

Um modelo para promover o engajamento estudantil no aprendizado de programação utilizando *gamification*

Tatyane Souza Calixto da Silva¹, Jeane Cecília Bezerra de Melo², Patricia Cabral de Azevedo Restelli Tedesco¹

¹Centro de Informática – Universidade Federal de Pernambuco (UFPE)
Av. Prof. Moraes Rego, 1235 - Cidade Universitária Brazil - CEP: 50670-901 - Recife - PE Brasil

²Departamento de Estatística e Informática – Universidade Federal de Rural Pernambuco (UFPE)
Rua D. Manoel de Medeiros, s/n - Dois Irmãos - CEP: 52171-900 - Recife - PE - Brasil
tscs@cin.ufpe.br, jeane.ufrpe@gmail.com, pcart@cin.ufpe.br

Resumo. *O aprendizado de programação apresenta desafios e dificuldades em diferentes níveis, podendo ocasionar baixo desempenho e desmotivação no estudante, aumentando o índice de evasão em cursos de Computação. A pesquisa nesta área é vasta, e apresenta diversas formas de abordar este problema, como utilização de jogos, gamification, entre outras. Neste estudo, utilizamos o engajamento estudantil, concomitantemente com as técnicas de gamification, como forma de minimizar os problemas acima citados. Os resultados da pesquisa apontam que a utilização do modelo proposto nesse estudo propiciou um maior engajamento dos estudantes, sendo mensurado através dos seus indicadores.*

1. Introdução

O Currículo de Referência da Sociedade Brasileira de Computação aponta que a disciplina de programação é essencial para formação sólida dos discentes de diversos cursos de Computação (SBC, 1999). Embora a disciplina de programação seja considerada fundamental na formação acadêmica dos discentes, estudos indicam que o seu aprendizado não é trivial (GIRAFFA; MULLER; MORAES, 2015). As dificuldades enfrentadas pelos alunos no aprendizado de programação citadas com maior frequência na literatura são: baixa compreensão das noções básicas, como estrutura de controle; criação de algoritmos que resolvam problemas concretos; aprender a pensar algorítmicamente; uma base matemática fraca; uso de linguagens de programação com sintaxes rebuscadas para estudantes em início de curso; o não entendimento dos enunciados; exercícios dissociados de problemas “reais” e pouca disponibilidade extraclasse para estudos (YANG *et al*, 2014; GOMES; MENDES, 2014).

Essas dificuldades causam inevitavelmente elevadas taxas de insucesso ou desistência dos estudantes no curso de Computação (GOMES, 2010; GOMES; MENDES, 2014). Segundo o INEP (2012) o índice de evasão dos cursos das áreas exatas é de 28%. Esse valor é considerado alto, uma vez que a média nacional é de 23%. Ao analisar a área de Computação, observa-se que a disciplina de programação é apontada como responsável, ou contribui de maneira efetiva, para evasão e reprovação nos primeiros períodos do curso (GIRAFFA, 2013).

Uma das implicações desse panorama é a desmotivação do estudante com a área de programação, refletindo no atraso do curso (BRITO; MADEIRA, 2015). Diante dessa situação, é necessário buscar maneiras de propor soluções que possam amenizar e remediar essa condição de forma efetiva (BRITO; MADEIRA, 2015).

Uma das formas utilizadas com o intuito de motivar, engajar e influenciar positivamente na mudança do comportamento de um público definido por intermédio da utilização de elementos de jogos é conhecida como *gamification* (DETERDING *et al.*, 2011). *Gamification* pode ser definido como a utilização de várias técnicas e elementos de *games* em contextos não orientados a jogos, com o intuito de motivar e engajar indivíduos a resolver vários problemas (BRAZIL; BARUQUE, 2015).

Com o intuito de tornar o processo de aprendizagem mais dinâmico, pode-se buscar personalizar o aprendizado, tornando-o mais eficiente. Um conceito que pode contribuir para isto é o Contexto Computacional, definido como qualquer informação que caracteriza a situação de uma entidade (pessoa, lugar ou objeto) considerado relevante para a interação entre uma pessoa e uma aplicação. Sistemas Sensíveis ao Contexto são aqueles onde há gerenciamento de informações contextuais relacionados a uma aplicação com o intuito de apoiar um agente na execução de uma atividade (VIEIRA; TEDESCO; SALGADO, 2009).

Diante desse cenário, esta pesquisa visa investigar a influência positiva de *gamification* no engajamento e desempenho dos estudantes no aprendizado de programação. Para tanto, um modelo conceitual foi definido, o qual utiliza *gamification*, baseando-se no perfil de jogadores e técnicas de engajamento para proporcionar um aprendizado personalizado e promover o engajamento estudantil.

