

Aplicação do Robocode como Instrumento para a Recepção de Calouros e Ensino de Programação de Computadores

Fillipe Almeida Paz¹, Kenia Kodel Cox¹

¹Departamento de Computação – Universidade Federal Sergipe (UFS)
Caixa Postal – 49.100-000 – São Cristóvão – SE – Brasil

{fillipe.paz, kenia}@dcomp.ufs.br

Abstract. *Although the computer permeates practically all human actions; what is due, among other things, to computer programs; there are, in Brazil, high rates of dropout and retention in courses of the area, especially in the programming classrooms; which needs to be reversed. Therefore, it is necessary, among other actions, an instrumentalization of education with resources that allows young people to feel welcomed on higher education, and motivated for programming, highlighted to this the digital games, especially Robocode - a battle game of virtual robots that requires an application to build a combat strategy, while motivates and entertains its users.*

Resumo. *Apesar do computador permear praticamente todas as ações humanas; o que se deve, dentre outras coisas, aos programas computacionais; observa-se, no Brasil, altos índices de evasão e retenção nos cursos da área, em especial em disciplinas de programação; o que precisa ser revertido. Para tanto, faz-se necessário, além de outras ações, a instrumentalização da educação com recursos que permitam aos jovens sentirem-se acolhidos no ensino superior, e motivados à programação, dentre os quais neste destaca-se os jogos digitais, em especial o Robocode - um game de batalha de robôs virtuais que requer a aplicação da programação para construção de estratégia de combate, ao mesmo tempo que motiva e diverte seus utilizadores.*

1. Introdução

O uso de computador na contemporaneidade é indispensável, pois esta ferramenta, no cotidiano, permeia praticamente todas as ações humanas: desde a consulta a dados bancários e institucionais em geral, processos de automação industriais, publicidade, até a integração total da relação dos seres humanos com a tecnologia, de maneira invisível e automática, conforme almeja a computação pervasiva, e a internet das coisas.

A apropriação dos inúmeros espaços hodiernos pelos recursos tecnológicos computacionais se deve à popularização da internet, ao desenvolvimento de hardware, bem como à construção de softwares, que potencializam, com programas de computadores, o uso das tecnologias de informação e comunicação para facilitar, agilizar, e até empoderar, a ação humana.

Através da programação, novos sistemas de comunicação e informação são criados, possibilitando a formação de redes sociais, criação de instrumentos de comunicação, aparelhamento da área médica, do entretenimento, e outros.

Porém, apesar da relevância da computação no contexto atual, observa-se uma

sensível discrepância entre o número de ingressantes e o número de concluintes dos cursos superiores da área [SBC 2016]. Isso deve-se aos elevados índices de evasão em tais cursos, mas também aos elevados índices de retenção em várias de suas disciplinas, principalmente as que envolvem programação [Gomes, Henriques & Mendes, 2008].

Considerando a importância das tecnologias computacionais, e seus impactos na sociedade contemporânea, o desinteresse do jovem brasileiro pelos cursos superiores da computação deve ser revertido. A instrumentalização da educação com recursos que possuem objetivo didático mas também são atraentes e motivadores aos jovens, a exemplo dos jogos digitais educacionais - jogos sérios - surgem como potenciais ferramentas a fim de possibilitar a superação deste quadro.

Segundo Cox & Bittencourt, alicerçados por múltiplos estudos:

Um jogo digital educacional pode ser entendido como um software que tem bem definidos objetivos próprios de educação: motiva os alunos para os estudos, promove a aprendizagem; isto por meio de atividades de diversão, ou seja, prazerosas e desafiadoras Savi, Von Wangenheim e Borgatto (2011), Lopes, Marques e Conte (2012), Pötter e Schots (2012), Merelles, Peixoto e Monsalve (2011), e assim, apresenta como requisitos possibilidades de efetuar testes de hipóteses, ou resolução de problemas geralmente complexos, e não se limita à simples memorização, Paraskeva, Mysirlaki e Papagianni (2010) e Mendes (2011) [Cox & Bittencourt, 2017, p 17].

Também, um fator relevante para a permanência dos ingressantes é a maneira como são recepcionados e como são apresentados aos aspectos administrativos, pedagógicos e relacionais na universidade.

