

Aprendendo Lógica de Programação por Meio de Jogos Digitais

Gladson Renato Queiroz Vidal¹⁴, Carlos Humberto Cruz Silva²⁴
Emanuel Ferreira Coutinho³⁴

¹Engenharia Elétrica

²Sistemas e Mídias Digitais

³Instituto Universidade Virtual (UFC Virtual)

⁴Universidade Federal do Ceará (UFC) – Fortaleza – CE – Brasil

gladson_renato@hotmail.com, carlosh@alu.ufc.br, emanuel@virtual.ufc.br

Abstract. *The understanding of algorithms and programming are extremely important points in the academic training of technology students. However, there are still difficulties in teaching and understanding these contents in the classroom, especially with the need for methodological changes. Having games as an efficient methodological tool, this work aims to develop games that could be implemented along with the basic programming discipline, in order to improve and facilitate the learning of the contents offered in the classroom. The proposed approach divided programming teaching into three major moments: “Initial presentation”, “Understanding syntax” and “Deepening content”. At the end of these steps, students are expected to have a more accurate understanding of algorithms and programming.*

Resumo. *O entendimento de algoritmos e programação são pontos de extrema importância na formação acadêmica de estudantes de tecnologia. Porém ainda se percebe dificuldades no ensino e compreensão desses conteúdos em sala de aula, principalmente com a necessidade de mudanças metodológicas. Tendo jogos como eficientes ferramentas metodológicas, este trabalho tem como objetivo o desenvolvimento de jogos que pudessem ser implementados junto à disciplina básica de programação, com o intuito de melhorar e facilitar o aprendizado dos conteúdos ofertados em sala de aula. A abordagem proposta dividiu o ensino de programação em três grandes momentos, sendo estes: “Apresentação inicial”, “Compreensão da sintaxe” e “Aprofundamento do conteúdo”. Ao final, dessas etapas, espera-se que os alunos possuam uma percepção mais precisa sobre algoritmos e programação.*

1. Introdução

A programação, bem como o estudo de algoritmos é um conteúdo essencial a todos os discentes nas áreas tecnológicas. Entretanto, podem ser citados, diversos problemas quanto ao ensino e aprendizagem de algoritmos e programação, culminando em desmotivação, evasão e até reprovação [Rapkiewicz et al. 2006].

Os jogos são ferramentas eficientes pois, divertem, motivam e facilitam a aprendizagem, aumentando a capacidade de retenção de conteúdo, além de exercitar as

funções mentais e intelectuais do jogador [Roland et al. 2004]. Diante disso, os jogos podem ser utilizados como poderosas aliadas nas táticas e estratégias pedagógicas [Rapkiewicz et al. 2006].

Uma das possibilidades presentes no tocante ao ensino de lógica de programação é o ensino “*Unplugged*” (desconectado) [CSUnplugged 2015]. Ou seja, trata-se do ensino de lógica, sem necessariamente o uso do computador. Dentro dessa temática, o projeto do CS Education Research Group da University of Canterbury, NZ, intitulado CS Unplugged (*Computer Science without a computer*), foi desenvolvido. O CS Unplugged é uma coleção de atividades gratuitas de aprendizado que ensinam Ciência da Computação através de jogos, e sem o uso de computador [CSUnplugged 2018]. São basicamente atividades que introduzem os alunos no Pensamento Computacional através de conceitos como números binários, algoritmos e compressão de dados. O sistema Unplugged é usado em todo o mundo há mais de vinte anos, sendo adequado para pessoas de todas as idades, desde o ensino fundamental até os idosos.

Além deste, outros métodos vêm sendo amplamente utilizados para o ensino de programação computacional. Um exemplo é o VisualG¹, que se propõe, a desenvolver programação, com pseudocódigo. O uso do pseudocódigo permite que o aluno se concentre inicialmente na solução do problema proposto sem ter que dominar a sintaxe de uma linguagem de programação, porém já apresentando conceitos mais profundos como declaração de variáveis, linhas de código, estruturas de repetição, estruturas condicionais, dentre outros, que podem ser depois transpostos com mais facilidade para ambientes reais de desenvolvimento [de Souza 2009].

Diante disto, objetivou-se neste trabalho, o desenvolvimento de jogos que pudessem ser implementados junto a disciplina básica de programação, com o intuito de melhorar e facilitar o aprendizado dos conteúdos programáticos ofertados em sala.

Nesse contexto, é comum em cursos de graduação disciplinas com a intenção de alinhar a teoria com a prática. Entretanto elas são normalmente muito difíceis de se conduzir devido à necessidade de se tratar diversos fatores técnicos (linguagens de programação, componentes, ferramentas etc.) e humanos (disponibilidade, comunicação, gestão etc.), tudo ao mesmo tempo [Coutinho et al. 2016]. Além disso, a atividade de motivar os alunos e motivar os próprios professores não é uma tarefa fácil, se verificando particularmente em disciplinas de primeiro ano de cursos de graduação. Isto se deve ao fato que os conteúdos abordados são na maioria das vezes, novidade para os alunos constituindo assim um desafio para todos.

