# Eu sei o que vocês fizeram (agora e) na aula passada: o TSTView no acompanhamento de exercícios de programação

**Ano:** 2013

**Relação com o trabalho a ser proposto**: Utilização de ferramenta para medir o desempenho e entender as dificuldades de cada aluno com relação a disciplina de programação. Muito semelhante ao artigo learning[...] que usamos e a questão do The Huxley nas ideias buscadas a levantar em nosso estudo.

**Resumo:** Em aulas práticas de programação introdutória, o professor precisa ter uma visão global do desempenho e do andamento da turma nas atividades planejadas sem perder a visão individual do progresso e das dificuldades de cada estudante. Neste artigo apresentamos nossa experiência com um ferramental que desenvolvemos para viabilizar o contínuo monitoramento das atividades dos estudantes em laboratório. Nossas primeiras experiências demonstram que o ferramental torna possível detectar com mais facilidade as dificuldades individuais dos alunos, e os problemas nas atividades planejadas para a aula. Assim, tem nos permitido atuar de forma mais precisa e ágil na solução de problemas de ensino e aprendizagem.

**Palavras-chave:** [Não apontadas pelo autor] Monitoramento de atividades; Visão individual do progresso; Dificuldades individuais dos alunos

**Problema de pesquisa/proposta**: Monitoramento do desempenho e identificação das dificuldades individuais de cada aluno na disciplina de programação por meio de uma ferramenta desenvolvida pelos pesquisadores.

**Metodologia:** Foi desenvolvida uma ferramenta pelos pesquisadores que combinou dois serviços: coleta e testes automáticos dos programas submetidos pelos estudantes (semelhante a juízes online e o The Huxley) e interface de visualização dos dados para acompanhamento das atividades dos alunos (TSTView) que oferece tanto uma análise e visão dos dados de forma geral quanto voltada para a individualidade de cada aluno, permitindo assim que nenhuma das visões seja excluída. O TSTView possui dois módulos principais de funcionamento: o módulo do professor e o módulo do aluno, onde o primeiro apresenta dados dos alunos realizando alguma atividade, no momento da consulta, e o segundo que apresenta esses dados/informações de forma individual em um formato de relatório histórico (visto que ao aluno, só importa seu próprio relatório). O login de cada módulo se deu por meio do google app engine, onde o sistema executa, o que traz segurança e desempenho para o projeto. A ferramenta utiliza de diversos dados e de uma espécie de sumário para auxiliar os alunos e professores a entenderem o que estão visualizando, fazendo com que seja mais acessível e prática a leitura do que está sendo mostrado. A ferramenta foi utilizada na disciplina de programação 1 do 2º semestre de Ciência da Computação da UFCG em 2012, foi-se utilizado 105 alunos, dentre esses novatos e repetentes, divididos em 5 turmas práticas nesse estudo. A lista de exercícios continha 258 questões e recebeu um total de 31.908 submissões de alunos. Das 258 questões 68 delas representaram mini-testes presenciais aplicados para contabilização de nota para a disciplina (média de 13,6 questões por turma). Durante os mini-testes, os alunos utilizaram um ambiente protegido que permitia o acesso apenas ao enunciado, ao serviço de submissão de respostas e aos resultados de teste. Com base no que se ia percebendo, ia ocorrendo intervenções para o melhor aproveitamento dos alunos com relação a disciplina.

**Fichamento:** O artigo apresenta não só a ferramenta desenvolvida, mas também a discussão sobre a relevância da mesma durante o semestre da disciplina de programação que foi utilizada. A ferramenta TSTView desenvolvida pelos pesquisadores parece ser uma ótima ferramenta para utilizar no estudo de learning analytics aplicado a programação. A ferramenta permite observar diversas possibilidades de cenários possíveis quando questões são submetidas, algumas por exemplo seriam: constatar uma má elaboração da questão/casos de teste/habilidades exigidas com base em um amplo reflexo no desempenho da turma, ou então notar dificuldade de um aluno específico com base nas submissões erradas do mesmo, ou ainda o oposto, também com base em suas submissões, entre outras coisas. O trabalho tem um aspecto um pouco mais distinto do buscado no nosso pela questão de ir gerando os dados à medida que foi avançando, porém, alguns princípios são muito bem-vindos. O trabalho se mostra bastante detalhista e embora pareça um pouco extenso, apresenta uma certa tranquilidade de absorvê-lo e aplicar os princípios em nosso próprio trabalho. Ele tem como referencial teórico os seguintes:

Ala-Mutka, K. M. (2005). A survey of automated assessment approaches for programming assignments. Computer Science Education, 15(2):83–102.

