# Juiz online como ferramenta de apoio a uma metodologia híbrida de ensino de programação

**Ano:** 2016

**Relação com o trabalho a ser proposto**: A utilização de uma ferramenta (juiz online) na pesquisa para medir o desempenho e entender se é positivo a adição de algo do gênero para os resultados dos alunos, que no caso do nosso trabalho, seria o The Huxley (que também é apontado nesse trabalho). Similar aos artigos já apontados eu sei o que[...] e learning [...].

**Resumo:** A alta reprovação em disciplina introdutória de programação para estudantes não ligados à área de computação levou um grupo de professores a adotar uma metodologia híbrida de ensino. Ela mesclava aulas presenciais com atividades baseadas em um juiz online desenvolvido por um dos autores. A ferramenta possibilitou que os estudantes praticassem mais exercícios de programação, com feedback imediato, proporcionando um aumento na taxa de aprovação. A percepção dos alunos, de forma geral positiva, também foi analisada, por meio do Método de Explicitação do Discurso Subjacente.

**Palavras-chave:** [não apontadas pelo autor] Computação; Juiz online; feedback imediato; metodologia híbrida de ensino

**Problema de pesquisa/proposta**: A alta reprovação em disciplina introdutória de programação e como buscar resolver esse problema através de um sistema de ensino híbrido com a utilização uma ferramenta de feedback aos alunos.

**Metodologia:** O trabalho consiste em uma metodologia híbrida que mescla o ensino presencial com o ensino virtual Essa metodologia foi aplicada em três de oito turmas de introdução a programação composta de calouros (as 3 turmas onde foi aplicado é o grupo experimental e as 5 turmas que não foi aplicado são o grupo de controle) e foi medido em ambos os grupos as taxas de aprovação, reprovação por nota e reprovação por frequência, onde esses valores somam 100% visto que um aluno não pode trancar a matrícula durante os dois primeiros períodos letivos. Essa metodologia utiliza 3 ferramentas principais: O juiz online CodeBench desenvolvido por um dos autores do artigo, que era utilizado para apoiar nas atividades do laboratório de codificação e avaliações parciais, fazendo por meio dele que os estudantes exercitassem as habilidades de programação; um ambiente virtual de aprendizado chamado colabweb, instância do ambiente moodle, que servia para apoiar questões teóricas e a avaliação final, fazendo por meio dele que os alunos realizassem exercícios conceituais voltados a habilidade de rastreamento de código e usado também para divulgar informações complementares sobre a disciplina; IDE Spyder, uma IDE para a linguagem Python, que é a principal linguagem utilizada para introdução a programação devido seu fácil entendimento em relação as demais linguagens de programação. No grupo experimental, foi criado uma divisão de equipes segundo 3 papéis: Professor, o docente da instituição que era responsável por preparar e ministrar as aulas de abertura dos módulos e elaborar questões do laboratório e das avaliações parciais; Tutor, que era responsável por tirar as dúvidas dos estudantes durante os laboratórios e auxiliar o professor na condução das avaliações práticas no juiz online; Gerente do juiz online, que era o responsável por cadastrar os laboratórios e as avaliações no juiz online, elaborar os gabaritos para todas as questões e os respectivos casos de teste. Foram colocados um professor e um tutor para cada uma das três turmas e o juiz online foi o mesmo para todas. Foi realizado um balanceamento entre aulas presenciais e aulas de cunho facultativo, levando em conta uma divisão de módulo introdutório e módulos temáticos feita para a disciplina e seus conteúdos. Foi atribuído notas aos laboratórios de codificação e de exercícios para estimular a realização deles entre os alunos, embora o maior peso fosse das avaliações parciais que encerravam o módulo na hora de obter a nota final. Foi então atribuído pesos distintos com base na dificuldade dos assuntos para motivar dois perfis de aluno: os que tiveram dificuldade no início da disciplina e os que tiveram facilidade no início, buscando fazer com que os primeiros recuperassem nota e os segundos mantivessem o bom desempenho nos estudos. Ao final da disciplina foi solicitado que os alunos das turmas experimentais avaliassem a metodologia por meio do preenchimento de um questionário que consistia de perguntas sobre o entendimento e dificuldade de realizar as atividades propostas assim como a opinião do aluno sobre a metodologia aplicada. Com os dados levantados aplicou-se uma análise qualitativa sobre os comentários dos participantes nas perguntas abertas, por meio do método de explicitação do discurso subjacente (MEDS).

**Fichamento:** O trabalho apresenta uma abordagem híbrida de ensino-aprendizagem que consiste em fornecer uma boa quantidade de prática ao aluno e feedback rápido sobre suas soluções para as questões solicitadas para que essas possam ser melhor ajustadas quando detectadas falhas ou erros, a fim de evitar que o aluno se frustre na tentativa de aprender os princípios da programação sem obter sucesso. Para isso foi pensada e aplicada uma reformulação na abordagem anterior aplicada na matéria, de modo a possibilitar a entrada da automatização da correção de exercícios (juiz online). O trabalho detalha bastante cada aspecto envolvido no processo desse trabalho, o que pode ser um pouco “desnecessário” para nossa visão que estamos mais focados nos resultados, por já termos uma base do assunto e não precisarmos de tanta introdução ao assunto em si, mas é um ótimo conteúdo para quem não está familiarizado com o tema e os termos abordados (IDE, Juiz online, entre outros) e até mesmo pra reforçarmos nosso conhecimento do assunto, embora exista outros artigos dos lidos que seja mais “direto”. O trabalho descreve também o funcionamento do CodeBench, o juiz online utilizado nesse estudo e como ele foi uma útil adição na metodologia da disciplina. O trabalho aponta como referencias os seguintes trabalhos (alguns em comum com referências de trabalhos já fichados):

Alves, F. P.; Jaques, P. (2014). Um Ambiente Virtual com Feedback Personalizado para Apoio a Disciplinas de Programação. In Anais do XXV Simpósio Brasileiro de Informática na Educação (Vol. 25, No. 1, p. 1078).

