

Tecnologia de apoio ao ensino e aprendizagem de programação em graduações tecnológicas profissionais: Juiz *On-line*

Supporting technology to the teaching and learning of programming in professional technology degrees: Juiz *On-line*

Tecnología para apoyar la enseñanza y el aprendizaje de la programación en grados de tecnología profesional: Juez en Línea

Carlos Vital Giordano¹

Lucio Nunes de Lira²

Celi Langhi³

Marcelo Duduchi Feitosa⁴

¹ Professor do Programa de Mestrado em Gestão e Desenvolvimento da Educação Profissional do Centro Paula Souza – Campus São Paulo. São Paulo, SP, Brasil. Orcid: <http://orcid.org/0000-0002-5557-9529>.

² Mestre do Programa de Mestrado em Gestão e Desenvolvimento da Educação Profissional do Centro Paula Souza – Campus São Paulo. São Paulo, SP, Brasil. Orcid: <http://orcid.org/0000-0003-3478-2576>.

³ Professora do Programa de Mestrado em Gestão e Desenvolvimento da Educação Profissional do Centro Paula Souza – Campus São Paulo. São Paulo, SP, Brasil. Orcid: <http://orcid.org/0000-0002-5527-2412>.

⁴ Professor do Programa de Mestrado em Sistemas Produtivos do Centro Paula Souza – Campus São Paulo. São Paulo, SP, Brasil. Orcid: <http://orcid.org/0000-0002-0969-4737>.

Recebido para publicação em: 30.9.2021

Aprovado em: 25.10.2021

Resumo

As disciplinas de programação de computadores se posicionam entre as com maior índice de reprovação em graduações em computação. Porém, existem tecnologias com potencial para apoiar o processo de ensino e aprendizagem, permitindo aos discentes estudar extraclasse com uma abordagem diferente da tradicional, de aulas expositivas, tendo em seguida um *feedback* imediato por meio de resolução de problemas e avaliação automática das soluções construídas. A tecnologia Juiz *On-line* possui características fundamentadas na abordagem de ensino comportamental. A presente investigação objetiva analisar e discutir características da tecnologia, conectando-a com a abordagem comportamentalista, e propor sua aplicação em cursos superiores tecnológicos de computação.

Palavras-chave: método; programação; algoritmos; comportamento.

Abstract

Computer programming subjects are situated among the subjects with a higher failure rate in computer science courses. However, there are technologies with the potential to support the teaching and learning process, allowing students to carry out extracurricular studies with a different approach from the traditional, expository classes, in which they receive an immediate feedback by means of problem-solving and automatic evaluation of the prepared solutions. The Juiz On-line technology has its features based on the behavioral teaching approach. This research aims at analyzing and discussing the features of such technology, connecting it with the behavioral approach, and proposing its application in technological undergraduate courses in computing.

Keywords: method; programming; algorithms; behavior.

Resumen

Las disciplinas de la programación informática se posicionan entre las que tienen mayor tasa de fallas en los grados en computación. Sin embargo, hay tecnologías con potencial para apoyar el proceso de enseñanza y aprendizaje, lo que permite a los estudiantes estudiar fuera de clase con un enfoque diferente del tradicional, de clases expositivas, teniendo un feedback inmediato a través de la solución de problemas y la evaluación automática de las soluciones construidas. La tecnología del Juez en Línea tiene características basadas en el enfoque de enseñanza conductual. Esta investigación tiene como objetivo analizar y discutir las características de la tecnología, conectándola con el enfoque conductista, y proponer su aplicación en carreras superiores tecnológicas de computación.

Palabras clave: método; programación; algoritmos; comportamiento.

Introdução

Tidas como essenciais nos cursos superiores da área de computação, as disciplinas de programação têm relevância reconhecida (ROBINS; ROUNTREE; ROUNTREE, 2003; SANTOS; COSTA, 2006; PEARS *et al.*, 2007), sendo ponto chave na estruturação do pensamento algorítmico. Inicialmente, o contato dos discentes com a programação de computadores ocorre nos primeiros semestres dos cursos tecnológicos profissionais, com aulas intercaladas ou integradas às disciplinas de introdução à lógica.

