

Experiências no ensino de Programação Orientada a Objetos: RoboCode, Greenfoot e Jogos de Tabuleiro no Ensino Superior

Luciene Cavalcanti Rodrigues^{1,2}, Giovani Cassiano Nogueira², Ana Paula Garrido de Queiroga³

¹ Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP) – Campus Votuporanga

Av. Jerônimo Figueira da Costa, 3014 - Bairro: Pozzobon - CEP: 15503-110 - Votuporanga, SP – Brasil

Grupo de Pesquisa CNPq: Tecnologias e práticas inovadoras aplicadas ao ensino

² Faculdade de Tecnologia de São José do Rio Preto

³ União das Faculdades dos Grandes Lagos

prof.luciene@ifsp.edu.br, giovani.cassiano99@hotmail.com, anaproj.eng@gmail.com

***Abstract.** The learning logic concepts is essential for courses in computer science, because it represents the logical reasoning for creating a sequence of steps to achieve a certain goal. The concepts of object-oriented programming language permeate all technical and superior courses and, together with the mathematics disciplines, end up forming the group of disciplines with greater evasion and reprobation. Within this context, this paper aims to demonstrate the application of alternatives to facilitate student learning, aiming to reduce failure and avoidance, as well as increase engagement, through greater integration and collaboration among students of higher education.*

***Resumo.** A aprendizagem dos conceitos de lógica de programação é essencial para os cursos da área de informática, pois representa o raciocínio lógico para a criação de uma sequência de passos para atingir determinado objetivo. Os conceitos sobre linguagem de programação orientada a objetos permeiam todos os cursos técnicos e superiores e, junto com as disciplinas de matemática, acabam formando o grupo de disciplinas com maior evasão e reprovação. Dentro deste contexto, este trabalho tem por objetivo demonstrar a aplicação de alternativas para facilitar o aprendizado do aluno, visando reduzir a reprovação e evasão, bem como aumentar o engajamento, por meio de maior integração e colaboração entre os alunos do ensino superior.*

1. Introdução

Segundo Mendonça Neto (2013) a lógica de programação é uma das mais importantes disciplinas para os alunos que ingressam em cursos na área de tecnologia da informação pois o desempenho do aluno na referida disciplina vai influenciar diretamente seu desempenho durante o curso visto que tais disciplinas exigem dos alunos raciocínio

lógico, resolução de problemas e capacidade de abstração da solução em uma representação formal e/ou em uma linguagem computacional (Jesus e Brito 2010).

De acordo com Oliveira, Rodrigues e Queiroga (2016), muitos educadores se empenham em experimentar novas metodologias ou ferramentas de possam minimizar este problema, tendo em vista que os alunos ingressantes no ensino médio e superior trazem posturas inadequadas para este tipo de aprendizagem, como por exemplo, o medo de errar, atrapalhando o desenvolvimento de exercícios que precisam de inferências e experimentações para chegar ao resultado satisfatório.

No tocante ao ensino de Linguagem de Programação Orientada a Objetos (POO) em cursos superiores de tecnologia, este trabalho tem o objetivo de demonstrar o uso do Greenfoot, Robocode, Jogo de Tabuleiro e APP Digital como ferramentas para auxílio à aprendizagem dos conceitos com a linguagem de Programação Java. As experiências relatadas foram realizadas com alunos do curso superior de Tecnologia em Informática para Negócios da Faculdade de Tecnologia de São José do Rio Preto e do curso superior em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – campus Votuporanga entre os anos de 2016 e 2017.

2. Jogos e Ferramentas Aplicados ao Ensino de POO

De acordo com Papastergiou (2009 apud Santos, Santos e Cysneiros Filho 2013), o termo “Aprendizado Baseado em Jogos Digitais” representa uma área de pesquisa que parte da premissa de que jogos de computador podem satisfazer os objetivos educacionais enquanto possuem o potencial de tornar conteúdos de uma disciplina mais: (i) fáceis de aprender, (ii) agradáveis, (iii) interessantes e, portanto, mais efetivos. O diferencial dessas mídias digitais está no fato de já serem parte do cotidiano e cultura dos alunos. Ademais o apelo entre crianças e adolescentes, esses jogos estimulam as sensações de desafio, fantasia e curiosidade. O diferencial dessas mídias digitais está no fato de já serem parte do cotidiano e cultura dos alunos. Ademais o apelo entre crianças e adolescentes, esses jogos estimulam as sensações de desafio, fantasia e curiosidade. Apesar deste conjunto de benefícios, jogos digitais têm sido pouco empregados no currículo das disciplinas do Ensino Fundamental e Médio.

