Ferramenta para a Avaliação de Aprendizado de Alunos em Programação de Computadores

Rodrigo de Barros Paes[1], Romero Malaquias[1], Márcio Guimarães[1], Hyggo Almeida[2]

[1]Instituto de Computação – Universidade Federal de Alagoas (UFAL)
Maceió - Alagoas – Brasil

[2]Departamento de Sistemas e Computação – Univ. Federal de Campina Grande (UFCG)
Campina Grande – Paraíba – Brasil

{rodrigo, rbsm, masg}@ic.ufal.br, hyggo@dsc.ufcg.edu.br

Abstract. *This paper introduces The Huxley, a tool for supporting the evaluation of students within Computer Programming disciplines. The main functionalities are detailed, and guidelines for using the tool in practice are described. Finally, some results of the application of The Huxley in computer programming classes at Federal University of Alagoas are presented.*

Resumo. *Neste artigo apresenta-se a ferramenta The Huxley para apoio à avaliação do aprendizado de alunos em disciplinas de programação de computadores. As principais funcionalidades da ferramenta são detalhadas e as diretrizes para utilização da ferramenta são descritas. Por fim, são apresentados resultados da aplicação da ferramenta em turmas de programação da Universidade Federal de Alagoas.*

1. Introdução

Em uma sociedade informatizada, a Tecnologia da Informação serve como suporte para a maioria das atividades empresariais, industriais e educacionais da população. Sendo o veículo moderno para a transmissão de conhecimento e a automação de atividades diárias, a TI torna-se indispensável à população, em todos os níveis e classes sociais, o que é respaldado pelos recorrentes investimentos do governo em inclusão digital. Diante deste cenário, a demanda por profissionais de TI é algo imediato e crescente em todo o mundo. Segundo estudo elaborado pela BRASSCOM (Associação Brasileira das Empresas de Tecnologia da Informação e Comunicação), apenas no Brasil, no final de 2012 há um déficit de 115 mil especialistas em TI, ante 92 mil em 2011.

Programação de Computadores é um dos principais fundamentos para profissionais de Tecnologia da Informação. Todos os cursos relacionados à TI, incluindo técnicos, superiores ou de especialização, possuem Programação de Computadores como disciplina básica, sendo seus conceitos utilizados em diferentes áreas, desde Engenharia de Software à Computação Gráfica. Além disso, várias instituições de ensino médio já incorporaram a disciplina em seus programas, havendo atualmente fóruns de discussão na Sociedade Brasileira da Computação para inclusão da disciplina no currículo básico do ensino médio.

DOI: 10.5753/CBIE.WCBIE.2013.203

O aprendizado de programação tem como um dos pilares a prática. Apesar de se tratar de uma atividade que exige criatividade e raciocínio lógico, o aperfeiçoamento vem mesmo com a prática e a repetição de exercícios. Ao resolver uma maior quantidade de exercícios, o aprendiz se torna apto a associar novos exercícios àqueles já resolvidos, adaptando soluções e ganhando aos poucos uma produtividade de nível profissional. O desafio está na viabilização deste processo, considerando a necessidade de formação em massa, com cada vez mais alunos em sala de aula. Além disso, a personalização do ensino se torna prejudicada, pois se tem uma relação bem maior de alunos por professor.

A ferramenta The Huxley se apresenta como uma solução eficaz para estes problemas, viabilizando a melhoria de formação massificada de alunos, porém sem perder o controle de evolução do aprendizado individual. Esta é a principal inovação da ferramenta, diretamente relacionada ao sucesso da formação de novos profissionais de Tecnologia da Informação.

Neste artigo apresenta-se a ferramenta The Huxley, suas principais funcionalidades e como utilizá-la na prática. São apresentados também os resultados da aplicação da ferramenta em turmas de programação de computadores no curso de graduação da Universidade Federal de Alagoas. Os resultados indicam que a ferramenta é efetiva no auxílio ao aprendizado, enquanto viabiliza o acompanhamento individualizado do desempenho de muitos alunos por parte do professor.

2. The Huxley

The Huxley é uma ferramenta web que permite a alunos submeterem código em diversas linguagens de programação como respostas a exercícios de uma base de centenas de problemas. Para cada submissão, o aluno recebe *feedback* da correção automática pelo sistema através de análise sintática do código e testes de aceitação.

