



Estudo de Ferramentas de Apoio à Correção de Atividades de Programação no Contexto do Projeto IntroComp

Luiz Otavio Gerhardt, Isabel M. R. Bustamante, Luiz Felipe F. Mai, Nicole R. Nunes, Lucio Sandrini, Gabriel C. Rezende, Gustavo S. Alochio, Roberta L. Gomes

Departamento de Informática, PET Engenharia de Computação

Universidade Federal do Espírito Santo (UFES)

Av. Fernando Ferrari, 514, Goiabeiras – 29075-910 – Vitória – ES – Brasil

{gabrielcdrezende, lffmai, luizgerhardt93, isabelbustamante02, lucio.snd, nicole.rizzi115, gustavo.alochio}@gmail.com, rgomes@inf.ufes.br

Resumo. *Visando atrair mais jovens para a área de Computação, foi criado na UFES o projeto IntroComp, o qual promove anualmente um curso de Introdução à Programação para alunos do Ensino Médio da rede pública da Grande Vitória. Com o intuito de diminuir a carga dos professores, viabilizando com isso a ampliação do projeto, decidiu-se estudar o uso de ferramentas de apoio à correção de exercícios de programação. Apresentamos neste artigo uma análise realizada em cima de duas ferramentas aplicadas no projeto: SOAP e run.codes. Foram comparados os resultados obtidos com essas ferramentas aos obtidos com correções manuais anteriores, constatando-se com isso a eficiência das plataformas e os benefícios trazidos ao curso.*

1. Introdução

Com a disseminação da Computação em praticamente todos os setores da sociedade, a demanda por profissionais dessa área vem crescendo a cada ano. O mercado Brasileiro vem apresentando uma carência crescente de profissionais de Tecnologia da Informação e Telecomunicações (TIC) [Joia and Mangia 2017]. Em 2015, esse déficit chegou a 195 mil profissionais¹.

Apesar da necessidade de se atrair novos profissionais para a área, observamos um crescente desinteresse por cursos relacionados a TIC. Paralelo a isso, os índices de evasão de alunos nos cursos dessa área estão entre os mais altos do País [Araújo 2013], refletindo diretamente na escassez de mão de obra disponível no mercado. Além do problema da evasão, observa-se em alguns cursos altos índices de reprovação, principalmente em disciplinas que envolvem programação [Priesnitz et. al. 2012].

A programação de computadores realmente não é uma tarefa trivial, exigindo habilidades técnicas e cognitivas como bom raciocínio lógico e capacidade de interpretação [Scaico et. al. 2012]. Portanto, quanto mais cedo os alunos tiverem a oportunidade de desenvolver essas habilidades, maiores as chances de sucesso [Hayes and Stewart 2016]. Com isso, trazer o ensino da programação para o ensino médio definitivamente representa uma estratégia chave para se quebrar essa barreira de

¹ <http://computerworld.com.br/falta-de-profissionais-de-ti-na-america-latina-chegara-32-ate-2019>

aprendizado, além de despertar o interesse de novos jovens para a área de Computação [Pereira Junior et. al. 2005].

Visando justamente despertar este interesse em alunos oriundos de escolas públicas de ensino médio da Grande Vitória (ES), integrantes do Programa de Educação Tutorial (PET) do curso de Engenharia de Computação da Universidade Federal do Espírito Santo (UFES) criaram em 2010 o projeto IntroComp – Introdução à Computação [Meneses et. al. 2015]. A ideia é desenvolver nesses alunos o pensamento computacional [Wing 2006] por meio de cursos presenciais de programação. O pensamento computacional é uma ferramenta fundamental que pode ser aplicada na resolução de problemas em diferentes áreas e, portanto, de extrema importância não somente para profissionais de TIC [Gomes e Melo 2013].