Segundo Silva e Kirner (2010) os jogos podem apresentar uma série de benefícios ao usuário, tais como desenvolvimento de raciocínio, auxílio na alfabetização, geração de regras e estratégias, ampliação cultural, memorização e muitos outros. O prazer na prática deve ser espontâneo, pois geralmente simulam atividades lúdicas do mundo físico. Saber converter estas atividades para um contexto educacional é importante para que a informação seja transmitida da maneira correta e eficiente. Para tal, o desenvolvimento dos jogos deve atender certas exigências: facilidade de interação, recompensa por metas alcançadas, acessibilidade, regras claras e, por fim, atrativo para o reuso. Alguns benefícios podem ser transmitidos através dos jogos: aprimoramento da coordenação manual, integração social e a compreensão e resolução de problemas lógicos. Todos estes conhecimentos sendo reforçados através de métodos de repetição e de tentativa e erro.

Nos cursos superiores de tecnologia supra citados, foram utilizadas apresentações, exemplos e atividades práticas com uso de jogos e ferramentas visando auxiliar a fixação dos conceitos de POO.

2.1. Greenfoot

Segundo Kölling (s.d.) o Greenfoot é uma ferramenta de software concebida para permitir que um iniciante adquira experiência com programação orientada a objeto. Ele suporta o desenvolvimento de aplicações gráficas na Linguagem de Programação Java. Al-Bow (2008) cita que este framework inclui classes para elementos de jogo úteis, como animações, movimentos, rolagem, caixas de texto, relógios e temporizadores, podendo ser utilizado com jovens e adolescentes para o ensino de programação. O framework possui comandos pré-estabelecidos e ao arrastar e soltar itens no ambiente virtual é gerado todo o código fonte em linguagem de programação Java, que pode ser alterado com conhecimentos bem simples.



Figura 1. Slides de POO – Utilizando o Greenfoot

Este framework foi utilizado para a explicação de conceitos básicos da POO durante os slides e exercícios, na Figura 1 é possível verificar que os slides da disciplina já contém exemplos com uso de um jogo com o Greenfoot. Para a criação de um jogo é necessária a utilização de classes, atributos e métodos já pré-definidos no ambiente, desta forma, os alunos precisam apenas entender o conceito e, utilizando a herança e a instanciação de objetos, criar as regras do jogo. Desta forma, além dos exercícios criados no NetBeans cujo foco é a solução de problemas comerciais simples, também é possível observar a aplicação dos conceitos em um ambiente de criação do jogo.

Durante uma aula os alunos seguiram um tutorial para a criação de um jogo exemplo (O Crab) e foram instigados a desenvolver um novo jogo para demonstração em sala de aula.

2.2. Robocode

O Robocode é um software criado com o objetivo de divulgar novas tecnologias de desenvolvimento, pois simula um ambiente de batalha entre robôs programáveis em linguagem de programação Java. O objetivo principal é a criação de um robô próprio, utilizando como base as classes disponibilizadas pelo ambiente, com o uso das classes e APIs da ferramenta, é possível desenvolver vários tipos de robôs e colocá-los em uma batalha. Os robôs possuem radar, mira e corpo, que, tendo sido programado, irá disputar com outros robôs dentro de uma partida configurada no ambiente.

Toda a programação dos robôs está baseada em herança e uso dos atributos e

métodos, que definirão seus movimentos e ações, tais como girar, atirar, poder de fogo da bala, velocidade entre outros. Dentro da batalha, ganha o robô que implementar a melhor estratégia e derrotar o adversário.