[...] as primeiras exigências universitárias, sejam elas burocráticas (matrículas, carteiras estudantis) ou acadêmicas (nível de exigência das aulas), podem ser percebidas como muito bruscas, fazendo com que alguns se sintam perdidos frente ao cotidiano universitário. Essa percepção de mudança brusca revela o despreparo que em geral o calouro apresenta frente às demandas e o modo de funcionamento da universidade. A falta de um maior conhecimento sobre o que é a universidade e o que esperar dela, tanto em termos acadêmicos quanto pessoais, é um fator que pode concorrer para as dificuldades de adaptação [Bardagi, 2007, Melo-Silva, 2003].

E com este objetivo, de recepção de calouros, visando favorecer sua adaptação e acolhimento ao ensino superior, os jogos, possivelmente digitais, também são úteis, ao servir para incentivar a colaboração e integração; dentre outros papéis que favorecem a este processo. Conforme [Mcgonigal 2012], jogos colaborativos favorecem à cooperação, coordenação e co-criação entre os participantes. Tais jogos podem ser pensados de modo a evitar sentimentos de agressão, raiva, desapontamento ou humilhação, que podem surgir em jogos altamente competitivos. Com base nisso, tais jogos são alternativa a fim de desenvolver práticas integradoras em ambiente universitário.

Com o objetivo de oportunizar o ensino de programação de maneira lúdica e prazerosa, conciliando-o ao processo de integração dos alunos calouros, foi aplicada a ferramenta Robocode como parte das atividades da Semana de Integração da Computação 2018 na Universidade Federal de Sergipe. O Robocode é um jogo onde robôs, programados pelos estudantes, competem em um campo de batalha virtual. Vence aquele que apresentar a melhor estratégia para economizar a sua energia, reagir e proteger-se aos ataques, além de reduzir a energia do(s) oponente(s).

O restante deste artigo está dividido em 4 partes principais: *Revisão Teórica* quando são apresentados elementos que alicerçam teoricamente a iniciativa relatada; *Relato de Experiência* quando é descrito o processo de aplicação do Robocode; *Validação* para apresentação da análise da eficácia da ferramenta frente à proposta de trabalho; e *Conclusão*, contendo principalmente os resultados da experiência.

2. Revisão Teórica

2.1. Jogos na Educação

“*Os jogos facilitam a aprendizagem de conteúdos complexos e desenvolvem, no aprendiz jogador, importantes habilidades cognitivas como, por exemplo, a capacidade de resolver problemas, a percepção, a criatividade e o raciocínio lógico*” [Pantaleão, Amaral & Braga e Silva 2017], logo, estes podem favorecer o ensino de programação.

Dentre as dificuldades encontradas pelos estudantes para o aprendizado de lógica de programação, segundo [Cambruzzi & Souza 2015, apud Hinterholz 2009], destacam-se: a falta de motivação; dificuldades de estabelecer a relação entre a teoria e prática; e, a dificuldade de assimilar as abstrações e o desenvolvimento de raciocínio lógico-matemático.

Diante desses argumentos é possível apontar a aplicação de jogos para o ensino de programação como uma alternativa. Estes podem servir como instrumentos visando a redução dos índices de retenção dos estudantes não só em disciplinas que envolvem programação, mas também em quaisquer outras.

2.2. Ensino de Programação

Segundo [Gomes, Henriques & Mendes 2008], os quais apontam diversos outros estudos:

O ensino das linguagens de programação tem como propósito conseguir que os alunos desenvolvam um conjunto de competências necessárias para conceber programas e sistemas computacionais capazes de resolver problemas reais. Porém, a experiência tem demonstrado que existe, em termos gerais, uma grande dificuldade em compreender e aplicar certos conceitos abstractos de programação, por parte de uma percentagem significativa dos alunos que frequentam disciplinas introdutórias nesta área. Uma das grandes dificuldades reside precisamente na compreensão e, em particular, na aplicação de noções básicas, como as estruturas de controle, à criação de algoritmos que resolvam problemas concretos. Diversos estudos apontam um conjunto de causas que estão na origem deste problema, Tobar et al. (2001), Jenkins (2002) e Lahtinen et al. (2005) são alguns exemplos. [Gomes, Henriques & Mendes, 2008, p. 162]

Além disso, foi observado por [Souza, Batista & Barbosa, 2016] que entre os principais problemas atuais no ensino e na aprendizagem de programação estão: dificuldades dos alunos em aprender os conceitos de programação; dificuldade dos alunos na aplicação desses conceitos durante a construção de programas; e, a falta de motivação entre os alunos na realização da atividade de programação. Como solução averiguou-se que a utilização de visualização de programas e algoritmos; utilização de *serious games* têm sido abordados com frequência.