Com o intuito de permitir a ampliação do projeto que, hoje, atende apenas 80 alunos por ano, decidiu-se estudar o uso de ferramentas de apoio para uma das tarefas que mais sobrecarregam os professores: a correção de exercícios de programação. Com isso, neste artigo apresentamos a análise de duas ferramentas aplicadas no projeto. Primeiramente, em 2016 foi analisado o SOAP [Oliveira, M. e Oliveira, E. (2014)], uma ferramenta de correção semi-automática em desenvolvimento na UFES, e posteriormente, em 2017 foi analisada a ferramenta run.codes².

Este artigo encontra-se estruturado conforme segue. Na seção 2 é apresentado o projeto IntroComp. Na seção 3 são apresentadas as análises realizadas em cima do uso das ferramentas SOAP e run.codes. Na seção 4 são listados alguns trabalhos que utilizam de alguma forma ferramentas de correção de código. Na seção 5 são apresentadas as conclusões e perspectivas do projeto.

2. Projeto IntroComp

O IntroComp³ foi criado no final de 2010 pelo PET Engenharia de Computação da UFES, visando proporcionar aos estudantes de Ensino Médio da rede pública da Grande Vitória a oportunidade de desenvolver a criatividade e o raciocínio lógico por meio do pensamento computacional. O projeto se baseia em uma estratégia adotada nos Estados Unidos [Kalelioğlu 2015], levando a esses alunos o ensino de programação básica de computadores. O foco na rede pública deve-se ao fato de que, em geral, os programas públicos de inclusão digital promovem treinamento apenas no uso de tecnologias (informática básica), e não na programação de computadores [Garcia, Correa and Shimabukur 2008].

Em 2011 ocorreu a primeira edição do Projeto, que incluiu somente uma escola da rede pública de ensino médio. Em 2012, estendeu-se para 4 escolas Estaduais. A partir de 2013 o Introcomp já abrangia praticamente todas as escolas da rede pública da Grande Vitória, incluindo as instituições federais. A Figura 1 apresenta os gráficos contendo os (a) números de inscritos, (b) números de alunos que realizaram o teste de seleção (prova de lógica), e (c) percentual de aprovação em cada edição do projeto. Vale observar que em 2015, como resultado de uma parceria com a Secretaria de Estado da Educação do Espírito Santo (SEDU), tivemos números massivos de inscritos em

² <https://run.codes/>

³ <http://introcomp.ufes.br/>



resposta a uma boa divulgação por parte da SEDU e conseqüente aumento na procura por parte das escolas. Com isso tivemos que pré-selecionar os alunos (com base em suas notas escolares de matemática, português e física) para realizarem a prova de lógica. No ano de 2016 notamos uma queda no número de inscrições se comparado ao ano anterior. Atribuímos isso ao fato de que não havia mais uma demanda reprimida como antes da realização da parceria, e o número de alunos inscritos passou a ser composto apenas por aqueles alunos não selecionados que ainda quiseram tentar o processo seletivo, e de novos alunos do ensino médio.

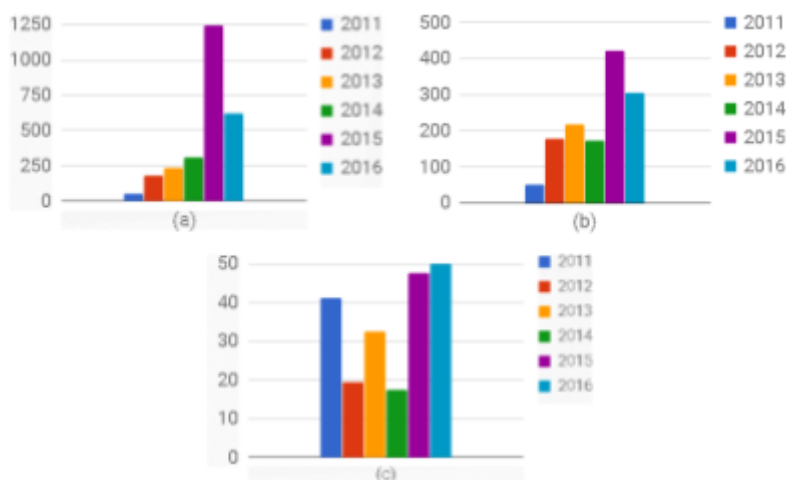


Figura 1: (a) Número de inscritos; (b) Quantidade de alunos que realizaram a prova de seleção; (c) Percentual de aprovados no Introcomp.