Este artigo está organizado como segue: na seção 2 são apresentados os trabalhos relacionados. A seção 3 aborda o modelo proposto. O estudo experimental que utilizou o modelo proposto é apresentado na seção 4. A seção 5 apresenta a discussão do estudo. Por fim, na seção 6 são feitas as considerações finais da pesquisa, destacando as contribuições do estudo e apontando algumas direções para trabalhos futuros.

2. Trabalhos Relacionados

Codecademy é uma plataforma *e-learning* para o aprendizado de programação, projetada com *gamification*. Nela, os cursos de programação são organizados em seções, constituídas por sequência de exercícios, que possuem um texto educativo para introduzir o tema, instruções que indicam ao estudante o que fazer e o exercício interativo real para conclusão dessa atividade. Os alunos são recompensados com *badges* quando atingem determinado número de pontos, ou de certas aulas ou cursos. Nessa plataforma o professor não tem a possibilidade de acompanhar o progresso do estudante através da plataforma. O sistema não é sensível ao contexto e não se adequa a interação do usuário com o sistema, para auxiliar no aprendizado do estudante.

O *Codeschool* é uma plataforma *online* para o ensino de programação. Os cursos são em forma de vídeos seguidos de exercícios, chamados de desafios. Utiliza elementos de jogos como pontuação para cada desafio cumprido e *badges* quando o usuário passa de um nível a outro. O *Codeschool* não possibilita o acompanhamento do professor ao estudante. Como também, não apresenta a possibilidade de personalizar o aprendizado levando em consideração o perfil de jogador do aluno.

3. Modelo Conceitual cod[edu]

O modelo conceitual aqui proposto, denominado cod[edu], tem o intuito de promover o engajamento estudantil e auxiliar no desenvolvimento das habilidades e competências necessárias para a aprendizagem de programação (Figura 1).

O cod[edu] é dividido em três partes: *perfil de jogadores*, as *técnicas de engajamento* e *elementos de games*. Os dois primeiros visam promover e manter o engajamento dos estudantes no ambiente de maneira contextualizada, sendo as técnicas de engajamento baseadas no perfil do jogador. No que se refere aos elementos de games, esses possuem o objetivo de proporcionar que o sistema possibilite desafios e promova diversão.

No modelo, as técnicas de engajamento: *Desafios* e *Missões*, *Ranking e Leaderboards*, *Interação Social*, *Achievements ou Badges*, dependem do perfil do jogador, escolhidas de modo a serem relevantes e favorecerem o engajamento dos estudantes, tornando a experiência de aprendizado mais significativa. As recomendações do sistema serão baseadas e adaptadas ao estilo de cada jogador.

No que se refere aos elementos de *games* o modelo contempla: pontuação, narrativa e *feedback* que se relaciona com todas as técnicas de engajamento. Em relação à competição, esta é promovida através da técnica de *ranking* e *leaderboard*, comumente utilizadas para promover a competição dentro do sistema. Por fim, as recompensas são associadas aos *Achievement* ou *badges* usados com intuito de premiar cada tarefa finalizada.

O modelo conceitual contempla a Taxonomia de Bloom ou taxonomia dos objetivos educacionais proposta por Bloom, Engelhart e Furst (1956), com o intuito de auxiliar no planejamento, organização e controle dos objetivos de aprendizagem. Sendo composta por três domínios: Cognitivo, Afetivo e Psicomotor. Para esse estudo, utilizaremos apenas o domínio cognitivo, pois se relaciona com o aprender e dominar um conhecimento. Envolve a aquisição de um novo conhecimento, do desenvolvimento intelectual, de habilidade e de atitudes. A taxonomia de Bloom é muito utilizada em várias áreas do conhecimento, inclusive na Computação. Pesquisas comprovam a aplicabilidade dessa taxonomia na elaboração de instrumentos de avaliação na disciplina de programação (WHALLEY et al, 2006).

No contexto do modelo conceitual, a taxonomia de Bloom foi associada às técnicas de engajamento, para avaliar os objetivos educacionais, desempenho e envolvimento dos estudantes durante o seu processo de aprendizagem de programação, usando para isso *gamification*. A seguir serão apresentadas as categorias da taxonomia de Bloom que o modelo conceitual contempla:

- i. **Lembrar:** essa categoria requer que o aluno resgate informações e/ou conhecimento da memória (SCOTT, 2003). No modelo conceitual ela está relacionada com *Achievement* ou *badges* que são utilizados como recompensas, com o intuito de promover o engajamento discente e a aprendizagem, uma vez que é exigido do estudante o conhecimento para receber as recompensas por intermédio de exercícios em forma de *quiz*.
- ii. **Entender:** categoria que aborda a capacidade do estudante em dar um significado e interpretar o conhecimento adquirido através das experiências educacionais (KRATHWOHL, 2002). No contexto do modelo conceitual, essa

categoria está relacionada com todas as técnicas de engajamento, que fornecem um *feedback* rápido, claro e relevante das interações dos estudantes, possibilitando ao estudante entender como um conceito é aplicado de diferentes formas.