Durante a disciplina, o Robocode foi utilizado como ferramenta para a fixação de conceitos de POO em Java, foi lançado o desafio às salas para que cada aluno criasse seu robô para a batalha, na data estipulada foram sorteadas as chaves e então, dados os parâmetros das batalhas, os alunos começaram a colocar os robôs para jogar. Foi interessante notar a empolgação dos alunos, foram feitas disputas individuais e em duplas para que fosse possível notar as diferentes estratégias. Após a primeira etapa os alunos já programaram uma disputa em duplas já combinadas, de forma que a dupla de alunos poderia conversar e programar os robôs para lutarem juntos e não um contra o outro. Durante as disputas também foi possível notar a troca de experiências, os alunos perguntavam uns aos outros sobre as características e métodos utilizados por cada um.

Após a atividade os alunos foram convidados a responder um questionário¹, infelizmente dos aproximadamente 60 alunos de duas instituições diferentes envolvidos, apenas 26 participaram da pesquisa. Os alunos foram questionados sobre os conceitos de POO que puderam ser observados e utilizados na programação do Robocode, sendo possível notar que os conceitos mais observados foram Herança, Classes, atributos, métodos, variáveis, constantes, setters e getters, características estas muito importantes para o aprendizado de linguagens de programação, sejam para sua aplicação para fins comerciais, científicos ou educacionais (Figura 2).

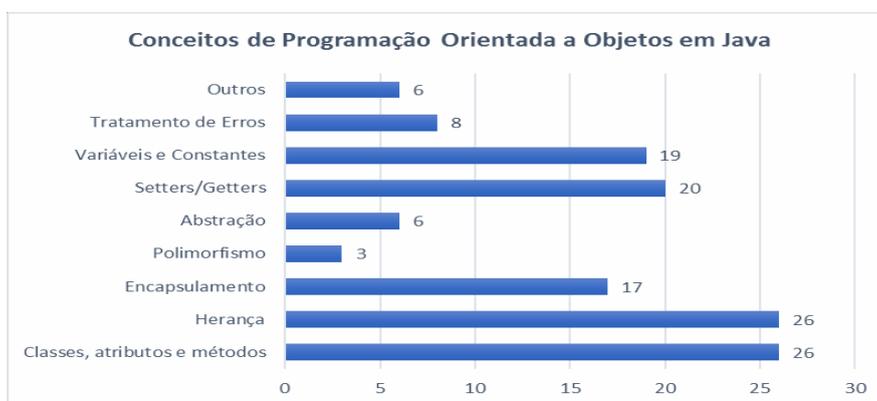


Figura 2. Conceitos citados como percebidos dentro do Robocode

Ao final do questionário, foi solicitado que os alunos fizessem um relato de sua experiência de programação do robô, indicando se foi prazerosa e se realmente tem algo útil na aprendizagem de programação, os resultados podem ser observados na Tabela 1.

Quanto aos pontos positivos e negativos, os alunos citaram como positivo novas formas de aprendizagem de linguagem de programação Java, interface intuitiva do Robocode, o desafio de pesquisa e aprendizagem de uma aplicação nova, com possibilidade de programação, como negativo foi citado que para atividades futuras, não permitir o uso de robôs prontos copiados da internet pois tiveram muitos robôs iguais.