2.1. Metodologia

O IntroComp ocorre no segundo semestre do ano com aulas no Centro Tecnológico da UFES. Durante o primeiro semestre do ano a equipe responsável pelo curso realiza o planejamento anual que envolve programação das aulas, elaboração do material didático e divulgação. A ordem e o conteúdo das aulas foram escolhidos com base no livro “Introdução à Programação: Uma Nova Abordagem Usando C” [Varejão 2014], cuja elaboração e edição contou com a participação de vários alunos envolvidos no projeto.

As aulas ocorrem durante os sábados, em dois turnos: matutino (9h às 12h) e vespertino (14h às 17h). Ao final de cada aula, disponibilizamos *on line* o material de apoio, juntamente com os *Workings* (conjunto de exercícios de fixação), que devem ser feitos em casa e que contam nota para a conclusão do curso. Inicialmente, os exercícios eram submetidos via Moodle⁴, mas em 2017 mudamos para a plataforma *run.codes*⁵, cuja utilização será discutida posteriormente neste artigo.

Além do conteúdo referente à linguagem C, também são abordados, em dias específicos definidos no calendário, tópicos extras relevantes para a área da computação em geral. Os *Hacking Days* (HD) ocorrem aproximadamente três vezes durante o curso e neles são ensinados temas como Scratch, Web Design e Robótica.

⁴ <https://moodle.org/>

⁵ <https://run.codes/>



Ao final do curso, também propomos um trabalho computacional e uma prova para testar todo o conhecimento aprendido, além de submeter um questionário aos alunos que nos permitam avaliar o projeto e realizar mudanças a fim de melhorá-lo.

2.2. Preparatório para a OBI

Além do curso de Introdução à Programação, no primeiro semestre do ano seguinte organizamos um curso preparatório para a Olimpíada Brasileira de Informática⁶ (OBI) nas diferentes modalidades de programação (Nível 1, 2 e Universitário). A OBI, organizada pelo Instituto de Computação da UNICAMP, é uma competição elaborada nos moldes das outras olimpíadas científicas brasileiras, com medalhas como forma de premiação para os alunos mais bem colocados. A competição exige conhecimentos aprofundados em programação de computadores, com questões de níveis fáceis a mais complexos, impondo aos alunos o desafio de resolver problemas de maneira eficiente e no menor tempo possível, utilizando alguma linguagem de programação. Em paralelo, eles também aprendem a lidar com ambientes de pressão, tão comuns em competições e, certamente, no mercado de trabalho.

Durante as aulas do preparatório, que acontecem aos sábados (9h às 17h), são apresentadas metodologias que simulam ao máximo a prova da OBI. Além do treinamento aos sábados, os alunos também contam com os chamados *Contests*, que são coleções de problemas de temas específicos para que possam solucionar.

2.3. Resultados do projeto

O projeto tem obtido resultados cada vez mais importantes desde a sua criação. Conforme uma pesquisa realizada com os alunos que participaram do projeto até 2016, foi possível constatar que 30% dos que se formaram optaram por prestar vestibular para um curso na área de Informática/Computação. Todos eles apontaram o IntroComp como um fator determinante para essa decisão.

Com relação aos resultados dos alunos do IntroComp na OBI, podemos afirmar que estes foram expressivos. Não havia sequer participação de alunos do Espírito Santo nos quadros de méritos da competição, na modalidade de Programação, até o ano de 2011. A partir daí, com a realização do IntroComp o Estado passou a ter resultados positivos. Desde que o projeto iniciou o preparatório para a OBI, acumulou-se desde 2012: 2 medalhas de ouro, 3 medalhas de prata, e 2 medalhas de bronze. Particularmente em 2014, o projeto contou com o destaque de um dos seus alunos, que recebeu, além do primeiro lugar geral do Brasil na OBI, a medalha de prata na Olimpíada Internacional de Informática (IoI) em Taiwan. Também foram observados resultados positivos em competições internas de programação realizadas na própria UFES⁷, nas quais alunos do IntroComp venceram por três anos seguidos, disputando de igual para igual com alunos da graduação.