O aprendizado em linguagens de programação necessita de conhecimentos prévios em lógica e matemática [Ribas et al. 2016]. Mesmo que o raciocínio lógico não seja trabalhado exclusivamente em disciplinas de Matemática, é nessa área que geralmente esses conhecimentos são mais explorados nas escolas. Nesse sentido, seria possível afirmar que cada vez mais os estudantes chegam com menor conhecimento prévio sobre matemática e lógica nos cursos de graduação. Vários trabalhos na literatura discutem diferentes relatos do ensino de lógica de programação e codificação em semestres iniciais de cursos de graduação [Ferreira et al. 2016][Marinho et al. 2016][Coutinho et al. 2017]. Uma dificuldade comumente discutida é como ensinar lógica de programação de maneira que o

¹<http://visualg3.com.br/>

aluno se motive a aprender, ao mesmo tempo em que se preparam para a utilização de ferramentas de codificação mais avançadas e aplicar conceitos de lógica de programação na prática. Motivação também é muito importante para o docente, pois muitas vezes este fica desmotivado com o desempenho na disciplina.

Uma maneira de facilitar e incentivar o aprendizado é a utilização de jogos para o ensino. Nesse contexto, jogos têm sido utilizados para ajudar no ensino de diversas áreas do conhecimento [Sousa et al. 2012]. Assim, com o objetivo de auxiliar a simulação de conceitos teóricos, jogos têm sido propostos na literatura [Bezerra e Coutinho 2013]. E assim, a utilização de jogos digitais se torna uma boa estratégia para o ensino de lógica e linguagens de programação. Além disso, neste cenário, jogos educativos de maneira geral são interessantes para auxiliar na aprendizagem devido ao fato que estes simulam situações reais de projetos de software [Bezerra et al. 2014].

Este artigo está dividido nas seguintes seções, além desta introdução: trabalhos relacionados, a metodologia aplicada, a abordagem proposta, apresentação e análise dos resultados, e por fim, as conclusões.

2. Trabalhos Relacionados

O ensino de lógica de programação e consequentemente a codificação são tarefas que requerem muito esforço por parte dos docentes [Coutinho et al. 2018]. Além disso, coletar dados sobre o desempenho dos discentes e docentes também não é trivial, pois existem muitas variáveis envolvidas, tais como metodologia, didática e tecnologias. Diversos trabalhos na literatura discutiram sobre a utilização de jogos para o ensino de programação e experiências em disciplinas de programação [Charão et al. 2016] [da Silva et al. 2016] [Raeder et al. 2016] [Barbosa et al. 2016] [Netto et al. 2017] [Coutinho et al. 2018]. Muitas situações são relacionadas a desmotivação e problemas no aprendizado.

Charão et al. (2016) propuseram um padrão pedagógico dedicado ao ensino de programação, para cursos de nível superior relacionados a Computação. Este padrão é orientado na exposição e discussão de bons e maus exemplos de códigos produzidos pelos alunos, gerando o que foi denominado respectivamente de “*Hall of Fame*” e “*Hall of Shame*”. Os resultados indicaram uma avaliação positiva do padrão pelos alunos e revelaram sua eficácia no incentivo de boas práticas de programação [Charão et al. 2016].

Barbosa et al. (2016) propuseram o PortuCol, uma pseudolinguagem de programação com instruções em português estilo Portugol, porém inspirada em C ANSI. Este trabalho também descreveu o PortuCol2C, uma ferramenta que traduz código escrito em PortuCol para C ANSI. Estudos empíricos foram conduzidos e demonstraram estatisticamente que o PortuCol é 27% mais similar a C ANSI do que o Portugol e o PortuCol é para C ANSI o que Portugol é para Pascal [Barbosa et al. 2016].

Raeder et al. (2016) relataram uma experiência diferenciada de ensino de lógica, programação e matemática para Computação, onde os sujeitos envolvidos são alunos da graduação. Formou-se um grupo de estudos denominado L2PM, que teve três professores como mediadores e ocorreu em período extraclasse. Os encontros do grupo contemplaram em um mesmo ambiente aulas integradas das disciplinas em questão, realizadas em sessões de 15 minutos, alternadas entre os professores e temas, e finalizando com uma

sessão que serve de integração, na qual se reuniram os conteúdos tratados no grupo. O resultado desta experiência apontou que a dinâmica adotada no L2PM possibilitou outras condições para a construção de conhecimento [Raeder et al. 2016].

No trabalho de da Silva et al. (2016) apresentou-se a elaboração de um jogo sério para o ensino de programação orientada a objetos, denominado POOGame. O desenvolvimento do jogo foi baseado em jogos de RPG (*Role-playing game*), de modo que o jogador controla um personagem, e a partir de batalhas é possível se invocar criaturas e controlar suas ações utilizando comandos baseados na linguagem JAVA. O jogo foi elaborado e avaliado no contexto da disciplina de Introdução ao Desenvolvimento de Jogos de um curso de Engenharia de Software, adotando uma metodologia para o desenvolvimento de jogos. Os resultados da avaliação do POOGame indicaram evidências de que o jogo contribui para o aprendizado de programação orientada a objetos [da Silva et al. 2016].