¹ Disponível em: <https://goo.gl/forms/dbefG7yd5536bUah1>

Tabela 1. Respostas dos alunos para o uso do RoboCode

Prazeroso e muito útil a aprendizagem na programação orientada a objetos
Eu gostei bastante de mexer com isso, apenas queria que não tivesse dado problema no meu computador e eu pudesse ter aproveitado mais a experiência e feito um robo mais competitivo
Foi muito interessante, apesar de ser um projeto com muitos comandos internos, sendo assim, muitas possibilidades para criação de robôs, porém, senti dificuldades de implementar tais comandos, já que o API está totalmente em inglês.
A aprendizagem com o robocode foi importante para o java
A experiência foi muito útil e prazerosa, podendo praticar conceitos de orientação a objetos
Programar os robôs nos induz à pesquisar mais como é o funcionamento das classes, atributos e encapsulamento.
O robocode me ajudou a aplicar os métodos, gets e sets, heranças, classes.
Gostei muito, além de uma nova experiência me ajudou muito em algumas partes da programação que estava com dificuldade.
foi muito boa a experiência do robô pois melhorei a parte de programação em java e aprendi bastante coisas na programação
Foi boa, porém muito cansativa e confusa. Os comando eram muito particulares do programa, tendo que aprender o que todos faziam através de manuais. A parte mais útil foi manipular uma inteligência artificial pela primeira vez.
Sim, ajuda a aprender programação de forma divertida e descontraída
Foi interessante, aprendi como funcionava os Setter e Getters através do robo, o conceito de threads não foi discutido em sala, porém através do robo deu pra entender também (movimentos simultâneos de direção do robo), a parte da disputa foi muito legal e divertida.
A programação foi prazerosa, muitos conceitos novos na programação foram aprendidos. Na aprendizagem foi útil, pois foi necessário usar a matéria aprendida em aula, para realizar a programação.
Foi divertida, confesso que para mim, o que mais valeu a pena, foi aprender como procurar, pesquisar e interpretar diferentes APIs,
Sim, foi útil pois mostrou que a programação não é algo inalcançável, mas que podemos conseguir ser programadores algum dia.
Programar o robô foi bem legal, porém um pouco difícil, a programação do robô ajuda aprender de uma maneira bem diferente a Linguagem JAVA.
Sim, foi muito bacana e é um jeito mais divertido de aprender.
Foi uma experiência divertida, pois é a primeira vez que programo um robô em Java e vejo o resultado, vejo funcionando com as coisas que fiz.
Foi diferente e interessante pois foi colocado em prática conceitos de POO em jogo
Ajuda de forma divertida a utilizar o conhecimento aprendido em sala de aula, Heranças, Polimorfismo, getters, setters
Muito bacana e didático. Melhor ainda são as batalhas!

2.3. Jogo de Tabuleiro

Para o ano de 2017, além das atividades com Greenfoot e Robocode, os alunos devem organizar suas atividades, e conseqüentemente suas notas, por meio de um jogo de tabuleiro, desta forma, cada turma recebeu um tabuleiro em branco (A3) com 216 posições, que deveriam ser pintadas (coloridas) com 4 diferentes cores representativas de 4 diferentes ações no jogo, que são: Amarelo → Sorte, Vermelho → Stand-up, Azul → Desafio, Verde → Seminário. As regras para o semestre foram estabelecidas na primeira semana de aula (Tabela 2), na segunda semana os alunos retornaram com o tabuleiro preenchido e na terceira semana foram iniciadas as jogadas (Figura 3).

É interessante notar que cada sala distribuiu as cores da forma que achou mais conveniente, facilitando ou não sua estratégia de conquista, algumas turmas realizaram reuniões para traçar a melhor estratégia para todas as equipes. Como forma de controle, cada equipe escolheu uma cor e trouxe uma etiqueta adesiva para controle dos movimentos no tabuleiro, a cada semana as equipes giram a roleta e fazem seus movimentos, escolhendo assim sua estratégia para o cumprimento do semestre (Figura 4).