Os argumentos acima tornam evidente a necessidade de propor métodos de ensino de programação que sejam capazes de estabelecer uma conexão clara entre os conceitos a exemplo de variáveis e estruturas de controle, e os seus objetivos e efeitos

práticos, o que pode ser realizado ao propor a construção de algoritmos que resolvam problemas concretos e possam ser validados visualmente.

2.3. Recepção de Calouros

Segundo [Teixeira *et al.* 2008], é necessário que os cursos estimulem a integração social dos estudantes, na medida em que o grupo tem um papel fundamental na construção da identidade dos novos universitários e também na construção de uma rede de apoio afetivo e acadêmico que possa auxiliá-los em caso de dificuldades. Atividades de integração podem ser propostas dentro de cada curso e também entre os cursos, promovendo o contato dos estudantes com diversidade de ideias e pessoas, acrescenta ainda: *“Estudantes que se integram acadêmica e socialmente desde o início de seus cursos têm possivelmente mais chances de crescerem intelectual e pessoalmente do que aqueles que enfrentam mais dificuldades na transição à universidade.”* [Teixeira *et al.*, 2008, p. 186].

Partindo deste pressuposto, a Universidade Federal de Sergipe, criou a Semana de Integração da Computação, objetivando estimular a integração entre os estudantes calouros e colegas de outros períodos e também facilitar a sua identificação, ou não, com o curso que escolheram. Ambos os aspectos com potencial para reduzir os índices de evasão dos estudantes dos cursos de Computação.

2.4. RoboCode

O Robocode [SourceForge, n.d.] foi desenvolvido pela AlphaWorks, da IBM, com o objetivo de divulgar novas tecnologias de desenvolvimento. Trata-se de jogo de simulação de batalha entre robôs virtuais, no qual cada jogador utiliza objetos - atributos e métodos implementados por classe pré-definidas - disponibilizados pelo ambiente para criar do seus próprios robôs. Posteriormente o robô pode ser posto em batalha com adversários geralmente criados por outros desenvolvedores; e assim confrontando estratégias feitas com programas computacionais, e possibilitando testes e aprendizado.

Cada robô é uma entidade que controla suas ações e reações. Quando uma batalha começa, um método específico, que contém parte dos comportamentos implementados para o robô, é disparado, e enquanto esse mantém-se em execução, o robô movimenta-se e/ou atira. o [Pantaleão, Amaral & Braga e Silva 2017].

A lógica dos robôs no Robocode está concentrada em métodos, que definem as ações e os movimentos destes. Existem métodos, baseados na manipulação de ângulos e outros conceitos matemáticos, para: acertar e validar tiros, verificar se robô foi alvejado, se bateu na parede ou em oponentes; detectar adversários; e sinalizar final de batalha. [Pantaleão, Amaral & Braga e Silva 2017].

O ambiente contém um editor de código-fonte próprio para programar os robôs [Pantaleão, Amaral & Braga e Silva 2017]. É possível ainda configurar regras das batalhas: quantidade de rounds e o tamanho do campo, inserir robôs e criar times de robôs.

Na Figura 1 pode ser visualizada uma batalha entre os robôs nativos do Robocode. No canto inferior esquerdo da imagem é possível ver opções para pausar ou alterar a velocidade da batalha, o que facilita a observação dos resultados da implementação dos robôs; oportunizando ao educando correlacionar códigos e

respectivos efeitos, e desta forma aprender. No canto superior direito estão dispostas informações sobre a energia dos robôs, a qual é iniciada em 100. Um dos objetivos do jogo é conseguir reduzir a energia dos robôs oponentes a 0.



Figura 1. Uma batalha no Robocode.