De acordo com Jesus e Brito (2009), Pereira (2010), Forbellone e Eberspacher (2005) e Valentim (2009), para que a natureza das disciplinas de programação seja entendida, é preciso conceituar pelo menos três termos comuns no contexto da computação:

- a. Algoritmo – procedimento que objetiva resolver um problema por meio da definição de uma sequência finita e logicamente ordenada de passos executáveis;

- b. Lógica de programação – técnica para desenvolver algoritmos considerando certas regras baseadas em lógica matemática e outras teorias básicas da ciência da computação, cuja finalidade reside na conversão do algoritmo em sua implementação em linguagem de programação;
- c. Linguagem de programação – conjunto de regras e comandos com sintaxe e semântica próprias que pode ser usado para traduzir algoritmos, feitos com base em lógica de programação, para algo executável por um computador.

Disciplinas diretamente relacionadas a algoritmos e programação de computadores pertencem ao grupo com maior taxa de reprovação nos cursos de computação (PIMENTEL; OMAR, 2008; BOSSE; GEROSA, 2015).

Um sistema de avaliação automática em que a verificação é centrada nas saídas que os programas submetidos geram, conforme as entradas fornecidas

Existem inúmeras razões que justificam as dificuldades dos discentes, como a exigência de pré-requisitos que comumente os ingressantes não trazem consigo, ou que ainda não estão suficientemente apurados, como raciocínio lógico (ALVES; JAQUES, 2014; RIBEIRO *et al.*, 2020). Também se ressalta, de acordo com Pereira (2010), a demora para que os alunos entendam a natureza dinâmica de um algoritmo, sobretudo quando implementados em uma linguagem de programação. Uma dificuldade a mais é gerada com a tradução dos algoritmos para uma linguagem não habitual (GOMES; HENRIQUES; MENDES, 2008).

Com o intuito de apoiar o processo de ensino e aprendizagem de programação de computadores, existem tecnologias que podem ser aplicadas antes, durante e após as aulas. Tais recursos se alicerçam em fundamentos pedagógicos de um processo de ensino comportamentalista, em contraste com o processo de ensino tradicional, cujas aulas predominam na exposição de conteúdos e em que a motivação para resolução dos exercícios pelos discentes é extrínseca, dependendo de características do docente para manter o aluno interessado (MIZUKAMI, 1986).

Apoiada por Souza (2017), a tecnologia a que esta investigação se refere, pertence ao grupo dos sistemas de avaliação automática, cujo objetivo reside em automatizar o processo de verificação de características particulares dos programas submetidos, considerando aspectos que cada ferramenta estabelece como pertinentes. Mais especificamente, os *softwares* abordados são denominados Juízes *On-line*.

Nesse contexto, um Juiz *On-line* se caracteriza, essencialmente, como um sistema de avaliação automática em que a verificação é centrada nas saídas que os programas submetidos geram, conforme as entradas fornecidas. Portanto, a validação da codificação se dá por análise empírica, com entradas pertencentes a um conjunto reconhecidamente válido e com as saídas geradas pelo programa submetido, confrontadas com saídas também reconhecidamente corretas.

A investigação tem como objetivo analisar determinadas características da tecnologia Juiz *On-line*, conectando-a com a abordagem de ensino comportamentalista, e também propor a sua aplicação em cursos de ensino superior tecnológicos nas

disciplinas diretamente relacionadas a algoritmos e programação de computadores, visando melhorias no processo de ensino e aprendizagem.

Referencial teórico

Os cursos de ensino profissional tecnológicos na área de computação “apresentam a disciplina de Algoritmos como base conceitual inicial, imprescindível e obrigatória. As definições de algoritmos variam de autor para autor, mas seu fundamento principal nunca muda: receber entradas, processá-las e produzir saídas” (VIEIRA; LIMA JUNIOR; VIEIRA, 2015, p. 6).