Tabela 2. Regras do jogo de tabuleiro

Regras Gerais
<ul style="list-style-type: none"> • Formar equipes com 3 integrantes, realizando jogadas semanais • Sortear as cartas de acordo com a cor do tabuleiro e sua jogada • Todos os membros da equipe devem participar • A equipe escolhe sua cor, nome e posição inicial no tabuleiro • Toda equipe deve realizar pelo menos 15 movimentos e no máximo 20 • O movimento só é válido se a tarefa for cumprida pela equipe • Batalha Robocode é obrigatória para todas as equipes • Serão 2 avaliações de conteúdo (provas) durante o semestre
Os Desafios
<ul style="list-style-type: none"> • São questões de múltipla escolha, cada integrante sorteia uma carta e digita o código no APP, aparecerá a questão que deverá ser respondida • Os desafios podem ser lançados semanalmente e também de acordo com os movimentos no tabuleiro • Também podem ser lançados como “surpresa”
Os Seminários
<ul style="list-style-type: none"> • Temas e conteúdos não apresentados em aula mas pertencentes à disciplina • Apenas 2 slides por aluno, excluindo-se o Inicial (Apresentação) , Código Fonte e o Final (Referências) • Tempo máximo de 2 aulas (incluindo montagem) • Cada integrante deve explicar uma parte • A equipe terá duas semanas para preparação do seminário
O Stand-up
<ul style="list-style-type: none"> • Apresentação de conceitos básicos vistos em aula (ou não) • Apresentação apenas de código-fonte, sem slides (Máximo 5 minutos) • O mesmo pode ser trazido impresso para ser distribuído na sala • Prazo de uma semana para a apresentação
Carta Sorte
<ul style="list-style-type: none"> • As cartas amarelas representam o quesito “Sorte” • <u>Pode trazer um benefício ou prejuízo para equipe, também pode beneficiar ou prejudicar outra equipe</u>
Regras Finais
<ul style="list-style-type: none"> • A equipe que conseguir cumprir primeiro todas as regras e chegar ao prêmio final será aprovada na disciplina, independente de suas notas (desde que tenha no mínimo 75% de presença e no mínimo 15 movimentos no jogo) • A equipe cujo membro for reprovado ou desistir perderá 5% da média • A equipe cujos membros forem aprovados sem recuperação ganham 0,5 ponto na média • A equipe que não entregar o interdisciplinar, além de ficar sem a referida nota, perde 2 movimentos, ou seja, também perde duas notas
Cálculo da Média
MF = Média_das_Jogadas * 0,3 + Prova1 * 0,2 + Prova2 * 0,2 + Interdisciplinar * 0,2 + Desafios * 0,1
Recuperação apenas para alunos com Média >= 4 e no mínimo 15 movimentos cumpridos

Figura 3. Tabuleiros coloridos pelos alunos



Como forma de controle, cada equipe escolheu uma cor e trouxe uma etiqueta adesiva para controle dos movimentos no tabuleiro, a cada semana as equipes giram a roleta e fazem seus movimentos, escolhendo assim sua estratégia para o cumprimento do semestre (Figura 4).



Figura 4. Jogadas

Para cada jogada existem cartas a serem sorteadas com temas para os seminários e stand-ups, também cartas de sorte. Os desafios serão respondidos via APP Aprendiz Digital, que, por meio de um login, o aluno poderá receber e responder questões de múltipla escolha sobre o conteúdo da disciplina.

As cartas são conhecidas pelos alunos, porém sorteadas, desta forma os alunos são capazes de escolher e cumprir as atividades, cabendo à equipe direcionar menor o percurso para o êxito no semestre (Tabela 3).

Tabela 3. Exemplos de Atividades para cada tipo de carta

A1-Perde o último movimento	A7-Escolha seu seminário e a data de apresentação	A13-A equipe dará monitoria da disciplina em dia e horário a combinar	A19-Desafio agora!
A2-Ganha um movimento	A8-Ganha 0,5 ponto na prova	A14-A equipe fará uma apresentação sobre o curso em uma escola, em dia e horário a combinar	A20-Dobra os pontos da última atividade
A3-Ganha consulta na prova	A9-Pede uma dica na prova	A15-A equipe dará monitoria de algoritmos antes da prova, em dia e horário a combinar	A21-Entrega parcial do interdisciplinar em 15 dias
Ve1-Apresentar o conceito de Threads com exemplos	Ve7-Apresentar a chamada de aplicações do SO através da aplicação com exemplos	Ve13-Apresentar sobre Collections com exemplos	Ve19-Apresentar a criação de Relatórios (iReport) com exemplos
Ve2-Apresentar o conceito de Sockets com exemplos	Ve8-Apresentar formatação de números com exemplos	Ve14-Apresentar o JavaFX	Ve20-Apresentar LookAndFeel Java com exemplos
Ve3-Apresentar leituras e escritas em arquivos texto com exemplos	Ve9-Apresentar padrões de projeto e expressões regulares	Ve15-Apresentar o RoboCode com exemplos	Ve21-Apresentar Bibliotecas para Java Swing com exemplos
Vo1-Conceito e exemplos: Classe, atributos, métodos e Herança	Vo10-Conceito e exemplos: passagem de parâmetros por valor e referência e seu uso como static	Vo19-Classe JFrame, atributos, métodos, herança e eventos	Vo28-Classe de Validação de Formulários
Vo2-Conceito e exemplos: Herança e construtores	Vo11-Conceito e exemplos: Validação de dados	Vo20-Classe JFormattedTextField,	Vo29-Classe de Conexão e chamada de procedures
Vo3-Conceito e exemplos: Sobrecarga	Vo12-Conceito e exemplos: Tratamento de exceções	Vo21-Classe JComboBox, atributos, métodos e herança	Vo30-Classe de Conexão e chamada de views