3. Relato de Experiência

A experiência fez parte de projeto de recepção de calouros intitulado Semana de Integração da Computação, desenvolvido em 2018, por docentes e discentes da Universidade Federal de Sergipe com o objetivo de: oportunizar, e até antecipar, o primeiro contato com a programação de computadores, de forma lúdica, bem como integrar os calouros dos cursos de computação entre si, com alunos veteranos, e com o currículo, por meio da aplicação do Robocode. Pode-se ressaltar 6 etapas principais no projeto: (1) planejamento do curso; (2) seleção dos tutores; (3) preparação dos tutores multiplicadores e definição das regras do campeonato; (4) composição das turmas de calouros; (5) capacitação dos calouros para uso dos recursos do RoboCode para treinar seus robôs e (6) campeonato de batalhas dos robôs treinados pelos alunos ingressos.

Na fase de Planejamento do Curso, todos os recursos do RoboCode foram analisados visando identificar-se o conteúdo mínimo para composição de curso integrador, de curta duração, e que possibilitasse o treinamento de robôs, e o primeiro contato com a programação. Tais recursos foram sequenciados por aluno do curso de Engenharia da Computação, de maneira a garantir a abordagem didática com gradação das ferramentas mais fáceis seguindo para as mais complexas; sob a orientação de professora do mesmo departamento.

Como resultado do planejamento foi confeccionado material, em slides na Figura 2, para ser aplicado na capacitação dos calouros, bem como na dos tutores multiplicadores; promovendo o primeiro contato dos alunos ingressos com a programação, e preparando os tutores para capacitar as turmas de iniciantes. Neste a apresentação dos recursos do RoboCode foi permeada pela resolução e proposição de exercícios, Figura 3, que exploravam as ferramentas do jogo, com o cuidado de não distanciar-se do contexto da batalha, e de se evidenciar a relação entre código - Java - e os efeitos no jogo.

ATIVIDADE- RESOLVENDO

Implemente um robô que só atire caso a diferença entre duas distâncias sucessivas tomadas com relação a um mesmo oponente detectado seja igual a 0.

```

public void onScannedRobot(ScannedRobotEvent e) {

//Lógica da mira aqui...
if (controle == 0) {
    d1 = e.getDistance();
    //t1 = getTime();
    controle = 1;
} else {
    d2 = e.getDistance();
    //t2 = getTime();

    if ((d2 - d1) == 0) {
        setFire(3);
    }

    controle = 0;
    d2 = 0;
    d1 = 0;
}
}

```

Qual a função da variável controle?

Essa variável controla a sequência em que os valores da distância são guardados. É uma maneira de expressar em código quando d1 e d2 devem receber os valores

Figura 2. Exemplo de slide utilizado no curso.

ATIVIDADE

Implemente um robô que só atire caso a diferença entre duas distâncias sucessivas tomadas com relação a um oponente detectado seja igual a 0.

Alguma ideia?

Existe alguma finalidade estratégica na implementação desses requisitos?

Figura 3. Exercício proposto em aula .

Na Figura 2 consta atividade que tem por objetivo apresentar o conceito de variáveis conectado a uma estratégia de batalha. O robô só deve atirar caso a diferença entre duas distâncias sucessivas em relação a um oponente seja nula. Uma vez que para calcular a diferença entre dois valores obtidos no momento da batalha é necessário armazenar o valor da primeira medida, faz-se necessário o uso de variáveis. Ao mesmo tempo, explora-se a estratégia de batalha em que um robô espera que seu oponente pare de se mover para atirar, aumentando assim as suas chances de acerto. Assim observa-se o educacional e o lúdico sendo trabalhado simultaneamente.

De modo geral as aulas envolviam problemas cuja solução implicava na aplicação de estruturas de programação e recursos do Robocode. A partir disso, tornou-se possível compor as seguintes unidades de ensino: apresentação do Robocode: suas regras e sua interface gráfica; conceituação e construção de algoritmos; estruturas de repetição e condicional; conceituação e uso de variáveis; eventos/métodos da API Robocode; implementações envolvendo contextos de batalha. Ver infográfico que consta na Figura 4.



Figura 4. Infográfico com os principais conteúdos abordados com o uso do Robocode no treinamento com os calouros.