Aureliano, V. C. O. and Tedesco, P. C. d. A. R. (2012). Ensino-aprendizagem de programação para iniciantes: uma revisão sistemática da literatura focada no sbie e wie. In Anais do Simposio Brasileiro de Informática na Educação, volume 23.

Borges, M. A. F. (2000). Avaliação de uma Metodologia Alternativa para a Aprendizagem de Programação. In Workshop de Educação em Computação, Congresso anual da SBC 2000, Curitiba, Brasil. SBC.

Chamillard, A. T. and Braun, K. A. (2000). Evaluating programming ability in an introductory computer science course. In Proceedings of the thirty-first SIGCSE technical symposium on Computer science education, SIGCSE ’00, pages 212–216, New York, NY, USA.

ACM. de Araujo, E. C., Gaudencio, M., Menezes, A., Ferreira, I., Ribeiro, I., Fagner, A., Ponciano, L., Morais, F., Guerrero, D. S., and Figueiredo, J. A. (2013). O papel do habito de estudo no desempenho do aluno de programação. In Workshop de Educação em Computação, Congresso anual da SBC 2013, Maceio, Brasil. SBC.

Fangohr, H. (2004). A comparison of c, matlab, and python as teaching languages in engineering. In Bubak, M., Albada, G., Sloot, P., and Dongarra, J., editors, Computational Science - ICCS 2004, volume 3039 of Lecture Notes in Computer Science, pages 1210–1217. Springer Berlin Heidelberg.

Ihantola, P., Ahoniemi, T., Karavirta, V., and Seppala, O. (2010). Review of recent systems for automatic assessment of programming assignments. In Proceedings of the 10th Koli Calling International Conference on Computing Education Research, Koli Calling ’10, pages 86–93, New York, NY, USA. ACM.

Korhonen, A., Malmi, L., Myllyselka, P., and Scheinin, P. (2002). Does it make a difference if students exercise on the web or in the classroom? In Proceedings of the 7th annual conference on Innovation and technology in computer science education, ITiCSE ’02, pages 121–124, New York, NY, USA. ACM.

**Resultados/Observações:** A medida que o projeto ia avançando foi-se percebendo dificuldades na turma de várias naturezas através da ferramenta e elas foram sendo trabalhadas para buscar um melhor resultado para a turma. É apontado que a ferramenta se mostrou eficiente para o acompanhamento das atividades e da produção dos alunos e é relatado que a experiência em sala de aula se mostrou positiva e auxiliou em diversos aspectos dessa questão (o ensino de programação e sua otimização). A utilização do TSTView durante as atividades em sala de aula permite ao mediador observar a turma de uma posição privilegiada, podendo ter acesso as submissões dos alunos para as atividades planejadas e vendo se elas estão corretas ou não de acordo com os testes automáticos, e possibilitando uma intervenção positiva e até precoce para o aluno ou grupo com dificuldade na mesma hora. A ferramenta possibilita também uma análise mais individual dos alunos e mantém um histórico dele na ferramenta, podendo identificar comportamentos de risco como pouca realização de atividades ou até mesmo os realizando apenas em sala de aula, entre outros. A utilização da ferramenta possibilitou que o professor e os alunos encontrassem um ponto de concordância para que ambas as partes se adequassem a um ritmo de produção que fosse bom para todos.

**Áreas abertas**: Implementar na ferramenta a criação de rankings para promover entre os alunos formas de compararem seus códigos e se desenvolverem em grupo e adicionar técnicas de mineração de dados a ferramenta para que possam indicar exercícios aos alunos quando não for possível a intervenção de um mediador para sugerir esses exercícios para ele.