- iii. **Aplicar:** categoria relacionada à execução de um procedimento numa situação específica ou em uma situação nova (JESUS; RAABE, 2009). Essa categoria, no âmbito do modelo conceitual está relacionada com os desafios e missões, pois é através dessas técnicas de engajamento que o estudante tem a possibilidade de aplicar o seu conhecimento.
- iv. **Analisar:** essa categoria contempla a capacidade do estudante de diferenciar, a informação relevante da irrelevante, bem como compreender a inter-relação existente entre as partes (JESUS; RAABE, 2009). No modelo conceitual, essa categoria está relacionada com desafios e missões, uma vez que possibilitam a resolução de problemas e enigmas permitindo a escolha de elementos distintos.
- v. **Avaliar:** essa categoria aborda a capacidade que o estudante tem de julgar critérios e padrões quantitativos e/ou qualitativos (SCOTT, 2003). No modelo conceitual, essa categoria está relacionada com *Ranking* e *Leaderboard*, uma vez que são técnicas de engajamento que são modificadas pela interação do jogador mostrando seu desempenho.

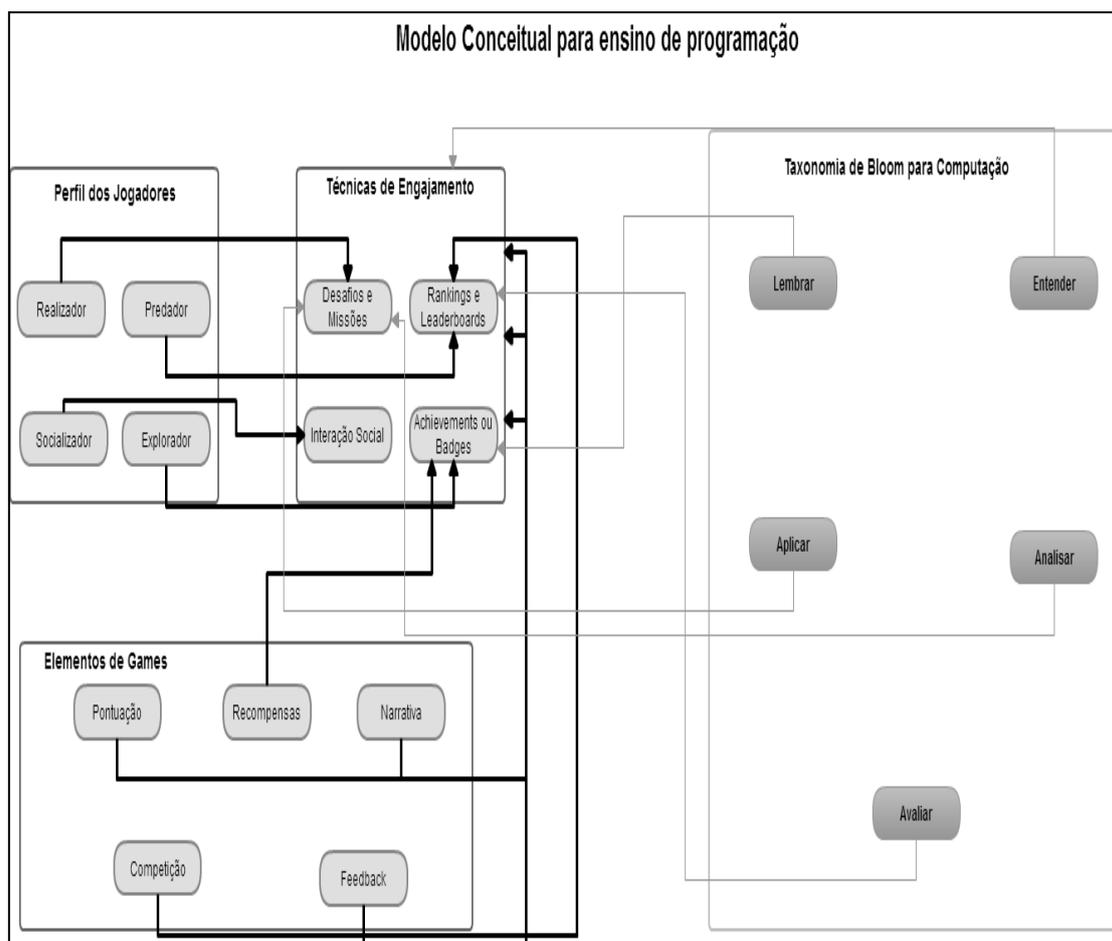


Figura 1: Modelo Conceitual para ensino de programação

3.1 Ambiente cod[edu]

O cod[edu] é um ambiente de aprendizagem *online* cuja a finalidade é auxiliar o docente no processo de acompanhamento do estudante. O ambiente visa promover o engajamento do estudante e auxiliar o seu aprendizado em programação, através de exercícios resolvidos com êxito ou não, apresentados em forma de *quiz*. Essa ferramenta foi desenvolvida baseando-se no modelo conceitual.

O sistema cod[edu] utiliza Recomendação baseada em Conteúdo, uma vez que usa as interações do usuário, para classificar seu perfil de jogador e lançar técnicas de engajamento, baseado em seu perfil. Uma das principais vantagens desta técnica é que ela não necessita da avaliação prévia de um item para gerar a recomendação, ainda possibilita a chance de todos os itens serem recomendados, uma vez que dependem unicamente do perfil do usuário.

Buscando validar o modelo conceitual proposto, um estudo foi realizado com dois professores e com quatro turmas. Os resultados deste estudo são descritos na próxima seção.

4. Estudo Experimental

Objetivando identificar a contribuição do modelo cod[edu] na aprendizagem de programação e a influência no engajamento estudantil, um quase-experimento (SHADISH et al., 2002) foi realizado de 19 de setembro a 30 de Novembro de 2015.