Chaves; J. O. M.; Castro; A. F.; Lima; R. W.; Lima; M. V. A.; Ferreira; K. H. (2013). Integrando Moodle e Juízes Online no Apoio a Atividades de Programação. In Anais do Simpósio Brasileiro de Informática na Educação (Vol. 24, No. 1, p. 244).

Giraffa, L.; Muller, L.; Moraes, M. C. (2015). Ensinando Programação apoiada por um ambiente virtual e exercícios associados a cotidiano dos alunos: compartilhando alternativas e lições aprendidas. In Anais dos Workshops do Congresso Brasileiro de Informática na Educação (Vol. 4, No. 1, p. 1330).

Hazzan, O.; Lapidot, T.; Ragonis, N. (2014). Guide to teaching computer science: an activity-based approach, 2ed. Springer.

Horn, M. B.; Staker, H. (2015). Blended: usando a inovação disruptiva para aprimorar a educação. Porto Alegre: Penso.

Ihantola, P.; Ahoniemi, T.; Karavirta, V.; Seppälä, O. (2010). Review of recent systems for automatic assessment of programming assignments. In Proc. of the 10th Koli Calling International Conference on Computing Education Research (pp. 86-93).

Nicolaci-da-Costa, A. M. (2007) O Campo da Pesquisa Qualitativa e o Método da Explicitação do Discurso Subjacente (MEDS). In: Psicologia: Reflexão e Crítica. vol.20 no.1. ISSN: 0102-7972. RS, Porto Alegre.

Paes, R.B.; Malaquias, R.; Guimarães, M.; Almeida, H. (2013). Ferramenta para a Avaliação de Aprendizado de Alunos em Programação de Computadores. In Anais dos Workshops do Congresso Brasileiro de Informática na Educação (Vol. 2, No. 1).

Pelz, F. D.; Jesus, E. A.; Raabe, A. L. (2012). Um Mecanismo para Correção Automática de Exercícios Práticos de Programação Introdutória. In Anais do XXIII Simpósio Brasileiro de Informática na Educação (Vol. 23, No. 1).

Píccolo, H. L.; Sena, V. F.; Nogueira, K. B.; Silva, M. O.; Maia; Y. A. N. (2010). Ambiente Interativo e Adaptável para ensino de Programação. In Workshop sobre Educação em Computação, pp. 555–566.

**Resultados/Observações:** Ele levanta gráficos acerca dos dados colhidos sobre aprovação, reprovação por faltas e reprovação por nota entre os anos de 2010 e 2015 (ano esse onde foi aplicado a metodologia), que permite notar que em relação ao ano de 2014, o grupo experimental teve um maior aumento na porcentagem de aprovados, enquanto o grupo de controle teve um aumento quase que nulo dessa porcentagem de aprovados em relação ao seu ano anterior. É possível observar também que a porcentagem de alunos reprovados por frequência aumentou pouco no grupo experimental em relação ao ano anterior (2014) e teve um aumento bem maior no grupo de controle em relação ao ano anterior, e ainda que em ambos os grupos essa porcentagem tenha crescido, é notável que no grupo experimental esse aumento foi muito menor em relação ao aumento do grupo de controle, que estava utilizando o método tradicional de ensino. Esses dados corroboram com a ideia de que a metodologia híbrida aumentou significativamente a quantidade de alunos aprovados na disciplina. É apresentado também a partir da análise qualitativa segundo o método MEDS as respostas dos estudantes sobre o questionário no fim da disciplina no grupo experimental, onde apresenta em uma tabela os pontos em comum percebidos em cada resposta dos alunos em relação a ferramenta utilizada, tais como feedback rápido, qualidade do material de apoio, estudo autônomo, entre outros pontos positivos. É apresentado também a transcrição de alguns comentários levantados pelos alunos, que de forma resumida mostram a importância do feedback rápido e da facilidade de uso da ferramenta, assim como aspectos que podem ser melhorados, como apontaram alguns alunos, a falta de mais exemplos para tester o programa, embora outros não tivessem essa tal questão. Foi apontado pelos alunos também que o método de correção do CodeBench podia ser melhorado, visto que ele funciona como “tudo ou nada”, ou acertou a questão ou a errou, sem considerar meio termo ou pequenos erros no código como “pequenas infrações” na hora de fornecer a nota final da questão. Destacou-se também a importância e relevância do contato com os professores e tutores ao longo da disciplina, a flexibilidade de tempo e lugar, estudo autônomo e melhor divisão de tempo entre as matérias do curso, respeitando o ritmo de cada estudante.

**Áreas abertas**: Aprimoramento do CodeBench para tornar mais intuitivo e de fácil acesso a sua utilização por professores e alunos, assim como aprimorar suas metodologias de correção de questões entre outras coisas.