A equipe de tutores foi composta por 10 alunos inscritos como monitores na Semana de Integração da Computação, sendo esta última composta também por palestras e outras atividades além desta relatada. Eram todos alunos de cursos de Computação e a maioria estava na segunda metade da graduação. A maioria dos tutores desconhecia completamente o jogo de batalha de robôs virtuais; e a Seleção foi efetuada pela identificação de quais monitores demonstravam interesse na apresentação da proposta de iniciativa com aplicação do Robocode.

A Capacitação dos Tutores ocorreu em dois encontros de 4 horas; sendo explorado todo material - slides e atividades contextualizadas - confeccionadas na fase inicial. Também nesta foram discutidas e validadas: a ordem dos conteúdos abordados, o design dos slides, os objetivos de cada unidade de ensino, estratégias de apresentação e envolvimento dos calouros; a exploração da programação de computadores, por meio de recursos lúdicos e motivadores; o estímulo à cooperação e integração entre eles.

As regras do Campeonato foram definidas juntamente com os tutores e sofreram influência das que são aplicadas pela Liga Nacional de Robocode, promovida pela UNICAMP. Definiu-se a Composição de Turmas, sendo estas em número de 3, compostas por 30 calouros cada, sendo a capacitação destas de responsabilidade de grupos de 3 tutores. A seleção dos robôs para a batalha final, de 5 rounds, deveria ser precedida por seletivas internas às turmas, onde cada batalha seria composta por 3 rounds. E definidas regras para favorecer a colaboração e a integração dos estudantes.

A Capacitação dos Calouros ocorreu em 2 dias, em encontros de 3 horas; sendo as turmas então divididas em 9 grupos de 3 integrantes. Cada grupo deveria desenvolver um robô para o campeonato. Nos dias de treinamento os calouros foram estimulados a desenvolverem seus robôs para combate. E todos foram instruída a considerarem que a rivalidade deveria restringir-se estritamente ao ambiente virtual.

Finalmente, no Campeonato foram realizadas as seletivas internas nas turmas, de modo a compor a batalha final. Cada turma era formada por 9 grupos que foram tomados 3 a 3 para compor a primeira rodada de batalhas. As rodadas posteriores foram constituídas pelos robôs vencedores da etapa anterior até à final. Ver Figura 5a. A batalha final ocorreu entre 3 robôs, representantes das turmas iniciais, Figura 5b.

Definiu-se que as equipes cujos robôs foram derrotados nas batalhas deveriam juntar-se às vencedoras para ajudá-las em ajustes de implementação. Desse modo buscava-se incentivar a cooperação e integração entre os participantes, ao mesmo tempo esperava-se evitar possíveis aspectos negativos de iniciativa competitiva. Devido a essa

regra esperava-se que na última batalha cada robô fosse produto de toda a turma de onde ele foi selecionado; bem como estimular ao máximo a integração almejada nos pressupostos de recepção de calouros.

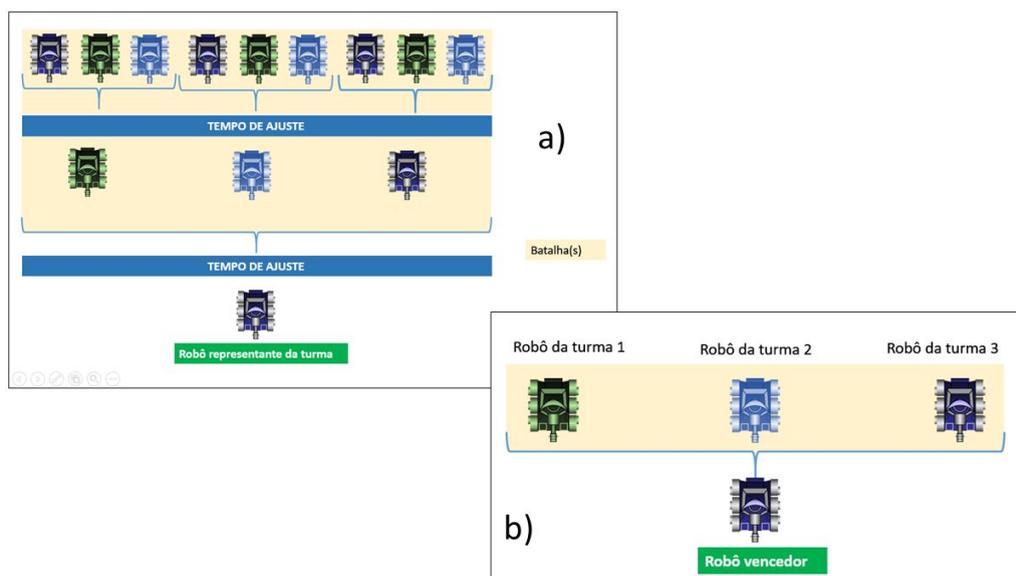


Figura 5. a) Processo de seleção para o robô representante da turma; b) Composição da batalha final.