O intuito geral deste jogo é a aprendizagem colaborativa, auto-gestão da equipe e aprendizagem autônoma, além, é claro, da diversão e motivação dos alunos. Para a formação da equipe foi sugerido que os alunos formassem grupos cujos integrante tenham aptidões diferentes, não apenas com aqueles que evidentemente tem maior facilidade na

área de programação, visando realmente que a aprendizagem seja colaborativa.

2.4 APP Aprendiz Digital

Devido às necessidades apresentadas, surgiu a ideia de desenvolver um site e um aplicativo para supri as necessidades dos professores, desta forma foi desenvolvido um APP com plataformas de desenvolvimento de aplicações híbridas e conceitos de análise de dados. Sua base de dados é em NoSQL, utiliza serviços do Firebase como autenticação e real-time, o Ionic Framework está sendo utilizado junto com o Cordova para ser executado nas principais plataformas e a biblioteca ngCordova para acessar as funcionalidades dos dispositivos como a câmera para o Avatar. No aplicativo o aluno realiza seu cadastro e login, posteriormente poderá receber questões agendadas pelo professor ou digitar o número de uma questão e responde-la em 10 minutos. Todos os alunos cadastrados participam do Ranking de sua turma e do Ranking geral dos alunos (Figura 5).

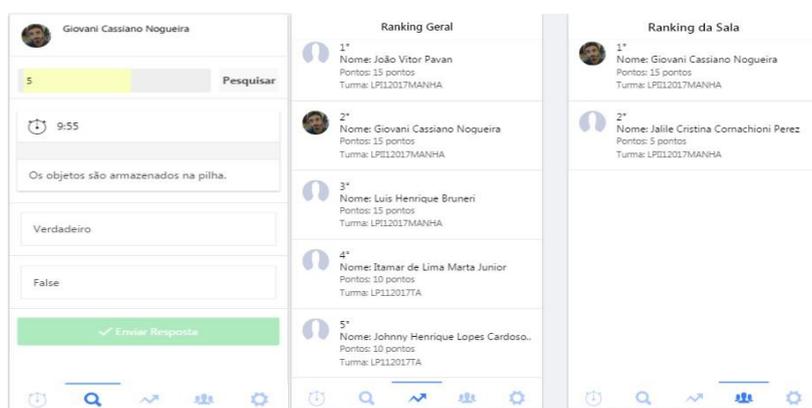


Figura 5. Telas do APP Aprendiz Digital

Ao final é possível verificar as notas de cada aluno para compor sua média final. O APP estará integrado à plataforma educacional também em desenvolvimento pelo grupo de pesquisas, onde os diversos jogos e ferramentas contarão com uma única base de dados e uma ferramenta de mineração de dados.

3. Conclusão e Resultados

É notório que o interesse dos alunos tem aumentado com o passar do tempo e com o apoio de ferramentas aliadas ao processo ensino-aprendizagem, o boa aula tradicional é mesclada com tais atividades lúdicas e colaborativas, fazendo com que o aluno sinta-se mais envolvido. As práticas com Robocode ocasionaram sensível envolvimento dos alunos com o conteúdo, dadas as respostas, nota-se que a atividade deve ser mantida nos demais semestres. O jogo de tabuleiro e o uso do APP Aprendiz Digital está movimentando as aulas, fazendo com que os alunos tenham que estar em constante contato com o conteúdo da disciplina, não retomando o assunto apenas na próxima aula, fato este extremamente prejudicial quando se fala do ensino de programação.