O presente estudo foi realizado com quatro turmas, sendo três turmas do segundo, terceiro e sétimo períodos da FACOL (Faculdade Escritor Osman da Costa Lins) nas disciplinas de Linguagem de Definição de Dados XML, Programação Orientada a Objetos e PHP respectivamente, do curso de Sistemas de Informação, ministradas pelo mesmo professor da referida faculdade. Além destas, uma turma da disciplina de Programação Orientada a Objeto da UFRPE (Universidade Federal Rural de Pernambuco) do terceiro período do curso de Licenciatura em Computação.

A população do grupo experimental foi composta por vinte e quatro alunos, no qual 7 estudantes foram da disciplina de PHP, 14 de Linguagem de Definição de Dados XML da FACOL e 3 da disciplina de Programação Orientada a Objeto da UFRPE.

O grupo de controle foi formado por sete alunos da disciplina de Programação Orientada a Objetos da FACOL. A atribuição dos participantes do grupo de controle e experimental foi realizada de forma não aleatória.

4.1 Coleta de Dados e Execução

Nesta etapa, os dados são reunidos para serem analisados e avaliados posteriormente e as fases da execução da pesquisa são descritas. O primeiro estágio da coleta de dados contemplou a aplicação de um questionário com intuito de conhecer e mensurar os indicadores de engajamento dos estudantes antes da utilização da ferramenta cod[edu].

O questionário dos indicadores de engajamento estudantil possui nove questões. Cada pergunta aborda um indicador, sendo distribuídos da seguinte forma: (Q1) *Autonomia*, (Q2) *Execução*, (Q3) *Social*, (Q4) *Entrega*, (Q5) *Participação*, (Q6) *Colaboração*, (Q7) *Cooperação*, (Q8) *Questionamento* e (Q9) *Diversão* (SEIXAS; GOMES; MELO, 2015). Para responder as perguntas, os alunos utilizaram a escala de

Likert com os seguintes itens: Discordo Totalmente (DT), Discordo parcialmente (DP), Indiferente (I), Concordo parcialmente (CP) e Concordo totalmente (CT) (Likert, 1932). O resultado do questionário dos indicadores de engajamento pode ser visualizado na Figura 2 e Figura 3.

Para a confiabilidade do questionário foi utilizado o coeficiente de *Alfa* de Cronbach (CRONBACH, 1951). O cálculo de *Alfa* de Cronbach do questionário pré-teste dos indicadores de engajamento resultou em 0,72, podendo ser classificado como consistente.