Além dos alunos, os professores também são grandes beneficiados com o processo. Com o aumento do número de alunos em instituições de ensino, professores de Programação de Computadores e disciplinas afins são obrigados a investir grandes esforços na concepção de exercícios e, principalmente, na correção dos mesmos. Ainda que se utilizem monitores, trata-se de uma tarefa árdua e que, aos poucos, torna-se pouco personalizada de acordo com a real necessidade dos alunos – quanto mais alunos, menos personalizado o tratamento.

Com The Huxley, o professor tem uma visão mais analítica e fidedigna do desempenho de seus alunos, incluindo a quantidade de problemas resolvidos, porcentagem de acertos/erros, tipos de problemas com mais erros, detecção de plágio e erros específicos de cada aluno. Em outras palavras, o professor passa a atacar de forma mais eficaz os problemas peculiares de aprendizado dos alunos, em vez de consumir seu tempo com a atividade de correção.

Do ponto de vista prático, o The Huxley possui uma interface web, com acesso via login e senha, disponível em www.thehuxley.com. Professores e alunos possuem diferentes visões, ambas baseadas no conceito de *dashboard*, que permite uma visualização global do status de exercícios, avaliações e conteúdo já na página principal do sistema (Figura 1).

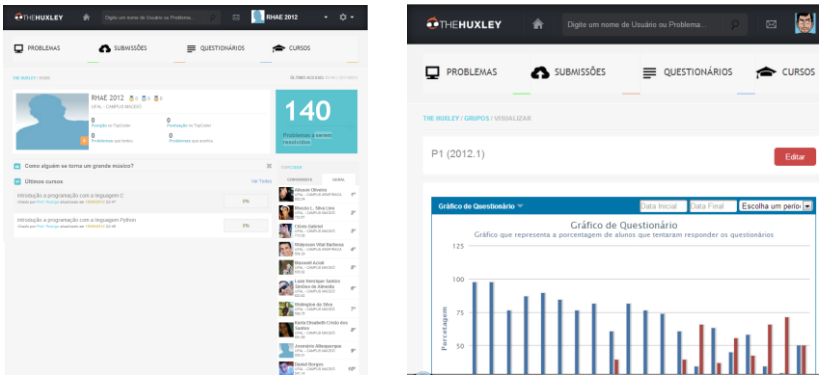

Figura 1. *Dashboard* do Aluno (esquerda) e do Professor (direita).

Através do módulo de análise do professor, é possível visualizar quais alunos estão com desempenho abaixo do esperado, eventuais plágios e, principalmente, o número de exercícios realizados (Figura 2).

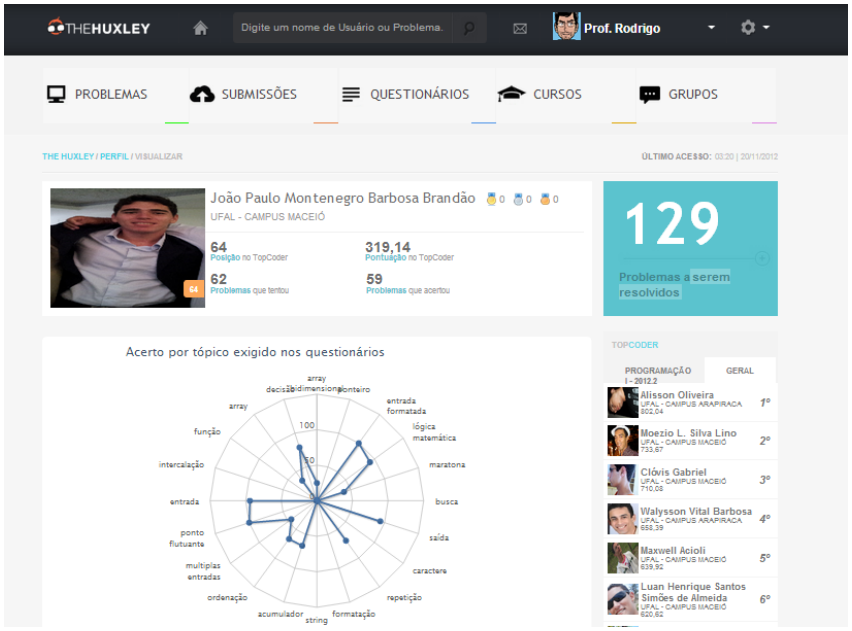

Figura 2. Análise de alunos.

