo o resto. Existe então uma intercessão com a arte, na construção harmoniosa deste cenário, segundo a estética de cada um. Em seguida vai sendo construída a figura do personagem, com suas partes constituintes. O personagem inicialmente se confunde com o cenário, pois nasce

estático, sem nenhuma função aparente. A narração então segue por dar vida ao personagem a medida que novas funções são agregadas a ele. Agora uma nova interação é feita com a arte cinematográfica no entendimento de como animar o movimento do personagem e suas partes. A lógica vem por final na interação do personagem e o mundo, como o personagem se detêm diante dos objetos do cenário e como ele pode atuar em cada um deles.

#### 3. Processo

Uma boa estratégia no ensino de programação de computadores é o uso de software livre. O software livre garante que o material de ensino e tudo que é produzido seja isento de taxas de comercialização que limitariam o escopo do projeto. O uso de software livre também facilita a personalização do produto para atender a requisitos didáticos. Neste projeto colaboramos com programadores internacionais na construção de uma plataforma de ensino que ajude a popularizar a programação de computadores. A medida que a plataforma ia sendo construída ela ia sendo testada pela nossa equipe nos nossos cursos. Os problemas e melhoramentos eram comunicados aos programadores, que iam modificando a plataforma para que se tornasse mais eficaz. Por fim nossa equipe introduziu algumas modificações para que plataforma tivesse um melhor aproveitamento na nossa didática.

A base do nosso modelo de ensino de programação é a linguagem Python, mais especificamente o Brython (Quentel 2014) que é um projeto que modifica o Python para ser usado em navegadores da internet. Esta modificação tem a vantagem de não necessitar a instalação de nada na máquina do estudante, bastando que se use um navegador moderno atualizado. A plataforma de ensino é feita pela mesma equipe do Brython, é o Brython-in-the-classroom ou pyschool.net (Earney 2014). O pyschool.net disponibiliza um editor de programas e um campo de provas onde se pode testar o programa e construir jogos. Para atender as nossas necessidades criamos um pacote para construção facilitada de jogos, o Brython-Crafty, que adapta um pacote já existente (Crafty) à linguagem Brython. Todos estes pacotes são software livre e podem ser baixados por qualquer um que queira ensinar programação. Entende-se que a necessidade de ensinar programação a todos é algo percebido a nível mundial e hoje diversas iniciativas neste sentido estão sendo elaboradas e aplicadas.

O nosso modelo de ensino se propõe a capacitar estudantes e professores a construir seus próprios jogos. Neste experimento em particular, está sendo desenvolvido pela turma um jogo de ensinar a programar. Esta técnica é muito usual e já tem sido usada a muito tempo. Vários jogos foram desenvolvidos com este propósito sendo que RoboCode e Code Commander foram as inspirações mais próximas do nosso projeto. O RoboCode é um jogo desenvolvido em Java (Nelson 2014) como uma competição de tanques de guerra. Cada participante envia seu código que comanda o tanque para se livrar dos ataques e destruir os outros. O Code Commander é um outro jogo de tanques, desenvolvido em JavaScript (Morris 2014) onde se programa um tanque em diversos níveis de combate, onde em cada um deles ele enfrenta obstáculos e inimigos mais difíceis. Nestes dois jogos o estudante usa o jogo pronto e desenvolve em programação a estratégia de seus tanques. A nossa proposta pega elementos destes dois jogos sendo que primeiro os estudantes terão que desenvolver o jogo de combate antes de comandar seus tanques.

No nosso modelo, em vez de encontrar o pacote do jogo pronto, o jogo terá que ser desenvolvido a partir do zero. O estudante entra na plataforma e a única coisa que tem disponível é o pacote de construção de jogos e um tutorial de como criar o jogo de tanques (Figura 1). O tutorial foi desenvolvido para ser integrado à nossa versão da plataforma pyschool.net e fazer uso das modificações que facilitam a construção dos jogos. O tutorial apresenta as funções do construtor de jogos e cria desafios para que os programadores desenvolvam o jogo dos tanques. O tutorial é dividido em um conjunto de lições que vão gradativamente introduzindo novas facilidades de construção de jogos e dando a oportunidade para que os programadores a usem para avançar a funcionalidade do jogo (Figura 2). No primeiro módulo os estudantes desenvolvem o jogo onde o tanque que obedece interativamente a comandos do teclado. O próximo módulo permite que o tanque seja controlado por programa e que os tanques de cada estudante se conectem e enfrentem uns aos outros. As estratégias de enfrentamento são desenvolvidas por cada estudante usando a criatividade e a lógica.



Figura 1. Plataforma Pyschool.