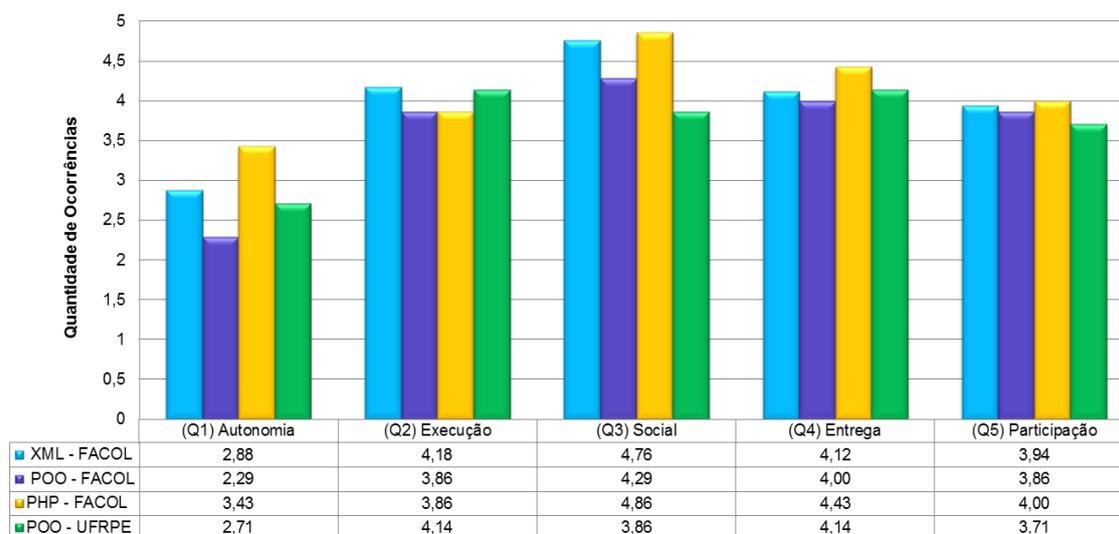


Figura 2. Indicadores de engajamento de todas as turmas.

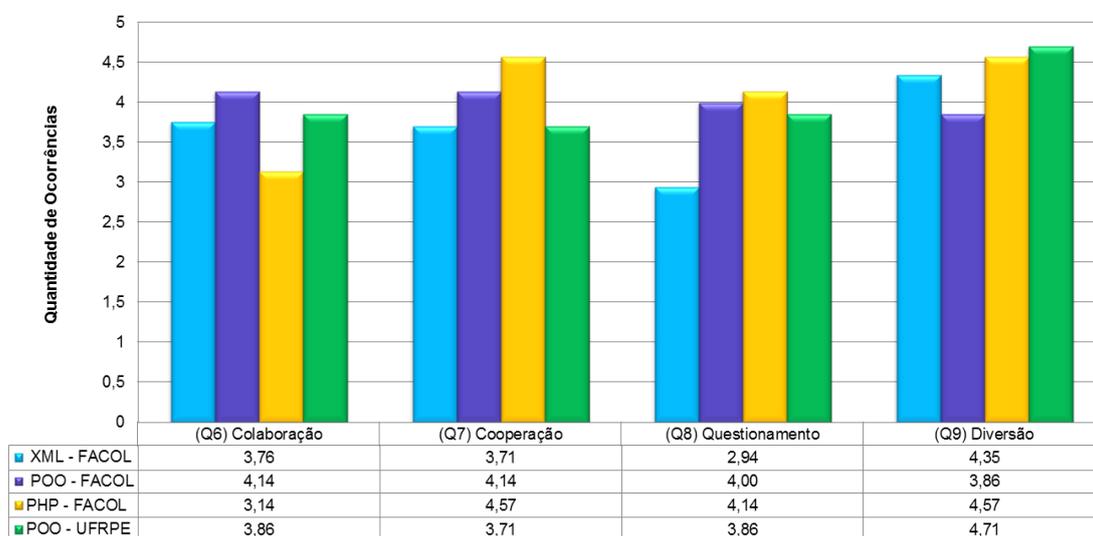


Figura 3 Indicadores de engajamento de todas as turmas.

Uma vez realizada a intervenção com a ferramenta e com a finalidade de saber se os indicadores de engajamento dos alunos foram alterados após a utilização do cod[edu], foi aplicado um questionário que abordou os mesmos elementos do questionário utilizado no pré-teste, no qual os estudantes responderam usando também a escala de Likert (LIKERT, 1932). Para esse questionário também foi realizado o cálculo de *Alfa* de Cronbach (CRONBACH, 1951) que obteve o resultado de 0,86, podendo ser

classificado como consistente. O resultado do pós-teste dos indicadores de engajamento podem ser visualizados nas Figuras 4, 5, 6 e 7.