Utilizando o The Huxley, o foco de trabalho do professor passa a ser na avaliação personalizada do aluno, com base nos erros específicos identificados durante as avaliações. Todo o trabalho de definição de exercícios e provas, correção dos mesmos, verificação de plágio, definição de notas e publicação dos resultados pode ser feita automaticamente através do The Huxley.

3. Funcionamento

Do ponto de vista prático, na Figura 3, tem-se uma visão geral de como o The Huxley funciona, com os passos sendo detalhados a seguir:

1. Através da interface do The Huxley, o professor cria questionários e avaliações a partir de uma base de centenas de exercícios existentes, podendo também criar

seus próprios exercícios. Os exercícios incluem descrição e testes de aceitação do resultado.

2. Os alunos acessam os exercícios definidos pelos professores, desenvolvem os programas e os submetem via interface da ferramenta.
3. Para cada programa em linguagem específica, The Huxley compila e testa o programa, armazenando o resultado para consulta posterior do aluno e do professor.
4. O resultado é visualizado em forma de planilha e gráfico pelos professores, que podem atribuir nota diretamente com base na avaliação do sistema. Os alunos também recebem o resultado para que possam re-submeter a solução caso haja algum problema de compilação ou nos testes de aceitação.

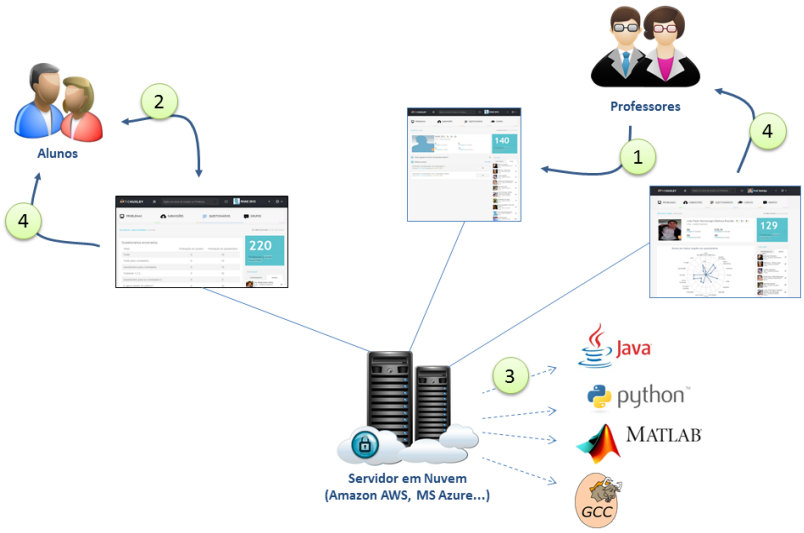

Figura 3. Visão geral de funcionamento da ferramenta.

Mais especificamente, as avaliações que valem nota, na maioria das vezes realizadas com o professor ou monitor em sala de aula, também se tornam automatizadas com o The Huxley. O professor define uma avaliação, que só fica disponível para os alunos na hora da prova. Após o tempo da prova ter se expirado, a avaliação fica indisponível no The Huxley, e o professor já pode exportar uma planilha com as notas após a correção do The Huxley.

Do ponto de vista dos alunos, há diversos mecanismos de incentivo que fazem com que os mesmos tenham motivação para acessar o sistema diariamente, extrapolando a relação estudo-nota. Por exemplo, o *top coder*, que lista os alunos com maior quantidade de exercícios resolvidos e as medalhas, para aqueles que possuem o maior Índice Huxley (que atribui maior valor aos exercícios mais difíceis), têm funcionado bem para garantir o sucesso do principal resultado da ferramenta: a maior prática da programação.

4. Caso Real: Aplicação na UFAL

Desde 2010.2, o Huxley vem sendo utilizado nas turmas de programação I da UFAL. O uso da ferramenta foi motivado por duas razões principais: (i) nós acreditamos que o quanto mais exercícios os estudantes praticam, mais eles aprendem, ou seja, nesse caso,

a quantidade tem como consequência uma maior qualidade do aprendizado; e (ii) os estudantes precisam obter *feedback* das suas tentativas de resolução de problemas o mais rápido possível, assim, as dúvidas não se acumulam e o próximo assunto a ser assimilado pelo estudante não fica prejudicado pelo não entendimento do assunto anterior. O Huxley lida com o primeiro ponto (i) fornecendo uma base de problemas classificados e estimulando os estudantes com algumas técnicas de *gamificação*. Em relação ao item (ii), a ferramenta possui um juiz online que dá uma resposta imediata ao estudante diante da sua submissão.