Netto et al. (2017) desenvolveram um trabalho para auxiliar na estruturação do pensamento cognitivo aliado à resistência implícita do aprendizado de algoritmos. Para isso, apresentou-se o Game Logic, um jogo voltado para o ensino-aprendizado de algoritmos por meio da programação em blocos, partindo do princípio cognitivo de que imagens são melhores memorizadas, tendo seu uso devidamente comprovado. Neste trabalho verificou-se que aspectos fundamentais para a construção de jogos educacionais devem ser seguidos, e a correta aplicação dos conceitos durante o desenvolvimento de qualquer ferramenta voltada público gera resultados positivos [Netto et al. 2017].

Gomes et al. (2018) apresentaram alguns jogos digitais utilizados no ensino de programação consistindo de uma revisão documental de trabalhos que relatam experiências de ensino de programação. A partir deste trabalho, com a descrição e comparação de algumas ferramentas para o ensino de programação com jogos, espera-se que professores possam identificar e selecionar com mais facilidade alguma ferramenta para aplicação em sala de aula [Gomes et al. 2018]. Além disso, com a descrição breve de cada jogo, também espera-se que este trabalho possa divulgar a quem se interessar por conhecer alguma ferramenta para o ensino de programação com jogos.

Segundo Medeiros e da Silva Aranha (2018), mesmo que os jogos educativos possam melhorar o desempenho dos alunos, os professores devem ter informações sobre o desempenho dos alunos, para conduzir melhor suas aulas ou prestar uma assistência personalizada, de acordo com as necessidades particulares de cada aluno. Para suportar e analisar esse desempenho, foi proposta uma abordagem estruturada como uma plataforma de suporte educacional baseado em jogos. Ao utilizar a abordagem, informações relevantes sobre o desempenho do aluno foram fornecidas, tais como erros comuns, simulação passo a passo, competências e habilidades, ajudando os professores em suas atividades acadêmicas [Medeiros e da Silva Aranha 2018].

Uma dificuldade comum é como ensinar lógica de programação de maneira que o aluno se veja motivado a aprender, ao mesmo tempo em que eles se preparam para utilizar uma ferramenta de codificação e aplicar os conceitos na prática [Coutinho et al. 2018]. O objetivo deste trabalho é relatar a experiência da utilização de uma ferramenta específica para o desenvolvimento de jogos digitais (Game Maker) para o ensino de lógica de programação em um curso de graduação. Em geral, os alunos gostaram da estratégia de utilizar o Game Maker como ferramenta, mas com algumas ressalvas.

3. Metodologia

Este trabalho propôs uma abordagem para o ensino de lógica de programação. Para isso, pesquisas iniciais em trabalhos relacionados e jogos foram realizadas, para em seguida a abordagem ser descrita. Após o fluxo de operação da abordagem ter sido definido, a ideia seria sua aplicação em sala de aula, para obtenção de *feedback* dos alunos.

Entretanto, a abordagem proposta não foi executada em sala de aula. Como forma de avaliação preliminar, e até para validar a ideia, um formulário *online* foi projetado para que professores de programação de semestres iniciais respondessem e colaborassem com a evolução e amadurecimento da abordagem.

O formulário *online* consistiu de 8 questões, sendo 2 demográficas (múltipla escolha), 3 para o *feedback* da abordagem (múltipla escolha, variando de 1 a 5, onde 1 é a pior opção e 5 é a melhor opção), e 3 para o *feedback* da abordagem em texto livre. A Tabela 1 lista todas as questões.

Tabela 1. Questões do formulário *online* (ME=Múltipla escolha; TL=Texto livre)

Tipo	Questão
ME	Quantas turmas de programação você já lecionou?
ME	Quanto tempo de experiência em anos você tem no ensino de disciplinas de programação?
ME	O que você achou da ideia do projeto (ensino de programação por meio de jogos)?
ME	Você aplicaria a abordagem no ensino de programação?
ME	Você acredita que a aplicação da abordagem trará benefícios para a aprendizagem dos alunos?
TL	Na sua opinião, quais os pontos fortes da abordagem?
TL	Na sua opinião, quais os pontos a melhorar da abordagem?
TL	Algum comentário/sugestão a mais sobre a abordagem?

4. Abordagem Proposta

Diante perspectivas de trabalhar com jogos no ensino de programação, desenvolveram-se estratégias, mediante uso dos jogos, para o ensino de lógica de programação.

Dividiu-se o ensino de programação em três grandes momentos, sendo estes: “Apresentação inicial”, “Compreensão da sintaxe” e “Aprofundamento do conteúdo”.