Figura 2. Primeira lição, em que os estudantes desenvolvem o cenário do jogo.

Na metodologia, o tutorial é oferecido para uma turma onde professores e estudantes sentam lado a lado. O professor, para o qual este assunto é novidade, tem a oportunidade de experimentar as mesmas dificuldades que os estudantes sentem ao primeiro contato com novos conceitos. Ele terá que seguir os mesmos caminhos de todo o estudante na descoberta de suas capacidades e bloqueios (Figura 4). É uma oportunidade para o professor manifestar uma nova modalidade de educação, onde ele

não é o senhor do conhecimento mas a sua experiência fala mais alto. Uma educação onde o professor é líder que guia uma expedição de estudantes em busca de descobertas comuns. O processo emprega o desafio como a didática, colocando o estudante (e o professor) diante do problema que ele vai solucionar com a sua perspicácia e criatividade. O tutorial pondera entre o que pode ser oferecido pronto e aquilo que o estudante já pode opinar e solucionar. Ele cria uma base estável onde o estudante pode construir a sua solução (Figura 3). Ao criar a sua solução o estudante ganha autonomia e se empodera na capacidade de criar o seu próprio jogo.

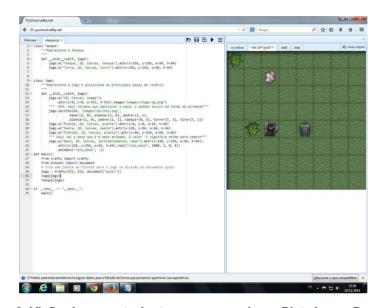


Figura 3. Visão de um estudante programando na Plataforma Pyschool.



Figura 4. estudantes discutindo e programando.

Uma sala de aula como esta está repleta de disparidades entre as diversas competências preexistentes e o ritmo com o que os estudantes adquirem o conhecimento. O trabalho do professor de programação é atender as dificuldades de cada estudante, observando onde estão os bloqueios. Muitos bloqueios são pela falta de atenção que é necessária em um trabalho de programação. Estes devem ser alertados

diretamente ao estudante para que ele possa prosseguir. Outros são dificuldades conceituais da natureza da tarefa e se tornam oportunidades para que se introduza uma explicação que seja importante para toda a turma.

## 4. Aplicação

O projeto consegue se adaptar tanto para laboratórios fixos quanto para itinerantes (Figura 5). Isso se deve aos tipos de dinâmicas de aprendizado que podem ser utilizadas. Para os laboratórios fixos são selecionados estudantes prioritariamente da rede pública de ensino que se deslocam até a universidade para a aula semanal que perdurará durante todo o ano letivo. Para os laboratórios itinerantes a equipe se desloca um ou mais dias agendados e é feita uma dinâmica de workshop com os estudantes e demais presentes.

No sistema de aulas ministradas na universidade - os laboratórios fixos - os estudantes tem mais tempo de adaptação com a equipe e aprendem a se organizarem entre as tarefas sem distinções na equipe de quem era estudante, professor, monitor ou assistente já que todos são englobados no mesmo projeto e ficam responsáveis por partes distintas do trabalho. No sistema itinerante se propõe a dinâmica de workshop, selecionando etapas do projeto que correlacionem as interações entre estudantes e equipe com partes selecionadas do conteúdo. A dinâmica do workshop pode ser feita em escolas, museus, centros científicos ou qualquer local que disponibilize computadores com acesso a internet em ambiente sadio para os estudantes.

Para os casos itinerantes o intuito é levar oficinas dinâmicas que fomentem a vontade dos usuários em continuar a partir do progresso em sala. São desenvolvidas tarefas mais rápidas em trabalhos de um ou poucos dias. O formato itinerante começou com a proposta de levar aos colegas dos estudantes recebidos pelo laboratório fixo uma amostra do trabalho realizado na universidade de forma que eles pudessem conhecer o trabalho realizado. Para início dessa etapa o formato itinerante consistia em a equipe ir ao encontro dos estudantes em sua área, sua escola e ministrar uma oficina do Flying Circus. A evolução do projeto criou novas visões sobre as possibilidades do laboratório itinerante, como o uso do celular no local do notebook e a ideia de não restringir as oficinas nas escolas.



Figura 5. Estudantes programando em um laboratório universitário.

## 5. Avaliação

Dentro do conceito do projeto, o sujeito agente, um estudante-professor, dotado de suas faculdades para aprendizagem e ensino, pode e deve comungar o conhecimento ali gerado. Esse conceito estudante-professor é inserido no momento em que não há divisão de níveis dentro de sala e todos, sejam estudantes, os professores responsáveis que os levam e a equipe do projeto estão sentados juntos programando games. Cada um com suas competências e conhecimentos colabora para o enriquecimento da turma, mostrando a necessidade da compreensão de que não se julga o saber mais ou saber menos, mas se compartilha os saberes diferentes.