Os métodos de ensino tradicionais não são perfeitamente adequados às necessidades de todos os estudantes

Gomes e Mendes (2007) apontam o ensino e a aprendizagem de programação de computadores como um desafio tanto para docentes quanto para alunos. Os níveis de fracasso nas disciplinas introdutórias, em qualquer sistema de ensino e em qualquer parte do mundo, são preocupantes, e a temática aparece em diversas pesquisas (ROBINS; ROUNTREE; ROUNTREE, 2003; PEARS *et al.*, 2007; PIMENTEL; OMAR, 2008; JESUS; BRITO; 2009; ALVES; JAQUES, 2014; BOSSE; GEROSA, 2015; MACHADO *et al.*, 2018; PEREIRA *et al.*, 2019), resultando em muitas propostas de aplicação e resultados diferentes.

O ensino de programação objetiva que os discentes desenvolvam suas capacidades, com aquisição de conhecimentos e competências necessários para criar programas e sistemas computacionais que solucionem problemas reais (GOMES, 2007). É importante que os alunos aprendam a como abordar sistematicamente os problemas para resolvê-los com algoritmos e programação, e isso deve ser a meta das disciplinas introdutórias. Porém, o que costuma acontecer difere dessas premissas, gerando dificuldades de aprendizagem relativas a conceitos abstratos e noções básicas (GOMES; MENDES, 2007).

Como ressaltado por Gomes (2010), os métodos de ensino tradicionais não são perfeitamente adequados às necessidades de todos os estudantes, tendo como razões para essas dificuldades:

- a. O ensino não é personalizado – o ideal seria um docente com disponibilidade integral para supervisionar cada estudante, fornecendo um atendimento individualizado. Atenção especial a conceitos não entendidos e respostas imediatas durante às resoluções de exercícios provavelmente mitigariam deficiências futuras decorrentes da não assimilação do conteúdo atual. Porém, por limitações de tempo e custos, é pouco viável;
- b. As estratégias de ensino não apoiam todos os estilos de aprendizagem dos discentes – como as pessoas aprendem de modos diferentes e apresentam preferências próprias pela forma como algo é ensinado, há um paradoxo em relação à educação tradicional, que impõe que todos os estudantes devem aprender de forma homogênea (GOMES; MENDES, 2007);

- c. O ensino de conceitos dinâmicos por meio de materiais estáticos – programação envolve conceitos dinâmicos e que muitas vezes têm suas explicações depositadas em representações estáticas, variando entre projeções de *slides*, explicações verbais, diagramas etc. Isso pode ser insuficiente para os alunos;
- d. Alguns docentes dão foco para o ensino de linguagens de programação e seus detalhes de sintaxe. No entanto, o ideal seria impulsionar o desenvolvimento de soluções de problemas envolvendo as linguagens – Gomes (2010) discorre sobre o propósito das disciplinas introdutórias de algoritmos e programação de computadores, que deveriam buscar incrementar as habilidades dos discentes em resolver problemas de forma sistemática e lógica, sem atentar demasiadamente para detalhes sintáticos da linguagem de programação usada.

Falkembach e colaboradores (2003) alertam para a diferença de resolver problemas de forma tradicional, sem o uso de tecnologias computadorizadas, e com o uso de computação. Segundo os autores, a resolução de problemas sem o apoio de recursos tecnológicos induz o discente à abordagem tradicional ao solucionar um problema específico. Por outro lado, a utilização de tecnologias computadorizadas, conceitos de abstração, generalização e formalização passam a ser exigidos, e a consequente representação simbólica pode se tornar uma barreira para muitos dos alunos.

Como concluído por Bez, Tonin e Rodegheri (2014), embora exista tecnologia interativa, muitas aulas são planejadas considerando a resolução de exercícios apenas com a escrita de algoritmos e suas implementações em alguma linguagem de programação, em papel. No entanto, ferramentas que permitem avaliações automáticas, com respostas imediatas e em um ambiente centralizador de exercícios, elevam o potencial de atração para os discentes, além de reduzir o tempo gasto pelo professor com atividades periféricas, como correções de exercícios (ALVES *et al.*, 2019).