Em pesquisa realizada ao final da disciplina com 21 alunos respondentes, matriculados em 6 turmas diferentes onde a mesma metodologia foi aplicada, algumas respostas foram tabuladas na Tabela 4, sendo que, em uma escala entre 1 (Nada

envolvido) e 5 (Muito envolvido), todos os alunos responderam que estavam envolvidos com o jogo, em relação às dificuldades com Algoritmos e Linguagem de Programação, entre os respondentes foi possível notar que cerca de 41% possui bastante dificuldade com algoritmos e que 68,2% em linguagem de programação (respostas entre 3 e 5). Em relação às jogadas e as atividades dos grupos, apenas 4,5% relata como ruim (resposta 1), sobre os temas de stand-up, seminários e desafios, todos os alunos responderam como sendo bom a muito bom (respostas entre 3 e 5), já no tocante às cartas de sorte, 18,1% relatam negativamente (respostas entre 1 e 2), acredita-se que este percentual refere-se às cartas que não davam um ganho significativo ao grupo, mas sim, algo mais a ser realizado para completar a atividade, ou mesmo, fazer com que o grupo perdesse pontos, jogadas ou atividades já realizadas.

Tabela 4. Pesquisa Final sobre o Game

Questões	Respostas (%)				
	1	2	3	4	5
Indique abaixo, de acordo com sua opinião, seu grau de envolvimento com as atividades do game, sendo 1 (Nada envolvido) e 5 (Muito envolvido)	0	0	13,6	40,9	45,5
Em relação à disciplina de algoritmos, indique seu grau de dificuldade, sendo 1 (Pouca) e 5 (Muita)	36,4	22,7	18,2	18,2	4,5
Em relação à disciplina de programação, indique seu grau de dificuldade, sendo 1 (Pouca) e 5 (Muita)	18,2	13,6	40,9	18,2	9,1
Sobre as jogadas em si, as casas e as atividades que seu grupo desenvolveu, avaliaria como, sendo 1 (Ruim) e 5 (Muito Bom)	4,5	0	31,8	27,3	36,4
Os temas de stand up, sendo 1 (Ruim) e 5 (Muito Bom)	0	0	27,3	45,5	27,3
Os temas de seminários, sendo 1 (Ruim) e 5 (Muito Bom)	0	0	18,2	45,5	36,4
Os desafios, sendo 1 (Ruim) e 5 (Muito Bom)	0	0	31,8	27,3	40,9
As cartas de sorte, sendo 1 (Ruim) e 5 (Muito Bom)	4,5	13,6	27,3	18,2	36,4

Fazendo uma relação entre os conteúdos da disciplina e as atividades do game (Tabela 5), os alunos foram perguntados sobre: 1 - Em relação às demais metodologias, com provas e projeto, foi: [a) Mais fácil assimilar o conteúdo, b) Mais difícil assimilar o conteúdo, c) Tanto faz, seria difícil de qualquer forma, d) Tanto faz, seria fácil de qualquer forma, e) Indiferente]; 2 - Em relação ao seu grau de envolvimento com a disciplina, foi: [a) Mais fácil assimilar o conteúdo, b) Mais difícil assimilar o conteúdo, c) Tanto faz, seria difícil de qualquer forma, d) Tanto faz, seria fácil de qualquer forma, e) Indiferente]; 3 - Em relação à sua nota final, o peso das atividades foi: [a) Mais fácil obter nota, b) Mais difícil obter nota, c) Tanto faz, seria difícil de qualquer forma, d) Tanto faz, seria fácil de qualquer forma, e) Indiferente]; 4 - Em relação à obrigatoriedade de estudos semanais, com contatos obrigatórios com temas da aula e entregas de atividades, foi: [a) Mais fácil assimilar o conteúdo, b) Mais difícil assimilar o conteúdo, c) Tanto faz, seria difícil de qualquer forma, d) Tanto faz, seria fácil de qualquer forma, e) Indiferente].

De acordo com as respostas tabuladas na Tabela 5, foi possível notar que 72,8 cita que foi mais fácil assimilar o conteúdo, obter nota devido ao maior envolvimento que o game proporcionou. Quando perguntados sobre sua crença de a gamificação do ensino possa ajudar na aprendizagem, 81% responderam que sim, 4,8% não e 14,3% talvez. Sobre uma futura participação em outras atividades com jogos ligados ao conteúdo das disciplinas, 71,4% responderam que participariam novamente, 14,3% não e 14,3% talvez.