Pensou-se como abordagem inicial, visto que, este é o primeiro contato dos alunos com a programação, o uso do sistema Unplugged mediante uma coleção de atividades gratuitas de aprendizado que ensinam Ciência da Computação através de jogos, disponibilizadas pelo CS Education Research Group em plataforma online. Desta forma, os alunos poderiam entender os conceitos iniciais de maneira lúdica, sem o uso do computador, que por muitas vezes, pode criar um ambiente desconfortável em uma abordagem inicial.

A partir do momento em que os alunos se familiarizem com a lógica, o enfoque passaria a ser na compreensão e memorização da sintaxe. Isto será feito, com o uso do “Jogo da memória com sintaxe”. Basicamente existirão pares de cartas. Uma das cartas conterá a definição e explicação do que àquela expressão (função) executa, e na outra a sintaxe. A ideia seria formar pares assim como no jogo da memória.

No terceiro momento, quando os alunos estão minimamente habituados a linguagem e a lógica de programação, pode-se utilizar o jogo intitulado de “Quebra Cabeça de programas”. Basicamente os alunos seriam divididos em grupos. Cada grupo receberia

uma parte do código de um programa. Em um primeiro momento eles analisaram esse código e tentaram entender minimamente o que ele faz. Em seguida, seria aberta uma discussão dentro da sala, para que em conjunto conseguissem resolver o quebra cabeça e determina a função do programa.

A Figura 1 ilustra a metodologia aplicada a esse trabalho.

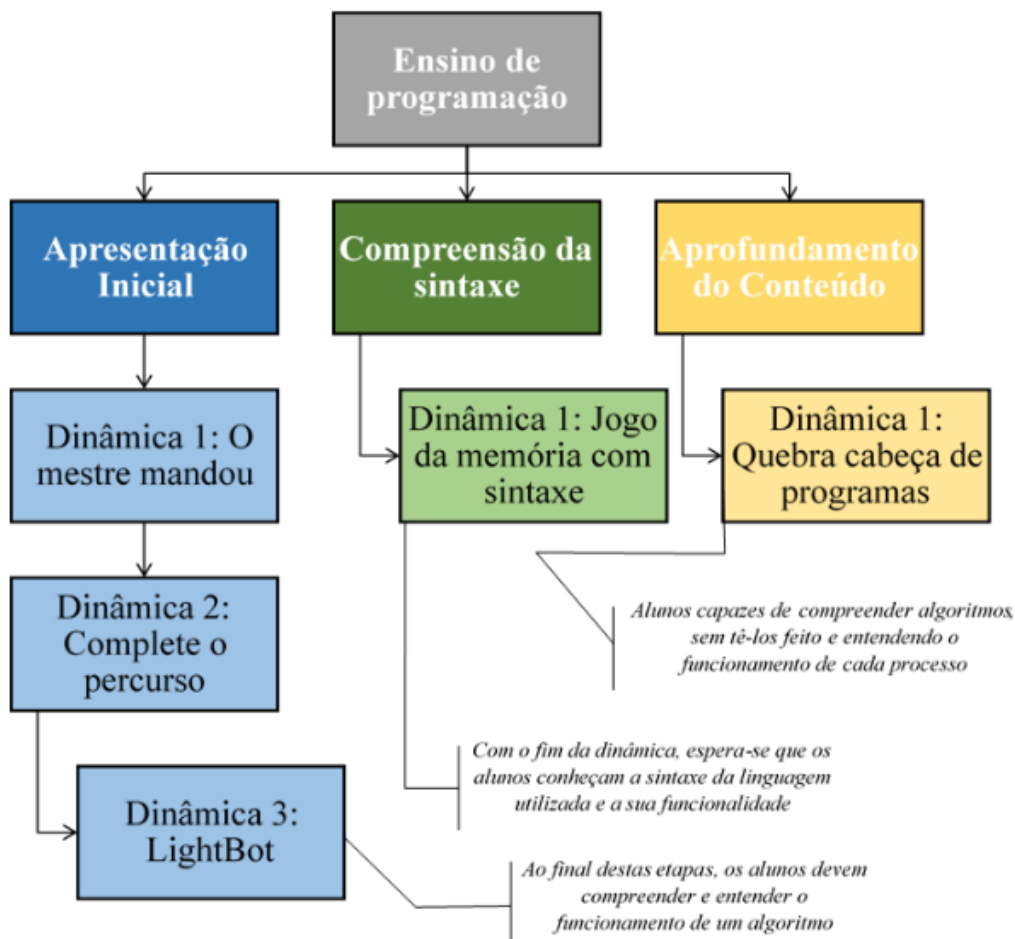


Figura 1. Diagrama da abordagem proposta

Cada atividade foi brevemente detalhada em forma de passos de aplicação.

4.1. Apresentação Inicial

DINÂMICA 01 (O MESTRE MANDOU): Um dos participantes é encarregado de ser o mestre e ficará à frente dos outros jogadores. Ele dará as ordens e todos os seguidores deverão cumprí-las. As instruções serão dadas em pseudocódigo (PORTUGOL), como em um programa, apresentando estruturas condicionais e de repetição. Exemplo: “Se você estiver de sapato, abaixe-se. Se não, dê um passo à frente”; “Se seu número de telefone termina em número ímpar, sente-se! Se não, pule até que a contagem de dez a zero acabe”.