Ferramentas que permitem avaliações automáticas, com respostas imediatas, elevam o potencial de atração para os discentes

Conhecidos também como Juízes *On-line*, segundo Chaves *et al.* (2013) e Wasik *et al.* (2018), essas ferramentas de autocorreção executam um procedimento de avaliação automática, compilando o código-fonte enviado por um usuário, executando-o e testando-o com entradas e saídas específicas, com o objetivo de averiguar empiricamente o correto funcionamento do programa.

O programa, gerado a partir do código-fonte submetido à correção, recebe entradas específicas e padronizadas ocultas ao usuário. Após processar cada entrada, compara-se a saída correspondente do programa com aquela previamente armazenada no Juiz *On-line*, supostamente correta. Então, o sistema fornece uma resposta apropriada com base nessas comparações (certo, errado, erro de execução, erro de compilação etc.).

Ressalta-se que tal método se assemelha aos utilizados em concursos/competições de programação (não pertencentes ao escopo desta investigação).

Ihantola (2010) menciona que é possível fazer uma mescla entre o uso da tecnologia de Juízes *On-line* e correções mais seletivas, desempenhadas pelo docente. O intuito da presença do professor na análise provavelmente colaboraria com as respostas emitidas pelos sistemas, além de fornecer pareceres mais específicos e orientações personalizadas.

É possível fazer uma mescla entre o uso da tecnologia de Juízes *On-line* e correções mais seletivas, desempenhadas pelo docente

Para Bez, Ferreira e Tonin (2013) e Selivon, Bez e Tonin (2015), um exemplo de Juiz *On-line* é o URI Online Judge¹, que permite instalar um módulo destinado aos docentes para acompanhamento da prática e da evolução do desempenho dos alunos devidamente cadastrados no sistema. Entre outras funcionalidades, o URI Online Judge mantém público um placar com informações sobre a pontuação obtida pelos usuários para a resolução de problemas, além de medalhas virtuais concedidas conforme os objetivos são cumpridos, características que incentivam a competitividade entre os participantes e entre as instituições de ensino, uma vez que podem ser vinculadas aos usuários.

Chaves e colaboradores (2013) citam a existência de diversos sistemas com esse propósito, além de resumir o macroprocesso de seus funcionamentos. Segundo o autor, os Juízes *On-line* disponibilizam problemas variados para serem resolvidos por meio de algoritmos que serão implementados em uma das linguagens de programação suportadas.

Geralmente o procedimento consiste em:

- a. Selecionar o problema que se deseja resolver;
- b. Optar pela linguagem que será usada;
- c. Escrever a proposta de solução de acordo com as entradas e saídas esperadas e declaradas no enunciado;
- d. Submeter a proposta de solução, isto é, o código-fonte, ao sistema para que seja avaliado automaticamente;
- e. Aguardar a resposta do sistema, que indicará o sucesso ou a falha da submissão.

Conforme enfatizado por Chaves (2014), Juízes *On-line* são facilmente encontrados na internet, por exemplo: Timus Online Judge², CodeBech³, SPOJ Brasil⁴ e The Huxley⁵ (PAES *et al.*, 2013). Porém, há também sistemas executados localmente e/ou com tempo de funcionamento determinado, como o BOCA, citado por Campos e Ferreira (2004), usado sobretudo em competições de programação.

O comportamento e as premissas básicas de funcionamento dos Juízes *On-line* verificados neste trabalho se assemelham. No entanto, divergem em recursos disponibilizados, principalmente quanto a idioma, potenciais incentivos aos discentes,

apoio aos docentes no gerenciamento de disciplinas, acompanhamento de desempenhos, construção e tipos de problemas. Alguns demandam um procedimento mais complexo para a criação e publicação de problemas, exigindo homologação pela equipe detentora dos direitos administrativos do sistema, além de estimular um formato de enunciados menos objetivo, algo também característico de competições de programação (URI Online Judge, SPOJ etc.). Outros, como The Huxley, permitem a publicação de novos problemas sem a necessidade de verificação humana, com formatos aproximados aos dos exercícios de fixação.