O funcionamento da ferramenta pode ser resumido da seguinte forma: o Huxley apresenta um conjunto de problemas e permite que os estudantes submetam o código fonte da solução desses problemas. Então, ele avalia a submissão e julga se o código do estudante produz os resultados esperados. Para o professor, ele fornece relatórios que permitem um diagnóstico detalhado do comportamento do estudante. Ele também oferece ferramentas para a realização de provas online, detector de similaridade entre códigos, fórum de discussão, dentre outras. Entretanto, para avaliar o impacto da ferramenta nas turmas de programação da UFAL, é importante entender as seguintes funcionalidades:

a) Base categorizada de problemas. O Huxley contém mais de 250 problemas de programação. Os problemas são categorizados por tópicos de programação (arrays, repetição, etc.) e por nível de dificuldade. O nível de dificuldade é calculado dinamicamente. Isto significa que quando um problema é resolvido por muitos estudantes, o problema é classificado como fácil. Este nível varia[1] de 1 a 10.

b) Os estudantes ganham pontos quando eles enviam submissões corretas. Quando um estudante envia um código fonte que produz os resultados esperados para um problema, ele ganha os pontos correspondentes aos nível de dificuldade do problema resolvido. Por exemplo, se um estudante resolve um problema de nível 7, ele ganha 7 pontos. Desta forma, quanto mais difícil é o problema, mais pontos o estudante ganha. A soma de todos os pontos obtidos compõe o TopCoder. Os 10 maiores TopCoder de cada turma aparecem com destaque na página inicial da turma. Já as 10 maiores pontuações de todas as turmas aparecem na página inicial da ferramenta, visível para todos os usuários.

Nos dados que obtivemos para fazer essa pesquisa, os estudantes de 2009.1, 2009.2 e 2010.1 não tiveram acesso a ferramenta. Já os estudantes de 2010.2, 2011.1, 2011.2, 2012.1 e 2012.2 usaram a ferramenta. No total foram analisados dados de 332 estudantes, alunos do primeiro semestre do curso de Ciência da Computação. Todos os dados e scripts utilizados nessa análise podem ser obtidos através do endereço: http://goo.gl/qhrhhD. O objetivo principal desse estudo é identificar como a ferramenta impactou o desempenho dos estudantes.

Desta forma, a primeira investigação diz respeito a identificar se o Huxley trouxe algum impacto nas taxas de aprovação ou desistência em programação 1. Consideramos

[1] Mais informações sobre como o nível dinâmico é calculado podem ser encontradas em: http://goo.gl/LSph1w

os semestres de 2009.1, 2009.2 e 2010.1 como sendo antes do Huxley e os períodos 2010.2, 2011.1, 2011.2, 2012.1 e 2012.2 como sendo depois do Huxley. Na Tabela 1 mostram-se os dados coletados desses dois períodos.

Situação dos Alunos/Percentual	Antes do Huxley	Depois do Huxley	Diferença
Aprovados	48.95%	51.12%	+2.17%
Desistentes	11.46%	19.54%	+8.08%
Mediana das notas	5.15	5.86	+0.71
Média das notas	5.72	5.60	-0.12

Tabela 1 – dados antes e depois da Ferramenta The Huxley.

Tanto o percentual de estudantes aprovados quanto o de desistentes aumentou. A mediana teve um pequeno aumento de 0.71 e a média diminuiu de 0.12. Entretanto, ainda não foi considerado o nível de dificuldade aplicado nas avaliações. A variação do nível de dificuldade durante os períodos pode levar a interpretações diferentes dos dados da Tabela 1. Porém, também sabemos que as avaliações utilizadas foram diferentes. Além disso, antes do Huxley, não temos dados sobre quais problemas foram aplicados e, portanto, não é possível estimar o nível de dificuldade para esse período. Após o Huxley, cada problema utilizado possui um nível de dificuldade, então com base nessa informação é possível avaliar o nível de dificuldade das avaliações para esse período.