⁶ <http://olimpiada.ic.unicamp.br/>

⁷ <http://topcom.pet.inf.ufes.br/>



3. Ferramentas para correção automática de código

Apesar do potencial que o projeto apresenta e da enorme demanda não atendida, não tem sido mais possível ampliar seu público alvo. Dentre as barreiras encontradas está o fato de que é, em grande parte, responsabilidade dos professores (alunos de graduação que trabalham no projeto) elaborar, corrigir e prover aos alunos o feedback sobre os exercícios de programação. Algumas ferramentas de auxílio existentes já vinham sendo utilizadas para dar suporte à submissão, comunicação extra-classe e envio de feedback (e.g. o Moodle). No entanto, a atividade que mais sobrecarrega os professores é a correção dos exercícios.

Anteriormente, no IntroComp, a correção era totalmente manual. Desta forma, eram gastas horas de trabalho da equipe, horas que poderiam ser utilizadas em prol da qualidade do curso. Devido ao alto número de atividades a serem corrigidas, os professores do curso começaram a buscar alternativas para agilizar esse processo. O grupo tinha conhecimento da plataforma de correção automática BOCA⁸, porém, a ferramenta utilizada para programação competitiva não era adequada para o aprendizado, já que suas respostas seriam basicamente “certo” ou “errado”, impossibilitando que os alunos saibam a porcentagem de acerto de seus códigos, tornando processo de aprendizagem desgastante e de difícil avaliação.

Com isso em mente, em 2016, o grupo estabeleceu uma parceria com os desenvolvedores do SOAP, com o objetivo de realizar testes em cima dessa ferramenta, ajudando igualmente no seu aprimoramento. Posteriormente, em 2017, com base nos resultados obtidos com o SOAP, o grupo escolheu uma segunda ferramenta para realizar novos testes. Foi então estabelecida uma colaboração com os desenvolvedores do run.codes. A seguir, são apresentados os resultados obtidos.

3.1. SOAP

A ferramenta SOAP foi desenvolvida utilizando-se as linguagens de programação PHP e *Shell Script*. Essa ferramenta é dividida em dois núcleos: o Núcleo de Avaliação Diagnóstica (NAD) e o Núcleo de Avaliação Formativa (NAF) [Oliveira, M. e Oliveira, E. (2014)]. O NAD, a parte que foi mais utilizada pelo Introcomp, realiza as funções de correção semi-automática. Como foi ilustrado em Oliveira, M. e Oliveira, E. (2014): “A correção semi-automática é realizada pelas técnicas de clustering e por modelos de regressão linear. O clustering reúne em grupos os exercícios de programação com soluções semelhantes e indica que características explicam as semelhanças entre esses padrões. O modelo de regressão linear prediz as notas dos padrões em cada agrupamento com base nessas características e em exemplos de exercícios pontuados por um professor.”

Inicialmente, o SOAP foi utilizado no IntroComp em conjunto com as correções manuais para que fosse possível calibrá-lo no nível de desenvolvimento dos alunos do curso. Para isso, diversos exercícios foram utilizados para criar uma base de dados para a ferramenta, incluindo códigos de alunos de edições anteriores. Além disso, como o projeto vinha utilizando, com sucesso, a plataforma Moodle para a submissão de

⁸<http://www.ime.usp.br/~cassio/boca/>



exercícios e comunicação com os instrutores, a mesma foi mantida em uso. Foi então necessário realizar uma integração do SOAP com o Moodle, o que foi possível uma vez que esta última é *Open Source*.