Na abordagem comportamentalista de ensino, o conteúdo transmitido pelo docente busca criar competências nos indivíduos

Como relatado por Santos *et al.* (2020), o Juiz *On-line* incentiva o comportamento ativo dos alunos de procurar seus próprios erros ao submeter propostas de soluções para os problemas, contrastando com as características da abordagem tradicional de ensino, em que o aluno deve aguardar parecer do professor sobre os trabalhos entregues, algo raras vezes imediato.

Mizukami (1986) afirma que a abordagem tradicional de ensino é com frequência associada ao método expositivo de aulas, em que o professor transmite o conteúdo para os alunos, que, de maneira usual, colocam-se em modo passivo. Após a exposição do assunto pelo docente, o trabalho essencialmente ocorre pela execução de exercícios predominantemente de repetição. A avaliação de aprendizagem será

feita em momento posterior, “visando a exatidão da reprodução do conteúdo comunicado em sala de aula. Mede-se, portanto, pela quantidade e exatidão de informações que se consegue reproduzir” (MIZUKAMI, 1986, p. 17).

Uma característica da abordagem tradicional de ensino reside na suposição de que todos os alunos devem trabalhar no mesmo ritmo. O professor usará o mesmo o material didático para todos e ministrará a aula com o intuito de que todos tenham os mesmos conteúdos assimilados (MIZUKAMI, 1986). Porém, há barreiras para o atendimento individualizado.

Na abordagem comportamentalista de ensino, o conteúdo transmitido pelo docente busca criar competências nos indivíduos. O uso de tecnologias também auxilia no fornecimento de porções adequadas de conteúdos, retirando tarefas do docente, segundo Mizukami (1986), potencialmente repetitivas.

Ainda de acordo com a autora, qualquer estratégia para instruir alguém que se baseie na abordagem comportamentalista, obrigatoriamente se preocupa com a caracterização científica, seja no planejamento, na forma como será desenvolvida e/ou na avaliação do processo.

Para a análise comportamental, há verificação de estímulo-resposta. O indivíduo é conduzido a mudar o comportamento por meio de treinamento e com objetivos pré-determinados. Os comportamentos desejados devem ser estimulados (a princípio por um terceiro, como o docente) e, idealmente, pelo próprio aluno em fase posterior (MIZUKAMI, 1986).

Os Juízes *On-line* em geral possuem sistemas de recompensas e estímulos, por exemplo: placares, comparativos de desempenho e selos de conquistas (os *badgets*), o que, conforme Irion, Pelegrino e Botelho (2016) e Ribeiro *et al.* (2020), incentiva a competitividade e o estímulo de adequação a que se refere a abordagem comportamentalista de ensino. Assim, essa forma de estudo apresenta viés de personalização e individualidade, e não necessariamente de repetição exata de conteúdos ministrados em aula.

Método

A investigação adota abordagem descritiva exploratória e se baseia em revisão bibliográfica, pesquisa-ação e relato de caso.

Tais métodos são indicados, de acordo com Sampieri, Collado e Lucio (2013), Thiollent (2018) e Yoshida (2021), quando o caso em exame se configurar pertinente, não suficientemente examinado e revestido de certa característica de originalidade, em virtude da área de conhecimento, *locus*, tema e método de ensino-aprendizagem em que ocorre (apoio aos discentes em disciplinas específicas), e quando um dos pesquisadores coadjuva e se envolve de fato nos eventos como profissional tutor.

Resultados e discussão

Para aplicação de um sistema de avaliação automática do tipo Juiz *On-line*, entre as diversas opções já citadas, seleciona-se aquela mais adequada ao estilo do processo de ensino do docente, com o objetivo de acentuar a aprendizagem segundo critérios prévios.

Ressalta-se a vantagem de não haver necessidade de encontros presenciais entre discentes e o profissional dedicado à manutenção das atividades que envolvem o instrumento.

Os Juízes *On-line* em geral possuem sistemas de recompensas e estímulos, como placares, comparativos de desempenho e selos de conquistas

Pelas características da tecnologia Juiz *On-line*, a aprendizagem de algoritmos e sintaxes das linguagens de programação se potencializa, exigindo, contudo, uma transformação no processo de ensino. Aulas essencialmente pautadas na exposição de conteúdos deverão conter uma abordagem comportamentalista, favorecendo a interação e a descoberta pelos discente. As experiências são essenciais.