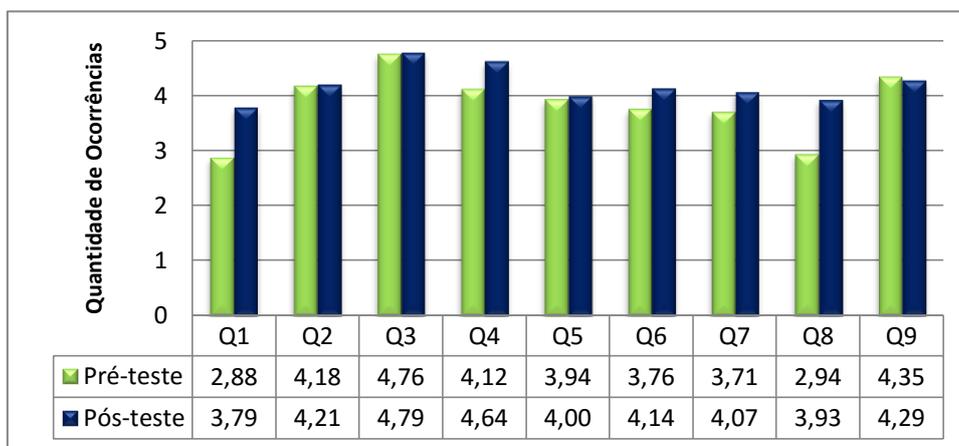


Figura 4. Valor médio das Respostas da Turma de XML – FACOL – Indicadores de Engajamento

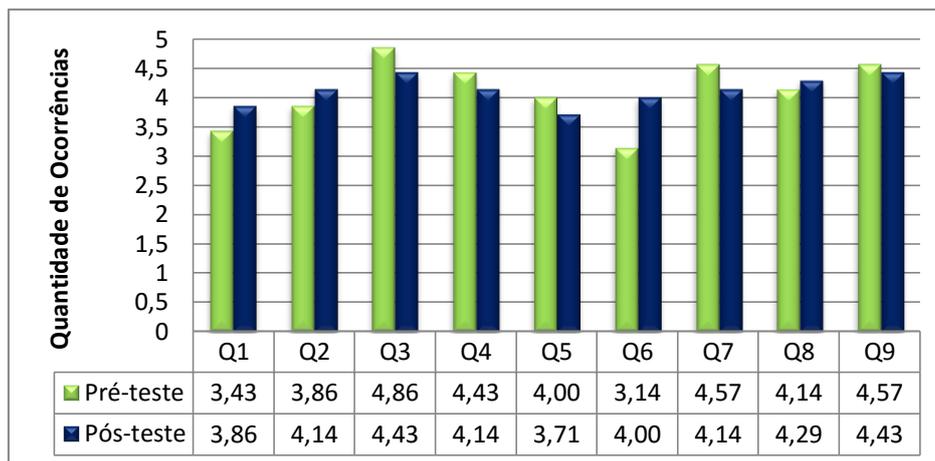


Figura 5 .Valor Médio das Respostas da Turma de PHP - FACOL - Indicadores de Engajamento

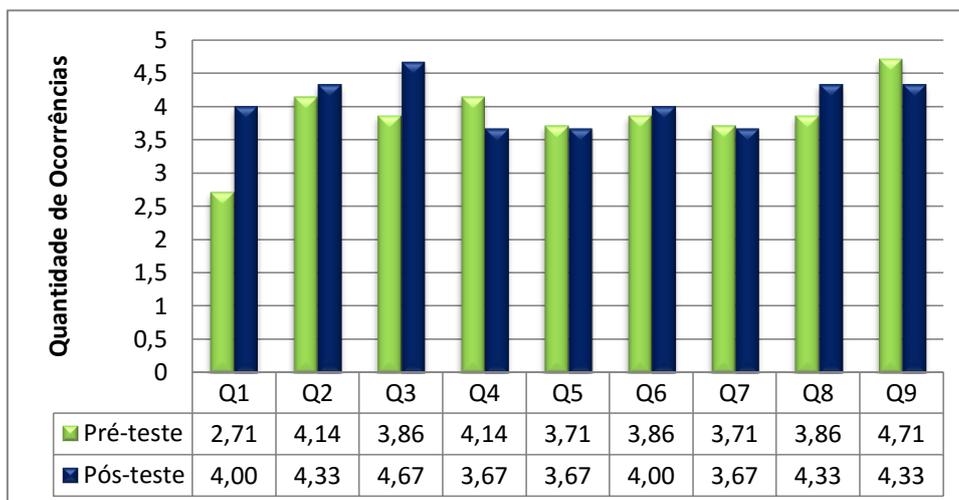


Figura 6.Valor Médio das Respostas da Turma de Programação - UFRPE - Indicadores de Engajamento