Todas as avaliações depois da introdução do Huxley foram feitas na própria ferramenta. O Huxley permite que os problemas sejam agrupados em questionários. Então, as avaliações foram questionários. A nota do estudante é a soma dos problemas que ele conseguiu acertar e que fazem parte do questionário. Mais especificamente, nós coletamos o nível de dificuldade de cada questionário. Este nível é calculado fazendo a soma ponderada das dificuldade dos problemas que o compõe. Por exemplo, suponha que um questionário é composto de dois problemas, um de nível 3 e outro de nível 7. Suponha também que a nota do questionário pode variar de 0 a 10, sendo que o primeiro problema contribui com 60% da nota e o segundo com 40%. Então o nível de dificuldade do questionário é calculado fazendo x* 0.6 + 7*0.4 = 1.8 + 2.8 = 4.6. Na Tabela 2 descreve-se o nível de dificuldade de cada questionário.

Questão/Período	**2010.2**	**2011.1**	**2011.2**	**2012.1**	**2012.2**
1	1.70	1.70	1.70	1.70	1.80
2	3.00	3.00	3.00	2.75	1.80
3	2.50	2.50	2.50	3.00	3.30
4	1.25	1.25	1.25	2.50	3.25
5	2.80	4.10	3.00	2.00	3.00
6	3.00	3.00	7.00	3.00	3.30
7	2.70	2.70	2.70	2.70	2.75
8	4.00	3.65	3.70	4.00	7.00
9	5.10	4.05	4.05	5.00	5.55
10	6.00	4.00	4.00	3.70	5.75
11	3.65	5.10	5.10	4.05	5.00
12	4.05	6.00	6.00	4.00	5.75
13	5.00	NA	7.00	5.10	4.95
14	7.00	NA	NA	6.00	7.00
15	5.40	NA	NA	6.00	5.67
16	NA	NA	NA	7.00	8.00
17	NA	NA	NA	5.70	NA
18	NA	NA	NA	5.60	NA

19	NA	NA	NA	6.60	NA
Média	**3.81**	**3.42**	**3.92**	**4.23**	**4.61**

Tabela 2 – nível de dificuldade de cada questionário.

A partir dos dados da Tabela 2, é possível observar que a média aumentou de 3.8 para 4.6 desde o início da ferramenta. Isto sugere que o processo de avaliação se tornou mais rigoroso durante esse período. Em outras palavras, existem indícios que a ferramenta trouxe impacto no desempenho, fazendo como que mais estudantes desistam, mas que dentre os que permanecem, mais estudantes conseguem boas notas (aumento da mediana), e os alunos se tornaram capazes de resolver problemas mais difíceis (aumento do nível de dificuldade com a média praticamente estável). Entretanto, todas as variações foram pequenas, desta forma, esta conclusão deve ser ponderada de acordo.

Até o momento, foram analisados os impactos que a ferramenta pode ter causado em programação I. Como a ferramenta continua a ser utilizada na UFAL, também realizamos uma investigação para saber se é possível prever o desempenho dos estudantes em programação com base nas interações com a ferramenta.

Desde de 2010.2, o total de estudantes de programação 1 foi de 140. Desses, 7 nunca utilizaram o Huxley e, portanto, foram excluídos dessa análise. Desses 133 estudantes, 68 foram aprovados em programação na sua primeira tentativa. Nós chamamos esse grupo de CS1-PASSED. Este grupo, possui uma média de 70 problemas resolvidos, 333 submissões, 225.48 pontos no topcoder e uma chance de 66.94% de obter uma nota acima de 7 em programação 1. Um professor pode utilizar esses dados para, por exemplo, definir uma meta para seus estudantes: todo estudante deve resolver pelo menos 70 problemas cuja a soma dos níveis de dificuldade seja maior que 225.48 pontos. O professor também pode usar a ferramenta para dar mais atenção aos estudantes que estiverem longe desses objetivos.

Um dos maiores benefícios dessa ferramenta é sentido pelo próprio professor. Antes da ferramenta, o professor de programação da UFAL solicitava que os estudantes enviassem para ele aproximadamente 20 exercícios por semestre. Então, para cada um dos exercícios, o professor fazia o download do código fonte que havia sido enviado para o seu email, compilava, executiva e testava o código para um conjunto de casos de teste que ele tinha que preparar. Este processo é lento, demorado e susceptível a falhas humanas. Em um semestre, o professor realizava esse passo aproximadamente 600 vezes[2] e o estudante resolvia no máximo 20 problemas em um semestre. Na prática, a média de número de problemas que o estudante tentava resolver era menor que as 20. Após a ferramenta, estas atividades passaram a ser executadas pelo Huxley e os estudantes possuem agora uma média de 230 submissões e 59 problemas diferentes.