3.1.1. Resultados obtidos com o SOAP

Uma vez calibrado, passamos a utilizar a ferramenta como fonte primária de correção. Desta forma, os alunos submetiam os códigos via Moodle. A ferramenta era então executada, atribuindo uma nota a todos os alunos e, a partir disso, eram realizadas revisões manuais para garantir que as notas não estivessem fora do esperado.

Durante esse experimento, foi possível poupar em até 30% o trabalho do instrutor na hora de avaliar os códigos entregues. No entanto, para uma considerável parte dos casos, as notas obtidas pela ferramenta não foram satisfatórias. A Tabela 1 apresenta uma amostra dos resultados para um dos exercícios aplicados durante o curso. Conforme exposto na tabela, podemos verificar que, embora algumas notas tenham chegado bem próximas das reais, outras foram totalmente discrepantes, tornando necessário o acompanhamento por parte dos professores para evitar que alunos ficassem com notas indevidas. Essa variação fica ainda mais explícita na Figura 2, em que pode-se verificar que a grande maioria das notas atribuídas teve uma discrepância de pelo menos 50 pontos.

Tabela 1. Comparação de avaliações manuais e com ajuda do SOAP

	Nota Original	Nota SOAP
Aluno 1	0	70
Aluno 2	0	100
Aluno 3	100	80
Aluno 4	95	100
Aluno 5	0	100
Aluno 6	100	90
Aluno 7	100	70
Aluno 8	87,5	100
Aluno 9	100	70
Aluno 10	87,5	100

De fato, esses testes também tinham como objetivo contribuir com o aprimoramento do SOAP, por meio dos códigos gerados pelos alunos do IntroComp uma vez que a mesma trata-se de uma ferramenta experimental. No entanto, por meio da análise dos resultados apresentados pela ferramenta, observamos que a taxa de acerto da mesma não era suficiente para que ela fosse utilizada de forma exclusiva na avaliação das atividades dos alunos. Com isso, mantivemos as revisões manuais em paralelo de modo a garantir que nenhum aluno fosse prejudicado.

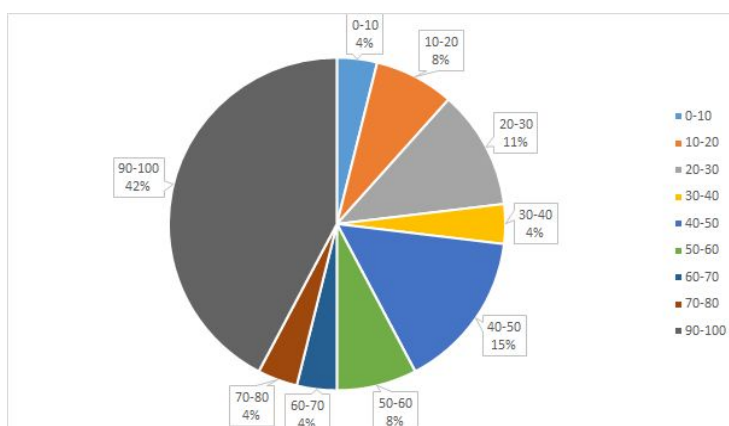


Figura 2: Divergência entre as notas atribuídas manual e automaticamente

3.2. Plataforma run.codes

Run.codes é uma plataforma on-line de correção automática de exercícios de programação. O ambiente virtual permite o cadastro de alunos, o envio de códigos realizados em diversas linguagens diferentes e, em questão de segundos, o recebimento da nota atingida.

Ao submeter uma atividade, esta é armazenada pelo site para posteriormente ser analisada e corrigida, podendo ser acessada e recuperada pelos professores ou pelo aluno durante toda a duração do curso. Uma vez submetido, o código fonte do aluno passa por um processo de compilação e teste. Considerando que a compilação ocorra sem erros, o run.codes irá executar o programa do aluno com as entradas dos casos de teste cadastrados pelos professores e usará a saída produzida pelo programa para atribuir a nota ao exercício.