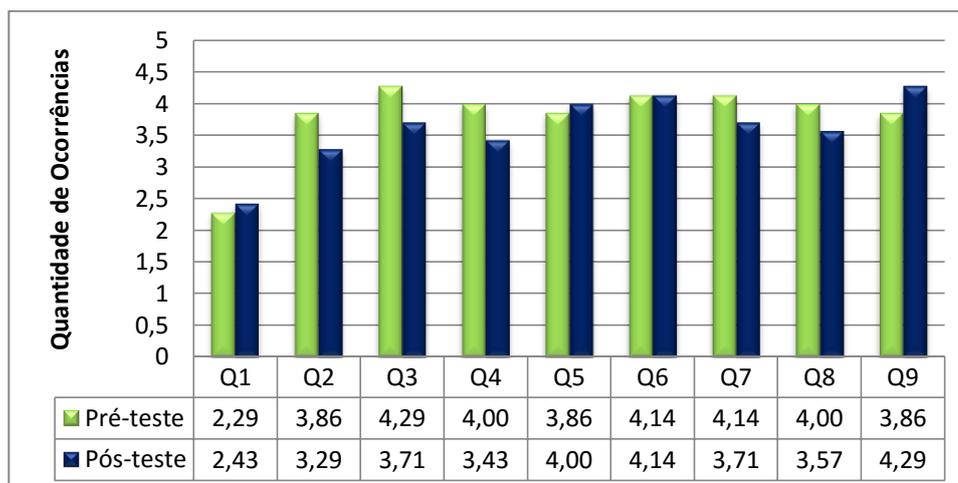


Figura 7. Valor Médio das Respostas da Turma de Programação - FACOL- Indicadores de Engajamento

5. Discussão

O experimento realizado foi conduzido por uma questão de pesquisa que pretendeu constatar a influência positiva da utilização do cod[edu] no engajamento e desempenho dos estudantes no processo de aprendizagem de programação, e obteve resultados significativos.

No que se refere aos indicadores de engajamento apresentaram mudanças positivas após a intervenção didática com o modelo. Tal resultado pode estar relacionado com a utilização de *gamification* no cod[edu] tendo como finalidade a promoção do engajamento de estudantes nas atividades.

No que concerne à avaliação do modelo conceitual, no qual a ferramenta cod[edu] foi baseada, observou-se que os estudantes mais engajados no sistema eram aqueles que tiveram o perfil de jogador classificados pelo cod[edu]. Com isso, podemos inferir que a personalização das técnicas de engajamento, baseadas no perfil de jogador e adicionadas no sistema através dos elementos de jogos, influenciaram de forma positiva o engajamento dos discentes.

Com intuito de propiciar uma visão mais detalhada sobre os resultados obtidos com o quase-experimento, foi realizada uma entrevista estruturada com os professores do grupo experimental. A avaliação do cod[edu] feita pelos professores foi positiva, uma vez que afirmaram que o modelo proposto auxilia no engajamento dos estudantes, ajudando também no processo de aprendizagem dos alunos na disciplina de programação.

6. Conclusão

A disciplina de programação é essencial na formação do discente de Computação uma vez que as habilidades adquiridas na sua aprendizagem corroboram para a resolução de problemas reais. Entretanto, os obstáculos encontrados para desenvolver o raciocínio lógico matemático, analisar e elaborar soluções para resolver os problemas algorítmicamente podem interferir no aperfeiçoamento dessa habilidade. Tais dificuldades são a grande causa, segundo Martins e Correia (2003), de reprovação, desmotivação e desistência dos estudantes nos cursos de Computação.

O ambiente de aprendizagem cod[edu] descrito, implementado e testado durante a pesquisa aqui reportada, tem a finalidade de amenizar esse problema, através de funcionalidades que permitem ao docente acompanhar e fornecer o *feedback* das atividades rapidamente, além de ajudar a promover e manter o engajamento do estudante durante o seu processo de aprendizagem.

A presente pesquisa cumpriu os objetivos traçados e obteve êxito ao utilizar *gamification* como estratégia no engajamento de estudantes no aprendizado de programação, no entanto os resultados obtidos conduziram aos seguintes trabalhos futuros:

- Mapear os períodos de maior e menor engajamento estudantil ao longo do processo de aprendizagem do discente.
- Replicar o experimento realizado em outras turmas de programação para alunos no início do curso, com um número maior de interações e participantes.