5. Ferramentas Relacionadas

Há diversas ferramentas relacionadas, porém com focos diferentes daquele abordado pela The Huxley. Exemplos de sistemas são: repositórios de cursos online, tais como Khan Academy (khanacademy.org), Udacity (udacity.com) e CodeSchool

[2] 29.5 estudantes por semestre X 20 submissões para cada estudante = 590

(codeschool.com); sites de competições e suporte à correção de código, tais como CodeAcademy (codeacademy.com), CodeChef (codechef.com), Programming Challenges (programming-challenges.com), Code Avengers (codeavengers.com) e HackerRank (hackerrank.com).

5.1 URI Online Judge (urionlinejudge.com.br)

Trata-se de uma ferramenta desenvolvida pela Universidade Regional Integrada de Erechin, baseada em online judge para melhorar o aprendizado no curso de algoritmos. A ferramenta permite correção em tempo real, interação entre usuários, flexibilidade na escolha das linguagens e dispõe aos alunos de um banco de questões categorizado em níveis e categorias. Entretanto, ela não possui ferramentas de apoio a turmas de alunos.

5.2 Khan Academy (khanacademy.org)

É um sistema que se propõe a melhorar a educação através de aulas gratuitas para qualquer um em qualquer lugar do mundo, a ideia principal é criar uma sala de aula global onde se possa aprender sobre qualquer assunto, mais de 4000 vídeos são dispostos, além de desafios e tutoriais. O sistema dispõe de ferramentas para acompanhamento dos alunos ou classe. Todo o conteúdo é gerado por usuários o que garante um ambiente colaborativo, o controle de qualidade do conteúdo é feito através de um sistema de votação. O sistema também dispõe de ferramentas de estimulo baseadas em gamefication, como a presença de achievements e gráficos de proficiências do aluno. No campo de programação, ele se restringe a conceitos básicos através de tutoriais, é possível interagir com um código da base, gerar seu próprio código e visualizar graficamente seus resultados. O ambiente não permite a construção de exercícios de programação mais complexos e não verifica similaridades entre as soluções.

5.3 Udacity (udacity.com)

O sistema tem como objetivo disponibilizar ao mundo um ensino superior mais efetivo, acessível e envolvente. Seus fundadores acreditam que o ensino superior é um direito básico da humanidade e almejam que seus estudantes avancem tanto na carreira quanto na sua educação. Udacity nasceu de um experimento criado na Universidade de Stanford onde Sebastian Thrun e Peter Norvig lecionaram um curso online de introdução a inteligência artificial. Desde então a equipe de educadores vem crescendo com a missão de mudar o futuro da educação. O sistema dispõe de cursos online gratuitos sobre diversas áreas, onde a qualidade do curso é garantida pela equipe de educadores. Os principais elementos do curso estão em vídeo aulas e questões sobre o assunto visto nas aulas. Os cursos são direcionados a um usuário. No campo de programação, não existe uma base de problemas para treino fora do mesmo, como também não existe suporte para um professor acompanhar o desempenho de uma turma.

5.4 CodeSchool (codeschool.com)

O sistema se propõe a ensinar de uma nova forma, que combina gamefication, vídeo aulas, prática no browser e conceitos de designer. Os cursos se propõem a ensinar uma variedade de habilidades de programação e de web design, com diferentes níveis de dificuldade. A qualidade dos cursos é garantida pela equipe do próprio sistema. Existem

ferramentas de estimulo baseadas em gamefication, como a presença de achievements. O principal foco do sistema é ensinar tecnologias web e não a disciplina de programação em si, não existe uma base de problemas para treino fora do mesmo, como também não existe suporte para um professor acompanhar o desempenho de um aluno ou uma turma, além de não existir verificação de similaridades.

5.5 CodeAcademy (codeacademy.com)

O sistema se propõe a construir um caminho melhor para ensinar qualquer a aprender ou lecionar como programar. Seu principal objetivo é transformar um mundo de consumidores de tecnologia em um mundo de construtores. O sistema ensina os conceitos de programação através de cursos online. A qualidade dos cursos é garantida pela equipe do próprio sistema, é possível criar cursos através do mesmo. O curso é direcionado a um usuário e não existe uma base de problemas para treino fora do mesmo, como também não existe suporte para um professor acompanhar o desempenho de uma turma. Existem ferramentas de estimulo baseadas em *gamefication*, como a presença de *achievements*.