Para podermos utilizar o run.codes no IntroComp, o primeiro passo foi ajustar os exercícios já existentes, uma vez que muitos eram sujeitos a respostas amplas e subjetivas. Os enunciados foram revisados para se tornarem mais claros e terem apenas um único formato possível de respostas. Com esse formato pré-estabelecido, foram criados arquivos de entrada e saída para futuras comparações feitas pela plataforma. Vale ressaltar que em alguns casos essa adaptação não foi possível e mantiveram-se os enunciados com respostas discursivas, amplas e subjetivas, principalmente os exercícios do início do curso, por serem mais conceituais e não envolverem diretamente programação em C. Nesses casos, a plataforma permite a entrega das respostas em um arquivo de texto para serem corrigidos manualmente.

Desenvolveu-se um manual de utilização do run.codes para instruir os alunos na criação das contas. Além disso, uma aula de laboratório para adequação dos alunos ao funcionamento da plataforma se fez necessária uma vez que alguns alunos enviavam soluções ainda subjetivas e era preciso habituar-se a padronização do formato dos envios.

3.2.1. Resultados obtidos com o run.codes

Ao utilizar o run.codes no IntroComp, os integrantes da equipe necessitaram de menos horas para corrigir os exercícios, analisando apenas os que não puderam ser adaptados ou que possuíam partes subjetivas de correção do código fonte (como utilização de



função). Outra vantagem está no fato dos alunos terem a possibilidade de enviar suas respostas diversas vezes antes do fim do prazo de envio da atividade, conseguindo ter uma melhor visualização de seu desempenho. Isto acarretou em um aumento da interação entre alunos e professores já que os alunos podem sanar suas dúvidas enquanto não tiverem obtido a nota máxima nos exercícios.

Visando-se analisar o impacto da plataforma run.codes, foram comparadas as médias dos alunos nos mesmos exercícios durante dois anos de curso, sendo o primeiro ano com correção manual e o segundo com correção automática. Com os dados obtidos foram feitos os gráficos da Figura 3, que representam as médias dos alunos em cada exercício. A correção automática foi utilizada a partir da terceira lista e apenas em 3 dos 5 exercícios da quarta lista, dado que as listas anteriores e alguns exercícios da quarta lista admitiam respostas subjetivas.

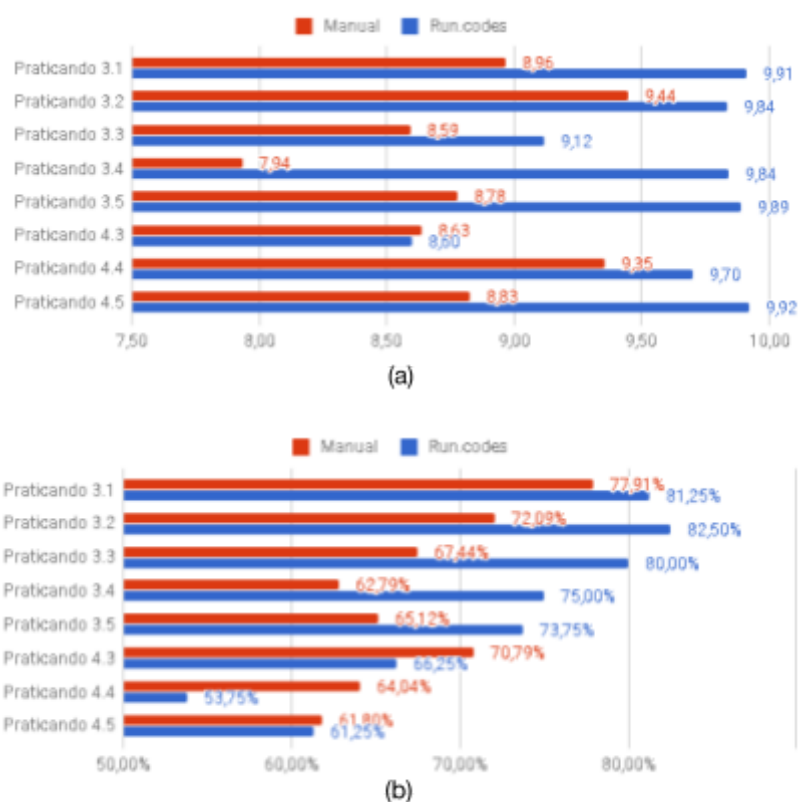


Figura 3: (a) Média de notas entre alunos que realizaram as respectivas atividades; (b) Porcentagem dos alunos que realizaram as respectivas atividades