DINÂMICA 02 (COMPLETE O PERCURSO): Para esta dinâmica, são necessários dois alunos por vez. O objetivo é completar um percurso no chão. Um dos alunos estará

vendado, o outro deverá ter toda a visão do percurso, de forma a instruir o colega do que ele deve fazer. O “aluno guia”, só poderá dizer as ações que o companheiro deverá realizar, no início do percurso. O “aluno conduzido” deverá memorizar tudo. Caso saia do percurso, uma nova equipe deverá ser escolhida.

DINÂMICA 03 (LIGHTBOT): O LightBot² é um jogo de plataforma digital. A ideia é, com a ajuda de um projetor, fazer os alunos jogarem, as fases iniciais. O objetivo do Lightbot é comandar um pequeno robô para navegar em um labirinto e acender lâmpadas. Os jogadores organizam símbolos na tela para comandar o robô a andar, girar, pular, acender uma lâmpada e assim por diante. O labirinto e a lista de símbolos se tornam mais complicados à medida que as lições progridem. Ao usar esses comandos, os jogadores aprendem conceitos de programação como *loops*, procedimentos e muito mais, sem inserir código em nenhuma linguagem de programação.

OBSERVAÇÕES IMPORTANTES: Ao final de cada dinâmica é importante comentar com os alunos sobre a relação da dinâmica com a programação.

4.2. Compreensão da Sintaxe

DINÂMICA 01 (JOGO DA MEMÓRIA COM SINTAXE): Neste jogo, formaram-se grupos de até 3 jogadores. O objetivo é, como em um jogo da memória, fazer o maior número de pares possíveis. Um par é a combinação de duas cartas: na primeira está escrita a definição de uma função; na outra, encontra-se a expressão em linguagem de programação. Exemplo: Carta 01 – Usada para estabelecer estruturas condicionais; Carta 02 – IF/ELSE.

4.3. Aprofundamento do Conteúdo

DINÂMICA 01 (QUEBRA CABEÇA DE PROGRAMAS): Nesta dinâmica, um programa bem estruturado em alguma linguagem de programação seria fragmentado em blocos (quantos forem necessários). A turma será dividida em grupos e cada grupo receberá uma parte do programa. As equipes individualmente devem tentar entender o que sua parte do programa realiza. Depois de 7 minutos, um grupo por vez deverá apresentar aos demais o que a sua parte do quebra cabeça realiza. Ao final os grupos devem discutir, para tentar entender do que se trata o programa.

DINÂMICA 01 (JOGOS COMPLEMENTARES): Tinycards³ é um aplicativo de “flash cards”, cartões de memória disponível para facilitar o aprendizado de palavras ou conceitos. O próprio usuário pode editar os cartões. Desta forma torna-se uma ótima ferramenta de aprendizagem, que pode auxiliar na compreensão da sintaxe.

5. Resultados

Nesta seção os resultados das respostas do questionário, preenchidos pelos professores, serão apresentados. Nesse caso, apenas 6 professores responderam à pesquisa, todos com experiência em linguagens de programação.

²<http://lightbot.com/>

³<https://tinycards.duolingo.com/>

5.1. Questões de Múltipla Escolha

A primeira questão apresentou a quantidade de turmas de programação que os professores já tinham lecionado. Nesse caso fica uma relação direta com a experiência do professor no ensino, em termos de quantidade de turmas. A Figura 2 ilustra esse quantitativo, sendo que metade dos professores informou que já ministrou mais de 10 turmas, o que é bom.

Quantas turmas de programação você já lecionou?

6 respostas

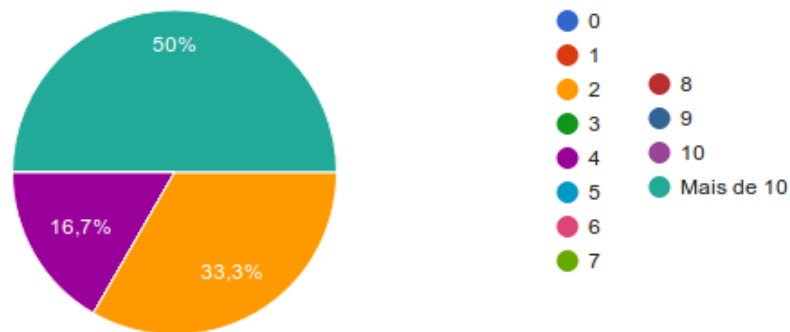


Figura 2. Quantas turmas de programação você já lecionou?

A segunda questão foi sobre a experiência do professor em anos. Novamente um ponto forte foi que metade dos professores informou que possuía 10 ou mais anos de experiência. Essa questão juntamente com a anterior indicam que os professores podem colaborar com o refinamento da abordagem proposta. A Figura 3 apresenta a experiência dos professores em anos.