5.6 CodeChef (codechef.com)

O sistema se propõe a ser uma comunidade global de programadores, onde os programadores podem se reunir, aprender, competir entre si e se divertir. Organizam eventos, competições e treinos para programadores. A base para o aprendizado é através da prática, competição e discussões em fóruns. O sistema tem uma base categorizada. Sua ferramenta de estimulo é baseada em rankings, que estimulam a competição entre os usuários. Porém o ambiente não tem um suporte para as atividades de um professor, através dele não é possível acompanhar o desempenho de uma turma, passar questionários ou verificar similaridades.

5.7 Uva (uva.onlinejudge.org)

Sistema que se propõe a ser repositório para problemas que podem ser usados tanto em competições como em treinos. Tem uma base categorizada e uma ferramenta de estimulo baseada em ranking, estimulando a competição entre os usuários. Porém o ambiente não tem um suporte para as atividades de um professor, através dele não é possível acompanhar o desempenho de uma turma, passar questionários ou verificar similaridades.

5.8 HackerRank (hackerrank.com)

O sistema se propõe a ensinar programação através de desafios dentro de diferentes domínios de Ciência da Computação. Tem uma base categorizada tem ferramentas de estimulo baseada em ranking, estimulando a competição entre os usuários. Porém o ambiente não tem um suporte para as atividades de um professor, através dele não é possível acompanhar o desempenho de uma turma, passar questionários ou verificar similaridades.

5.9 LearnBoost (learnboost.com)

O sistema se propõe a ser uma solução all-in-one que permite os professores a gerenciar suas classes, onde é possível criar planos de aula, listas de presença, calendários, além

de gerar relatórios de participação do aluno ou turma. Não é direcionado para programação, portanto não tem uma base de problemas, avaliador ou detector de similaridades.

6. Conclusões

Neste artigo foi apresentada a ferramenta The Huxley, que dá suporte à avaliação do aprendizado de alunos de programação em computadores. A ferramenta web permite que alunos submetam código em diversas linguagens de programação como respostas a exercícios de uma base de centenas de problemas. Para cada submissão, o aluno recebe *feedback* da correção automática pelo sistema através de análise sintática do código e testes de aceitação, sendo direcionado para conteúdo de suporte no caso de erros.

Os resultados da aplicação da ferramenta em turmas de programação de computadores no curso de graduação da Universidade Federal de Alagoas indicam que a ferramenta é efetiva no auxílio ao aprendizado, enquanto viabiliza o acompanhamento individualizado do desempenho de muitos alunos por parte do professor.

Do ponto de vista de infraestrutura, a ferramenta provê alta disponibilidade e escalabilidade, completamente integrada e implantada a plataformas de computação em nuvem existentes, as quais também fornecem outras características herdadas pela ferramenta, tais como segurança, tolerância a falhas e balanceamento de carga.

Como trabalho futuro, tem-se como foco a utilização de mecanismos de *Learning Analytics* para responder questões de análise mais aprofundada, tais como: qual o erro mais comum ao se utilizar determinada estrutura de um programa (for, while, etc)? Quais tipos de exercício têm mais acertos em uma primeira tentativa? Qual a evolução, dia a dia, de cada aluno em relação aos demais do mesmo grupo e em relação aos demais na ferramenta The Huxley? As respostas a estas questões podem ser extremamente úteis para a melhoria do processo de ensino-aprendizagem

Referências

Lori Lockyer and Shane Dawson. (2012). “Where learning analytics meets learning design”. In: Proceedings of the 2nd International Conference on Learning Analytics and Knowledge (LAK '12), Simon Buckingham Shum, Dragan Gasevic, and Rebecca Ferguson (Eds.). ACM, New York, NY, USA, 14-15.

Hendrik Drachsler and Wolfgang Greller (2012). The pulse of learning analytics understandings and expectations from the stakeholders. In Proceedings of the 2nd International Conference on Learning Analytics and Knowledge (LAK '12), Simon Buckingham Shum, Dragan Gasevic, and Rebecca Ferguson (Eds.). ACM, New York, NY, USA, 120-129.

George Siemens and Ryan S. J. d. Baker. (2012) Learning analytics and educational data mining: towards communication and collaboration. In Proceedings of the 2nd International Conference on Learning Analytics and Knowledge (LAK '12), Simon Buckingham Shum, Dragan Gasevic, and Rebecca Ferguson (Eds.). ACM, New York, NY, USA, 252-254.