Na batalha final verificou-se intensa participação e motivação de todos os envolvidos no curso de curta duração, movidos pelo desejo de ver o robô da sua turma vencedor.

4. Validação

Os resultados observados no Campeonato, a partir da realização satisfatória deste, da participação quantitativa e qualitativa dos calouros, e dos tutores, evidenciam que o Robocode é adequado à recepção de calouros, e para ensino de programação.

Visando confirmar estes resultados, foi aplicado questionário aos tutores, abordando a plataforma Robocode como ferramenta de ensino, a receptividade e aprendizado dos calouros e as capacitações realizadas. A partir da análise das respostas, são observados sinais da adequação da aplicação do Robocode como instrumento de ensino e aprendizagem de programação.

Sobre o potencial do Robocode como ferramenta para ensino de programação¹, mais de 50% dos tutores responderam positivamente, valorando a iniciativa, neste aspecto, com 4 ou 5 numa escala de valores inteiros de 1 a 5, onde 1 refere-se ao pior caso e 5 ao melhor caso. As capacitações², na perspectiva dos tutores, tiveram aprovação acerca da qualidade do material e sobre os efeitos destas em geral. E quando perguntados se a realização do campeonato contribuiu para a proposta do treinamento³,

¹ Para esta análise foram usadas as questões P1, P3 e P6 da Tabela 1.

² Os tutores, para esta análise, responderam com as duas maiores notas do intervalo, as questões P4; e 100% dos tutores responderam como suficiente a quantidade de material utilizado e selecionaram 4 ou 5 quanto a qualidade do conteúdo, conforme a pergunta P2 da Tabela 1.

³ Para esta análise foram consideradas as respostas dadas à questão P5 da Tabela 1.

90.9% responderam positivamente. Ver Tabela 1.

Em complemento às questões de composição da Tabela 1, houve ainda questão relativa à: (a) suficiência do material utilizado, para a qual 100% dos tutores responderam ‘sim’, e (b) sobre se o campeonato contribuiu para o aprendizado, para a qual 80% dos tutores também responderam ‘sim’.

Tabela 1. Parte 1 do questionário de validação.⁴

Perguntas	Alternativas				
	1 = pior caso	2	3	4	5 = melhor caso
P1: Como você avalia a receptividade dos conteúdos pelos alunos?	0%	0%	18.2%	63.6%	18.2%
P2: A qualidade (conteúdo) do material didático foi:	0%	0%	0%	63.6%	36.4%
P3: Quanto ao aprendizado por parte dos alunos foi possível notar que:	0%	0%	27.3%	45.5%	27.3%
P4: O número de horas do treinamento e campeonato foi:	0%	27.3%	63.6%	9.1%	0%
P5: Classifico, de um modo geral, o treinamento e campeonato realizados como:	0%	0%	18.2%	45.5%	36.4%
P6: Como classificaria a importância do RoboCode enquanto ferramenta utilizada para ensino de programação no treinamento?	0%	0%	9.1%	54.5%	36.4%

Para se ter uma visão panorâmica desta Validação, foi construído o gráfico que consta no Gráfico 1. Analisando-o percebe-se que somente na questão P4 - relativa ao tempo dedicado às capacitações, a iniciativa em análise foi avaliada negativamente, ou seja, as maioria das notas se aproximam do resultado negativo; sinalizando um aspecto das capacitações oferecidas que deve ser ajustado nas versões anteriores.