O run.codes vem sendo utilizado no curso desde o começo da edição de 2017 e mudanças já foram constatadas, principalmente pelos professores. Ao analisar os resultados mostrados nos gráficos da figura 3, podemos perceber uma pequena melhora nas notas e participação dos alunos sendo que, em média, 4 alunos a mais passaram a entregar as atividades. A maior diferença, porém, foi verificada pelos professores que conseguiram remanejar o tempo para outras atividades do projeto e puderam se empenhar ao desenvolvimento de dinâmicas e didáticas em sala, além de novos exercícios e acompanhamento acadêmico dos alunos, dando mais ênfase ao conteúdo com baixo desempenho por parte dos estudantes. Vale ressaltar, então, que mesmo as



notas e participação dos alunos se mantendo, a mudança é viável e benéfica para o projeto.

4. Trabalhos Relacionados

O estudo sobre plataformas virtuais utilizadas na correção automática de códigos é datado de alguns anos, existindo diversas outras opções de ferramentas semelhantes às mencionadas anteriormente neste artigo (SOAP, BOCA, Run.codes), cada qual com suas particularidades para tratar o problema. Entre as ferramentas que mais se destacam nesse contexto podemos citar o CodeTeacher, MOJO e ProgTest.

CodeTeacher é uma ferramenta para desktop voltada para a correção de programas em Java. Tal ferramenta tem como princípio a correção do programa de maneira mais abrangente, analisando a estrutura, comportamento, conceito e saída padrão do código submetido. Outra particularidade do CodeTeacher é a necessidade do código já estar compilado antes do envio, impedindo a análise textual.

MOJO é uma integração de uma ferramenta de submissão de exercícios online, o Moodle, com um módulo que implementa os conceitos de Juízes Online (JO). A ferramenta permite a submissão de arquivos-códigos, em seguida a sua compilação e avaliação, a qual é realizada comparando resultados esperados com resultados obtidos pela execução do programa com dados formatados como entrada.

O ProgTest (Souza et. al., 2011) permite a criação de definições de testes por parte de professores e/ou alunos para avaliação do código. É utilizado um programa de referência, denominado Oráculo, fornecido pelo professor para avaliação das atividades dos alunos, utilizando diferentes ferramentas para testes como JUnit e CUnit. Ele também fornece um módulo para professores, em que se pode criar disciplinas e associar alunos a essas disciplinas e direcionar as atividades. O ambiente ProgTest foi desenvolvido para aceitar programas escritos em Java.

Ao procurar a ferramenta que mais se adequava ao IntroComp, percebemos que o CodeTeacher e o ProgTest não poderiam ser utilizados por aceitarem apenas programas desenvolvidos em Java e, como citado anteriormente, nosso curso aborda a linguagem C. O MOJO, por sua vez, possui uma interface que julgamos ser mais complicada para compreensão e uso pelo usuário e também para organização dos professores, se comparado ao run.codes que permite uma melhor acessibilidade tanto para alunos quanto para professores. Por fim, consideramos também a oportunidade de contato direto com os desenvolvedores dessa última plataforma.

5. Considerações finais

Este artigo apresentou as iniciativas e estratégias do projeto IntroComp e a aplicação e testes de ferramentas de correção automática de código. A crescente participação de alunos, os resultados das pesquisas realizadas junto a ex-alunos do curso, e os resultados do Estado nos últimos anos da OBI mostram que este projeto tem grande potencial para contribuir com a iniciação de nossos alunos de Ensino Médio na prática da programação.