Nosso primeiro caso é um menino de sete anos, da rede municipal de ensino que viu no programa a oportunidade de criar seu próprio jogo com uma equipe de colegas, estudantes-professores, que o auxiliariam. O menino por conta própria não só definiu o jogo e sua equipe como estudou mais a fundo a plataforma no intuito de encontrar as ferramentas necessárias para construir o game. Não houve orientação prévia sobre os mecanismos. O estudante, observando a lógica da plataforma, decifrou a área de conteúdos para aprender os mecanismos necessários para a formulação de seu projeto de jogo. O modelo estudante-professor foi importante nesse caso para que o aluno tivesse a oportunidade de se sentir à vontade para trocar experiências com os mais velhos sem se sentir pressionado pela hierarquia. Assim foi capaz de delegar funções para otimizar o tempo da equipe quando achava necessário, sem repreensões ou desmoralização por parte dos mais velhos.

No segundo caso um historiador, professor voluntário de matéria adjacente a usada para implementação da Plataforma Flying Circus, que nunca em outro momento havia se deparado com o desafio de programar algo, participou de um workshop da Plataforma junto de seus estudantes da graduação, muitos destes já letrados em programação. O momento partilhado entre os estudantes e o professor voluntário naquele workshop validou teorias vistas em sala - como o encadeamento de escolhas para amarrar o enredo - dentro da aplicação na montagem real do jogo. Usando a Plataforma, ele, o historiador, se afeiçoou ao programa e às dinâmicas utilizadas pela equipe. Essas dinâmicas são utilizadas para agilizar o processo de criação a partir dos descobrimentos dos usos da Plataforma. O historiador se manteve ativo na Plataforma nas semanas seguintes, desenvolvendo a continuação das tarefas do workshop e indo além, construindo uma estrutura de calculadora plenamente funcional.

Por último temos o exemplo da estudante do ensino médio técnico em informática, acostumada com o padrão tradicional de ensino, onde o professor leciona unilateralmente para os alunos. Dentro do projeto, conforme as aulas foram evoluindo, a estudante foi criando sua estrutura interna de estudante-professora. Nessa experiência ela assimilou o mútuo aprendizado envolvido nas várias etapas do desenvolvimento de um game - matéria que não há no técnico. Como resultado, convidou a equipe do Labase/NCE-UFRJ para sua escola para promover uma oficina de aprendizagem de programação diretamente voltada para jogos usando a Plataforma Flying Circus. Ela como assistente da matéria, fomentou a vontade dos seus colegas a também aderirem ao processo criado pela experiência do projeto. Após essa oficina a própria estudante reuniu um grupo que se organizou para terminar o jogo da oficina. Nesse interim cada

um dos membros do grupo foi descobrindo suas aptidões para as áreas adjacentes necessárias ao game como sintetizar midis - formatos de música em bits -, criar sprites - sequências de imagens que criam gifs animados -, pesquisar materiais de apoio e aprender a fazer um *Game Developer Document*.

São exemplos como esses que fazem a eficácia do programa se confirmar, pois é um aprendizado dinâmico e lúdico que mostra que qualquer um pode programar. Ao contrário de um ensinamento exclusivo para pessoas aptas na área de exatas, como apenas para quem já tem uma base da lógica de programação. O Flying Circus demonstra que o senso comum não é verdadeiro. A plataforma tem a capacidade de transpor as barreiras dos níveis escolares e ensinar um mesmo produto final a todos com a singularidade de dar espaço para que cada um se utilize das ferramentas disponíveis da melhor forma que lhes convir encontrando sua própria maneira de evoluir na matéria.

### 6. Conclusão

Aprender uma linguagem de computador implica em adquirir a capacidade de organizar pensamentos soltos em um fluxo coerente de ações, concatenando causas e consequências. O processo de transformar a educação em jogos de computador traz satisfação imediata para a necessidade de saber, uma vez que, para que isso aconteça no jogo, é necessário desenvolver o processo de aprendizagem como ele realmente acontece. Agora temos a sinergia entre a necessidade imediata em saber do estudante e o professor pronto para oferecer o conhecimento e a orientação para que isso aconteça de forma eficiente.