Quanto tempo de experiência em anos você tem no ensino de disciplinas de programação?

6 respostas

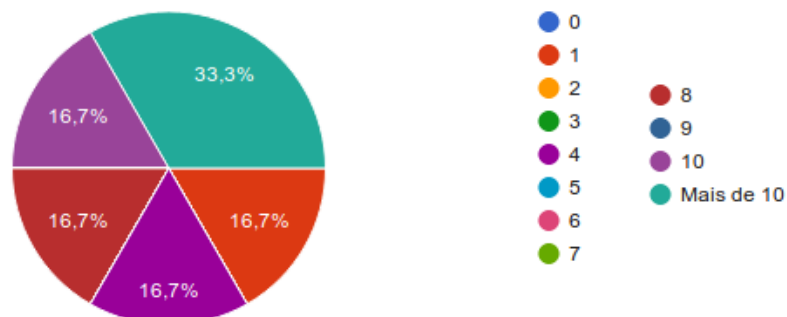


Figura 3. Quanto tempo de experiência em anos você tem no ensino de disciplinas de programação?

A Figura 4 exibe que nenhum professor achou ruim a abordagem, e a maioria achou muito boa, apesar que apenas 6 professores responderam ao formulário. O ideal é ter a opinião de mais professores, e que esses professores tenham aplicado a abordagem em aula, o que não foi possível de se fazer no semestre da pesquisa. Resta saber se ao aplicar na prática em sala de aula se esses resultados ainda se manteriam positivos.

O que você achou da ideia do projeto (ensino de programação por meio de jogos)?

6 respostas

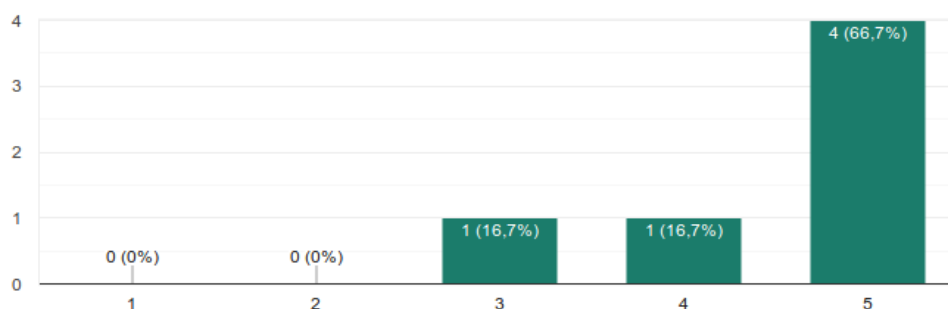


Figura 4. O que você achou da ideia do projeto (ensino de programação por meio de jogos)?

Quanto à aplicabilidade em sala, a maioria optou por ser neutro ou aplicaria fortemente a abordagem. Talvez por necessidade de mais detalhes na abordagem ou pelos professores não se interessarem por jogos no ensino de programação tenha levado a essa pontuação neutra. A Figura 5 ilustra esse resultado.

Você aplicaria a abordagem no ensino de programação?

6 respostas

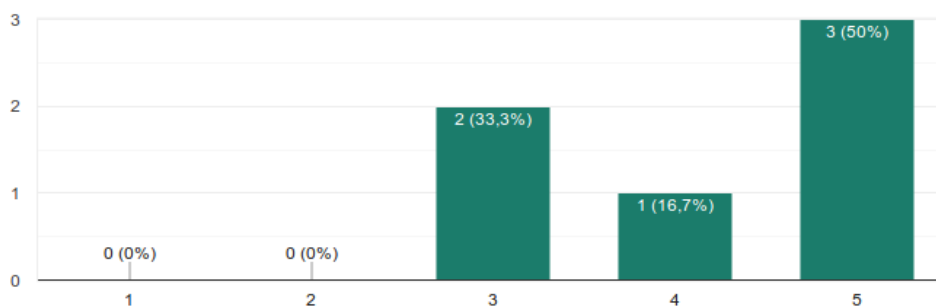


Figura 5. Você aplicaria a abordagem no ensino de programação?

Por fim, quanto aos benefícios da aplicação da abordagem, a maioria apontou que acredita ser possível, destacado na Figura 6. Mas novamente, para se comprovar esses benefícios seria necessário a aplicação da abordagem.

Você acredita que a aplicação da abordagem trará benefícios para a aprendizagem dos alunos?