Tabela 5. Pesquisa Final sobre o Game

Perguntas	Respostas (em %)				
	A	B	C	D	E
1	85,7	4,8	4,8	4,8	0
2	81	9,5	4,8	0	4,8
3	47,6	19	4,8	9,5	19
4	76,2	4,8	0	14,3	4,8
Total Geral	72,8	9,5	3,6	7,2	7,2

Quando perguntados sobre possíveis melhorias as respostas foram diversas (Tabela 6), as mesmas serão consideradas em um próximo semestre.

Tabela 6. Melhorias

As regras poderiam ser mais detalhadas.
Poderia ter um campo onde mostra as questões que já foram respondidas
No geral pra mim foi assim, eu gostei muito de preparar os seminários e stand ups, pq aprendi muita coisa, mas peguei temas fáceis e creio que por isso não tive tanta dificuldade, creio que seria mais fácil pro resto da sala se os temas dos seminários e stand ups fossem divididos por semanas, por exemplo, o professor vai explicar nessa semana polimorfismo ou herança, aí então na semana seguinte poderíamos pegar os seminários ou stand ups referentes a esses temas, pq assim todos já iriam saber tbm do que se tratava e caso o aluno se confundisse e explicasse algo q não fosse tão correto, ficaria até mais fácil pra não nos confundirmos e até mesmo para participar mais dos seminários e stand ups dos colegas, pq já sabendo do q se tratava, poderíamos fazer mais perguntas e ter uma interação melhor, além disso, a pessoa seria obrigada a estudar mais pq como o professor já teria dado uma introdução, ela teria que trazer algo a mais pra turma... Seria legal também no caso de desafios aleatórios que pudéssemos ter um tempo de tolerância para responder, tipo 24 horas, pq no caso de algum imprevisto acontecer, daria tempo de avisar o professor ou o responsável e informar o ocorrido. Eu gostei das cartas de sorte, achei bem criativas, foi muito legal tbm poder trabalhar com o Robo Code... eu aprendi muito e foi divertido... Meu stand up foi muito legal, acho que nunca mais na vida vou esquecer For Each... rs =>) No geral, foi muito bom, como eu disse ter uma ordem cronológica pra trabalhar seria bacana pra mim, mas quanto a ideia da gameficação, é incrível, espero mesmo que outros professores possam aderi-la.
Acho que deva ser mantido, toda inovação que traga melhorias deve ser valorizada e aplicada.
A didática aplicada na disciplina foi positiva porque motiva o aluno a estudar constantemente, e buscar informações além da sala de aula.
Pra melhorar, as cartas de stand up e seminários poderiam não ser aleatórias e sim seguir uma sequência logica do conteúdo. Acho que a dificuldade de muito foi por conta disso. Se as jogadas fossem numa plataforma web ganharíamos tempo de aula.

Referências

- KÖLLING, Michael. Tutorial do Greenfoot. Kent University. Disponível em: <<http://www.greenfoot.org/files/translations/Brazilian/Tutorial%20do%20Greenfoot.htm>>. Acesso em : 17-12-2014.
- MENDONÇA NETO, V. S. (2013). A utilização da ferramenta Scratch como auxílio na aprendizagem de lógica de programação. Anais do II Congresso Brasileiro de Informática na Educação (CBIE 2013). Workshops (WCBIE 2013).
- OLIVEIRA, Manassés Vitorino; RODRIGUES, Luciene Cavalcanti; QUEIROGA, Ana Paula Garrido de. Material didático lúdico: uso da ferramenta Scratch para auxílio no aprendizado de lógica da programação. Anais do XXII Workshop de Informática na Escola (WIE 2016). V Congresso Brasileiro de Informática na Educação.
- SILVA, Kleber Anderson Corrêa e; KIRNER, Claudio. Vantagens educacionais no uso de jogos em Realidade Aumentada. In: CINTED-UFRGS Novas Tecnologias na Educação. V. 8 N° 3, dezembro, 2010.