A educação deste século exige que os estudantes sejam criados para serem autônomos e pró-ativos. O desenvolvimento de jogos requer uma desenvoltura para tomadas de decisões onde os estudantes que trabalham dentro deste formato de projeto mostram elevados índices de autonomia e criatividade. Eles trabalham em longas descrições de cenário, decidem pela criação de novos personagens e uma grande diversidade de artefatos interativos com roteiros diferentes para o desenrolar da trama. Essas habilidades são na sua maioria adquiridas devido à exposição densa a um ambiente rico de informações, mas uma educação baseada em jogos abre a oportunidade para que essas habilidades floresçam e sejam alimentadas.

Os computadores estão na vida diária de cada pessoa. Este fato cria uma nova divisão social entre alfabetizados e analfabetos digitais. Uma educação onde os professores e estudantes são parceiros na construção de jogos está preparada para fechar essa lacuna e abrir novos caminhos para o desenvolvimento humano. Em nossa experiência é visto que estudantes e professores encontram neste empreendimento uma motivação legítima para continuar sua jornada pela educação.

#### 6. Referências

Bandura, A. (1977). "Self-efficacy: Toward a Unifying Theory of Behavioral Change". Psychological Review 84 (2): 191–215.

Bunchball, Inc (2010) Gamification 101: An introduction to the use of game dynamics to influence behavior. Disponível em: <a href="https://www.bunchball.com/sites/default/files/downloads/gamification101.pdf">www.bunchball.com/sites/default/files/downloads/gamification101.pdf</a> Acesso em Dezembro.

- Earney, B. Brython-in-the-classroom. Disponível em: <a href="https://github.com/brython-dev/brython-in-the-classroom">https://github.com/brython-dev/brython-in-the-classroom</a> Acesso em Dezembro.
- Elia, M. F., Sampaio, F. F., Motta, C.L.R. (2003). u o i e a users oriented internet education architecture. In: II International Conference on Multimedia and Information & Communication Technologies in Education, 2003, Badajoz, Spain. Proceedings of theII International Conference on Multimedia and Information & Communication Technologies in Education. Badajoz, Spain: JUNTA DE EXTREMADURA Consejería de Educación, Ciencia y Tecnología. v. 2. p. 1147-1149.
- Freire, P. (2005). Pedagogia do Oprimido. Paz e Terra. Rio de Janeiro. 42.ª edição.
- Gee, J.P. (2003). What video games have to teach us about learning and literacy. Palgrave/St. Martin's. New York.
- Marques, C. V. M. et al. Uma abordagem meta-cognitiva na construção coletiva do conhecimento. Rio de Janeiro: NCE/UFRJ, 2009
- Marques, C. V. M. et al. Neuropedagogia e Informática I: A Revolução Cognitiva um estudo sobre a teoria de Franco Lo Presti Seminério. Rio de Janeiro: NCE/UFRJ,2009.
- Medina, B. (2014). Gamification, Inc. Como Reinventar Empresas a Partir de Jogos. 120 p. MJV Press.
- Mendonça, A., Marques, C. V. M., Oliveira, C. E. T. de, Rebello, C. (2009). Uma proposta de ensino de linguagem programação de computadores neuropedagógica. In: XX Simpósio Brasileiro de Informática na Educação, 2009. Sociedade Brasileira de Computação. Porto Alegre.
- Morris, R. Code Commander. Disponível em: <a href="http://morrisgames.info/">http://morrisgames.info/</a> Acesso em Dezembro.
- Nelson, W. Robocode Home. Disponível em: <a href="http://robocode.sourceforge.net/">http://robocode.sourceforge.net/</a> Acesso em Dezembro.
- Pichon-Reviére, E. O processo grupal. 6°ed. Rev. São Paulo: Martins Fontes, 1998.
- Quentel, Pierre, Brython. Disponível em: <a href="http://brython.info">http://brython.info</a> Acesso em Dezembro. Seminério, Franco Lo Presti. Elaboração dirigida: um caminho para o desenvolvimento metaprocessual da cognição humana. Rio de Janeiro: FGV; ISOP, 1987.
- Silva, V.S.R., Dargains, A. R., Felicio, S.P.A.S., Souza, P. R. A., Sampaio, F. F., Motta, C.L.R., Borges, M. R. da S., Gomes, J. O., Carvalho, P.V.R. (2014). STOP DISASTERS: SERIOUS GAMES WITH ELEMENTARY SCHOOL STUDENTS IN RIO DE JANEIRO. In: 8th International Technology, Education and Development Conference, 2014, Valencia, Spain. INTED2014 Proceedings. UK: Thomson Reuters. v. 1. p. 1648-1659.
- Simões, J. & Aguiar, A. (2011). Schoooools.com: A Social and Collaborative Learning Environment for K-6, em EDULEARN11 3rd Annual International Conference on Education and New Learning Technologies, EDULEARN11 Proceedings, Barcelona.