6 respostas

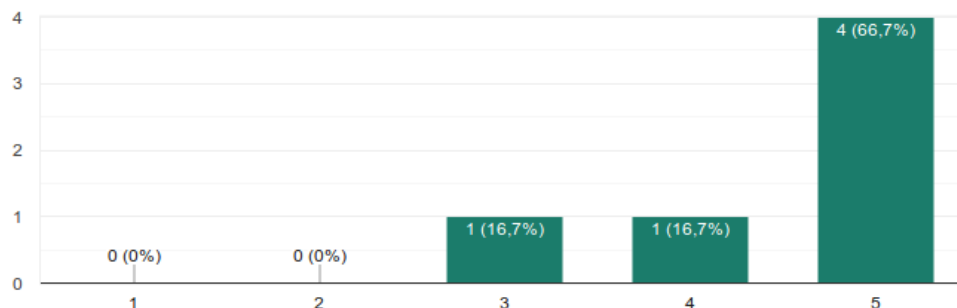


Figura 6. Você acredita que a aplicação da abordagem trará benefícios para a aprendizagem dos alunos?

5.2. Questões Livres

Três questões de texto livre foram planejadas para a obtenção da opinião sobre pontos fortes, de melhoria e sugestões sobre a abordagem.

5.2.1. Na sua opinião, quais os pontos fortes da abordagem?

Como pontos fortes destacados nos comentários dos professores surgiu o termo “foco”, que é um desafio a ser lidado pelos docentes, destacado em “*Faz com que o aluno tenha um maior foco na disciplina de lógica de programação*”. Outro aspecto foi quanto ao entendimento e o apelo por ser um conjunto de jogos, o que atrai alunos, ressaltado em “*Por ser algo lúdico, para iniciantes que nunca viram nada de programação pode auxiliar*”. A associação com sintaxe também se destacou, pois é um problema constante: “*Particularmente a associação de sintaxe me parece atrativa, pois é uma dificuldade constante*”.

5.2.2. Na sua opinião, quais os pontos a melhorar da abordagem?

Como ponto para melhoria, sugeriu-se um maior formalismo à proposta, como se dá a entender pelo comentário “*Algumas abordagens de ensino de programação de jogos não focam o suficiente no formalismo da linguagem ou da lógica de programação*”. Também é interessante se exibir ou avaliar o aprendizado do aluno ao executar a abordagem: “*Incluir métodos para atestar o grau de aprendizado dos alunos*”. Por fim, balancear atividades sem e com programação ou codificação: “*Não sou muito fã de muitas atividades de programação sem programação, mas acho que as atividades de brincadeiras poderiam ser mais direcionadas*”.

5.2.3. Algum comentário/sugestão a mais sobre a abordagem?

Como sugestão, existiram comentários para utilização de ferramentas para facilitar o entendimento dos diversos conceitos relacionados à programação, como destacado em “*Seria interessante utilizar ferramentas que facilitassem o aprendizado de conceitos mais abstratos de programação*”. Além disso, para uma maior prática, sugeriu-se mais atividades, como relatado em “*Acho que apenas a inclusão de mais atividades, com mais tempo para fixação e adaptação dos conceitos*”. E ainda em relação às atividades, estas poderiam ter mais detalhes, facilitando a compreensão: “*As atividades poderiam ter mais detalhes. Uma imagem / fluxo de atividades que representasse cada uma das atividades seria bastante útil*”.

6. Conclusão

Este trabalho apresentou uma abordagem para o ensino de programação baseada em alguns princípios de jogos, e alguns momentos de trabalho, sendo estes: “Apresentação inicial”, “Compreensão da sintaxe” e “Aprofundamento do conteúdo”. Utilizou-se também o sistema Unplugged mediante uma coleção de atividades gratuitas de aprendizado que ensinam programação e codificação por meio de jogos.

Apesar da proposta, a abordagem não foi aplicada em sala de aula, consistindo em um trabalho futuro. Entretanto, uma pesquisa com professores de programação de semestres iniciais opinaram sobre a abordagem.

Percebeu-se com a realização deste projeto, que é extremamente possível, por meio de ideias simples, melhorar o aprendizado dos alunos no que tange a lógica e a programação. Como perspectiva futura, objetiva-se a aplicação destas práticas em sala de aula e a verificação, mediante coleta de dados, do real ganho da introdução desta metodologia para a aprendizagem.

Agradecimentos

Este trabalho foi realizado com recursos do projeto **Aprendendo Lógica de Programação por Meio de Jogos Digitais**, Bolsa de Iniciação Acadêmica/PRAE/2018, da Universidade Federal do Ceará (UFC).

Referências

- Barbosa, L. L., Couto, C. M. S., e Terra, R. (2016). Portucol: uma pseudolinguagem inspirada em c++ para o ensino de lógica de programação e algoritmos. In *XXIV Workshop sobre Educação em Computação (WEI2016) - XXXVI Congresso da Sociedade Brasileira de Computação*.
- Bezerra, C. I. M. e Coutinho, E. (2013). Avaliação do jogo itestlearning: Um jogo para o ensino de planejamento de testes de software. In *XXI Workshop sobre Educação em Computação (WEI2013)*.
- Bezerra, C. I. M., Coutinho, E. F., Santos, I. S., Monteiro, J. M., e Andrade, R. M. C. (2014). Evolução do jogo itestlearning para o ensino de testes de software: Do planejamento ao projeto. In *XIX Conferência Internacional sobre Informática na Educação (TISE2014)*, Fortaleza.