Os resultados apresentados neste artigo mostram que a utilização de plataformas de correção automática só tem a acrescentar ao projeto. Inicialmente o uso de tais



ferramentas foi colocada em questionamento, pois torna mais difícil a detecção de erros ou vícios que podem ser adquiridos durante o processo de aprendizagem. Após períodos de teste foi possível constatar que essa dificuldade se torna menor quando é levada em consideração a proximidade das notas obtidas pela ferramenta com as atribuídas manualmente, bem como a economia de tempo que pode ser aproveitada para melhorar a qualidade do curso bem como viabilizar a ampliação do projeto. Acreditamos que esse seja um caminho fundamental em direção ao objetivo da ONG CODE.ORG de se colocar o ensino de linguagens de programação ao lado de matérias tradicionais como matemática, biologia e física.

Agradecimentos

Fundação de Amparo à Pesquisa e Inovação do Espírito Santo - FAPES. Processo nº 71904891.

Referências

- Araújo, A. (2013) “Evasão de alunos reflete na escassez de mão de obra”, In: *Jornal Diário do Comércio*, São Paulo, Janeiro, 2013.
- Oliveira, M., & Oliveira, E. (2014). Metodologia de Diagnóstico e Regulação de Componentes de Habilidades da Aprendizagem de Programação. In *CSBC 2014-XXII WEI (Workshop sobre Educação em Computação)*.
- Garcia, R.E., Correia, R.C.M., Shimabukuro, M.H. (2008) “Ensino de Lógica de Programação e Estruturas de Dados para Alunos do ensino médio”, In: *Anais do Workshop sobre Educação em Computação (WEI'2008)*, Belém, PA, pp. 246-249.
- Gomes, T., & Melo, J. (2013). O Pensamento Computacional no ensino médio: Uma Abordagem Blended-Learning. In: *Anais do XXI Workshop sobre Educação em Computação (WEI'2013)*. Maceió, AL–Brasil, p.651-660.
- Hayes, J., Stewart, I., (2016). Comparing the effects of derived relational training and computer coding on intellectual potential in school-age children”. In: *British Journal of Educational Psychology*, v. 86, n. 3, p. 397-411.
- Joia, L.A., Mangia, U. (2017). Career transition antecedents in the information technology area. In: *Information Systems Journal*, v. 27, n. 1, p. 31-57.
- Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code. org. In: *Computers in Human Behavior*, v.52, p.200-210.
- Meneses, L.F., Mai, L.F.F., Rosario, J., Oliveira, E. and Gomes, R.L., (2015). IntroComp: Atraindo alunos do ensino médio para uma instigante experiência com a programação. In *Anais do XXIII Workshop sobre Educação em Computação (WEI 2015)*, Recife, PE, SBC.
- Pereira Júnior, J.; Rapkiewicz, C.E.; Delgado, C.; Xexeo, J.A.M. (2005) “Ensino de Algoritmos e Programação: Uma Experiência no Nível Médio”. In: *Anais do XIII Workshop de Educação em Computação (WEI'2005)*. São Leopoldo, RS, Brasil.
- Priesnitz Filho, W., Abegg, I., Simonetto, E.O (2012). Uma Abordagem Diferenciada no Ensino de Algoritmos Através da Utilização de uma Lousa Digital. In: *Revista GEINTEC-Gestão, Inovação e Tecnologias*, v. 2, n. 2, p.129-137.
- Scaico, P.D. et.al. (2012)“Programação no ensino médio: Uma Abordagem de Ensino Orientado ao Design com Scratch”, In: *Anais do Workshop de Informática na Escola*.
- SOFTEX (2012) “Software e Serviços de TI: A Indústria Brasileira em Perspectiva”. Brasília, DF: Serifa Conhecimento e Comunicações, maio 2012.
- Varejão, F., (2014) *Introdução à Programação: Uma Nova Abordagem Usando C*. Elsevier.
- Wing, J. M. (2006). Computational thinking. In: *Communications of the ACM*, v.49,n.3,p.33-35.