7. Agradecimentos

Tatyane Souza Calixto da Silva agradece ao CNPq pela concessão de bolsa de mestrado no período de realização desta pesquisa

Referências

- BLOOM, B.S.et al. Taxonomy of educational objectives: The classification of educational goals. Handbook 1: Cognitive domain. New York: David McKay, 1956.
- BRAZIL, A; BARUQUE, L. Gamificação Aplicada na Graduação em Jogos Digitais. In: Anais do Simpósio Brasileiro de Informática na Educação. 2015. p. 677.
- BRITO, A.; MADEIRA, C. XP & Skills: gamificando o processo de ensino de introdução a programação. In: Anais dos Workshops do Congresso Brasileiro de Informática na Educação. 2015. p. 1124.
- CODECADEMY. Disponível em< <https://www.codecademy.com/pt>>. Acesso em: 09 de abr. 2016.
- CODE SCHOOL. Disponível em <<https://www.codeschool.com>>. Acesso em: 09 abr. 2016.
- CRONBACH, L. J. Coefficient alpha and the internal structure of tests. *psychometrika*, v. 16, n. 3, p. 297-334, 1951.
- DETERDING, S. et al. From game design elements to gamefulness: defining gamification. In: Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments. ACM, 2011. p. 9-15
- GIRAFFA, L. M. M.; MORAES, M. C. Evasão na Disciplina de Algoritmo e Programação: Um Estudo a Partir dos Fatores Intervenientes na Perspectiva do Aluno. In: Tercera Conferencia sobre el Abandono en la Educación Superior (III CLABES), México. 2013.
- GIRAFFA, L.; MULLER, L.; MORAES, M. C. Ensino Programação apoiada por um ambiente virtual e exercícios associados a cotidiano dos alunos: compartilhando alternativas e lições aprendidas. In: Anais dos Workshops do Congresso Brasileiro de Informática na Educação. 2015. p. 1330.

- GOMES, A.; MENDES, A. A teacher's view about introductory programming teaching and learning: Difficulties, strategies and motivations. In: *Frontiers in Education Conference (FIE), 2014 IEEE*. IEEE, 2014. p. 1-8
- INEP.(2012). Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira. Disponível em: <http://portal.inep.gov.br/> Acesso em maio de 2016.
- JESUS, E. A. de; RAABE, A. L. A. Interpretações da Taxonomia de Bloom no Contexto da Programação Introdutória. In: *Anais do Simpósio Brasileiro de Informática na Educação*. 2009.
- KRATHWOHL, D. R. A revision of Bloom's taxonomy: An overview. *Theory into practice*, v. 41, n. 4, p. 212-218, 2002
- LIKERT, R. A technique for the measurement of attitudes. *Archives of psychology*, 1932.
- MARTINS, S. W. ; CORREIA, L. H. A. (2003). “O Logo como ferramenta auxiliar no desenvolvimento do raciocínio lógico – um estudo de caso”. *Internacional Conference on Engineering and Computer Education – ICECE* Disponível em: <http://www.dcc.ufla.br/~lcorreia/bibtex/icece2003.pdf>
- SBC – Sociedade Brasileira de Computação. Currículo de Referência da SBC para Cursos de Graduação em Computação e informática. 1999.
- SCOTT, T. Bloom's taxonomy applied to testing in computer science classes. *Journal of Computing Sciences in Colleges*, v. 19, n. 1, p. 267-274, 2003.
- SEIXAS, L.,R.; GOMES, A. S.; MELO, I. J. Identifying engagement indicators to support educational research. In: *Information Systems and Technologies (CISTI), 2015 10th Iberian Conference on*. IEEE, 2015. p. 1-6.
- SHADISH, W. R.; COOK, T. D.; CAMPBELL, D. T. *Experimental and quasi-experimental designs for generalized causal inference*. Wadsworth Cengage learning, 2002.
- VIEIRA, V.; TEDESCO, P.; SALGADO, A. C. Modelos e Processos para o desenvolvimento de Sistemas Sensíveis ao Contexto. André Ponce de Leon F. de Carvalho, Tomasz Kowaltowski.(Org.). *Jornadas de Atualização em Informática*, p. 381-431, 2009.
- WHALLEY, J. L. et al. An Australasian study of reading and comprehension skills in novice programmers, using the bloom and SOLO taxonomies. In: *Proceedings of the 8th Australasian Conference on Computing Education-Volume 52*. Australian Computer Society, Inc., 2006. p. 243-252.
- YANG, T.; YANG, S.; HWANG, G. Development of an Interactive Test System for Students' Improving Learning Outcomes in a Computer Programming Course. In: *Advanced Learning Technologies (ICALT), 2014 IEEE 14th International Conference on*. IEEE, 2014. p. 637-639.