- Charão, A. S., Neto, A. F. K., de O. Stein, B., e de A. Barcelos, P. P. (2016). Hall of fame/shame: um padrão pedagógico para o ensino de programação. In *XXIV Workshop sobre Educação em Computação (WEI2016) - XXXVI Congresso da Sociedade Brasileira de Computação*.
- Coutinho, E. F., Bonates, M. F., e Moreira, L. O. (2018). Relato sobre o uso de uma ferramenta de desenvolvimento de jogos para o ensino introdutório de lógica de programação. In *Anais dos Workshops do VII Congresso Brasileiro de Informática na Educação (WCBIE 2018) - IV Workshop de Ensino em Pensamento Computacional, Algoritmos e Programação (WAlgProg 2018)*.
- Coutinho, E. F., de Lima, E. T., e Santos, C. C. (2017). Um panorama sobre o desempenho de uma disciplina inicial de programação em um curso de graduação. *Revista Tecnologias na Educação*, 19(9).
- Coutinho, E. F., Gomes, G. A. M., e José, M. L. A. (2016). Applying design thinking in disciplines of systems development. In *2016 8th Euro American Conference on Telematics and Information Systems (EATIS)*, pages 1–8.
- CSUnplugged (2015). The book - computer science unplugged. <https://classic.csunplugged.org/books/>. Acessado em novembro, 2018.
- CSUnplugged (2018). Cs unplugged. <https://csunplugged.org/>. Acessado em novembro, 2018.
- da Silva, L. M. P., Bonfim, B. C., Silva, R. C., da Silva, J. B., nad Carla I. M. Bezerra, W. L. M., e Jucá, P. M. (2016). Poogame: Um jogo sério para o ensino de programação orientada a objetos. In *XXIV Workshop sobre Educação em Computação (WEI2016) - XXXVI Congresso da Sociedade Brasileira de Computação*.
- de Souza, C. M. (2009). Visualg - ferramenta de apoio ao ensino de programação. *Revista Teccen*, 2(2).
- Ferreira, A. C., Santos, J., Silva, R., Oliveira, A. T. R., Zabet, D., Abdalla, D., e Matos, E. (2016). Hello world: relato de experiência de um curso de iniciação à programação. In *II Workshop de Ensino em Pensamento Computacional, Algoritmos e Programação (WAlgProg)*, pages 1306–1315. SBC.
- Gomes, H. A. C., Melo, J. M., Farrapo, H. P., Bonates, M. F., e Coutinho, E. F. (2018). Descrição e comparação de jogos digitais para auxiliar no ensino de programação. *Revista Sistemas e Mídias Digitais (RSMD)*, 3(1).
- Marinho, C. S. S., Moreira, L. O., Coutinho, E. F., Paillard, G. A. L., e Neto, E. T. L. (2016). Experiências no uso da metodologia coding dojo nas disciplinas básicas de programação de computadores em um curso interdisciplinar do ensino superior. In *II Workshop de Ensino em Pensamento Computacional, Algoritmos e Programação (WAlgProg)*, pages 1097–1106.
- Medeiros, H. B. e da Silva Aranha, E. H. (2018). Apoiando a avaliação do desempenho dos alunos com base em jogos educacionais. In *XXIV Workshop de Informática na Escola (WIE 2018)*.
- Netto, D., Medeiros, L. M., de Pontes, D., e de Moraes, E. (2017). Game logic: Um jogo para auxiliar na aprendizagem de lógica de programação. In *XXV Workshop sobre*

Educação em Computação (WEI2017) - XXXVII Congresso da Sociedade Brasileira de Computação.

- Raeder, M., Py, M., Rigo, S., e Pinheiro, J. (2016). L2pm: relato de uma experiência sobre o ensino integrado de lógica, programação e matemática para computação. In *XXIV Workshop sobre Educação em Computação (WEI2016) - XXXVI Congresso da Sociedade Brasileira de Computação.*
- Rapkiewicz, C. E., Falkembach, G., Seixas, L., dos Santos Rosa, N., da Cunha, V. V., e Klemann, M. (2006). Estratégias pedagógicas no ensino de algoritmos e programação associadas ao uso de jogos educacionais. *RENOTE - Novas Tecnologias na Educação*, 4(2).
- Ribas, E., Bianco, G. D., e Lahm, R. A. (2016). Um curso de programação a distância com metodologias ativas e análise de aprendizagem por métricas de software. *RENOTE - Revista Novas Tecnologias na Educação*, 14(2).
- Roland, L. C., Fabre, M.-C. J. M., Konrath, M. L. P., e Tarouco, L. M. R. (2004). Jogos educacionais. *RENOTE - Novas Tecnologias na Educação*, 2(1).
- Sousa, V., Bezerra, C. I. M., Coutinho, E., e Santos, I. S. (2012). itest learning: Um jogo para o ensino do planejamento de testes de software. In *V Fórum de Educação em Engenharia de Software (FEES 2012)*, Natal - RN.