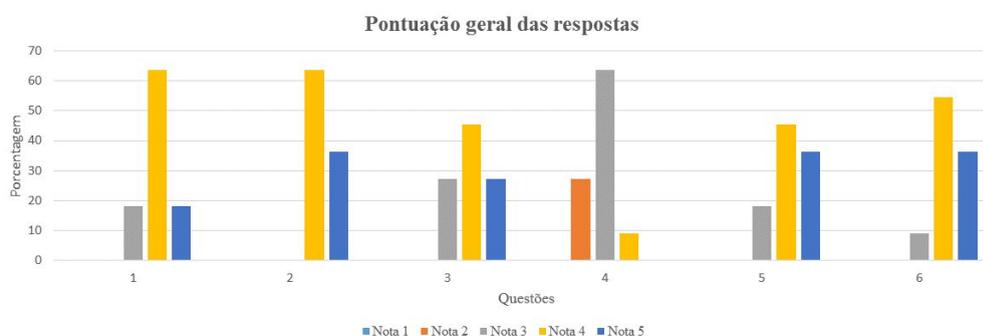


Gráfico 1. Panorama das pontuações às questões.

5. Conclusão

Com a aplicação do Robocode, é possível correlacionar os efeitos dos códigos

⁴ Na Tabela 1 são apresentadas as questões, e respectivos resultados, cujas respostas correspondem a valores de uma escala. Em complemento a estas houve ainda questão relativa à suficiência do material utilizado e sobre a influência do campeonato no aprendizado.

sobre uma batalha entre robôs representada graficamente, assim fomentando o aprendizado de programação de maneira contextualizada. O entorno de um confronto virtual confere objetivo às implementações, requerendo que os robôs atendam à regras para vencerem, e também fornece motivação necessária para que os estudantes continuem empenhados num processo colaborativo, ideal para momento de integração.

Por meio da observação da participação dos alunos na iniciativa, e da análise das respostas ao questionário aplicado, foi possível constatar que, de modo geral, o Robocode cumpre com o objetivo de servir como instrumento para recepção de calouros e ensino de programação, contudo, a fim de confirmar a efetividade das conclusões deste trabalho, observou-se também que fazem-se necessárias alterações e adições em versões futuras: (a) contemplar a RECEPÇÃO DE CALOUROS no questionário; (b) ampliar os tempos de capacitação, em especial para os calouros; (c) aplicar questionário também aos calouros; (d) organizar os questionários por tema; (e) organizar mais edições da SIC de forma a ampliar as evidências.

Referências

- Sociedade Brasileira de Computação (SBC) (2016). Educação Superior em Computação Estatísticas. 2016. Disponível em: <sbc.org.br/documentos-da-sbc/>. Acesso em: 02 jun. 2018.
- MCGONIGAL, Jane. Superpoderes Colaborativos (2012). In: MCGONIGAL, Jane. A realidade em jogo. 1. ed. Rio de Janeiro: BestSeller, 2012. p. 267-294.
- Gomes, A., Areias, C., Henriques, J., e Mendes, A. J. (2008). Aprendizagem de programação de computadores: dificuldades e ferramentas de suporte. Revista Portuguesa de Pedagogia, v. 42, n. 2.
- Cox, K. K.. & Bittencourt, R. A. (2017). Estudo Bibliográfico sobre o Processo de Construção de Jogos Digitais: A Necessidade de Sinergia entre o Educar e o Divertir. Revista Brasileira de Informática na Educação - RBIE, 25(1), 16-43.
- Cambruzzi, E., & Souza, R. M. (2015). Robótica educativa na aprendizagem de lógica de programação: Aplicação e análise. Anais do XXI Workshop de Informática na Escola (WIE 2015), 21–28.
- Hinterholz, O. (2009). Tepequeém: uma nova ferramenta para o ensino de algoritmos nos cursos superiores em computação. In ~ XVII-Anais do Workshop sobre Educação em Informática.
- Pantaleão, E., & Amaral, L. R., & Braga e Silva, G. (2017) Uma abordagem baseada no ambiente Robocode para ensino de programação no Ensino Médio. Revista Brasileira de Informática na Educação – RBIE, 25(3), 95-111
- Souza, D., & Batista, M., & Barbosa, E. (2016) Problemas e Dificuldades no Ensino e na Aprendizagem de Programação: Um Mapeamento Sistemático. Revista Brasileira de Informática na Educação – RBIE, 24(1), 95-111.
- TEIXEIRA, Marco Antonio Pereira. et al. (2008) Adaptação à universidade em jovens calouros, Revista Semestral da Associação Brasileira de Psicologia Escolar e Educacional (ABRAPEE), páginas 185-251.