Algumas versões dos Juízes *On-line* se pautam em fornecer um ambiente voltado para competições entre os usuários. Com essa premissa, os alunos adquirem motivação intrínseca voltada para a superação de seus pares e com oportunidades de obter prêmios por apresentarem um desempenho superior. O papel do docente em estimular os usuários do sistema, com incentivos aos comportamentos desejados, deve ser contínuo e atencioso com relação ao controle de efeitos colaterais, mitigando a ocorrência de conflitos ou a alienação.

O objetivo maior reside na prática do que foi ministrado em aula, porém efetuada extraclasse e permitindo acompanhamento remoto.

Auxílios na resolução de dúvidas dos alunos devem ser primordialmente individualizados, favorecendo o ritmo de cada um. Uma forma de aplicação da tecnologia de Juízes *On-line* sugerida pelos autores desta investigação é:

- a. Determinar um profissional tutor para acompanhar os conteúdos e exercícios solicitados pelo docente responsável pela disciplina, para que os recursos permitidos e as exigências requeridas sejam conhecidos;
- b. Montar lista de problemas (nomenclatura comum em Juízes *On-line*) com conteúdos idênticos ou similares aos exercícios trabalhados na disciplina apoiada (variando conforme particularidades do sistema escolhido), considerando um intervalo de tempo que permita agrupamento de tópicos. Os agrupamentos de problemas podem combinar os de autoria própria com os previamente disponíveis no sistema, observando a pertinência;
- c. Definir a linguagem de programação padrão para as submissões de soluções, de modo que apenas aquelas que forem implementadas na linguagem definida sejam consideradas;
- d. Determinar o prazo pelo qual a lista de problemas estará disponível para ser concluída, com data e horário de início e término;
- e. Disponibilizar no sistema a lista de problemas, instruindo os discentes sobre os procedimentos para a utilização da tecnologia;
- f. Para acompanhamento a distância e interativo, durante o prazo determinado e em dias previamente combinados:
 - i. analisa-se as propostas de solução dos discentes para cada problema;
 - ii. envia-se, por meio eletrônico particular, apontamentos e sugestões sobre as soluções propostas;
 - iii. responde-se a dúvidas relacionadas à lista de problemas.
- a. Após a finalização do prazo, formula-se um parecer sobre cada participante, considerando seu desempenho no cumprimento da tarefa;
- b. Repete-se o procedimento enquanto houver, na disciplina apoiada, conteúdos inexplorados pelas listas de problemas.

Ressalta-se que o sistema de avaliação automática Juiz *On-line* proporciona *feedback imediato* para os estudantes. Após cada proposta de solução submetida aos casos de teste, mensagens com pareceres sobre o sucesso ou falha da implementação são exibidas, permitindo que correções sejam feitas e novamente colocadas para a avaliação da plataforma.

Em contrapartida, como o sistema não tem a capacidade de aconselhar os participantes e averiguar o uso correto de técnicas e recursos de programação, o profissional tutor assume o papel de, factualmente, corrigir as soluções propostas, além

de fornecer, o que se denomina na investigação, *feedback seletivo*, a fim de aconselhar acerca de práticas adequadas e autoquestionamentos, sem a limitação sistêmica da indicação binária de certo ou errado.

Considerações finais

Na investigação foram expostas e relacionadas algumas peculiaridades de sistemas de avaliação automática do tipo Juiz *On-line*, bem como as potenciais consequências de seu uso. Evidenciou-se a considerável variedade de plataformas e funcionamentos, notando-se divergências na concepção de uso, fator influenciador na prática docente e, conseqüentemente, no aprendizado discente.

Entre os aspectos abordados sobre a tecnologia em questão, um dos pontos de destaque residiu na facilidade de construção de problemas pelo próprio docente, com sistemas que exigem aprovação humana, por meio de administradores, e outros com processos automatizados de validação.

Distinguiram-se, também, pelos estilos de problemas, tendo exemplares com enunciados predominantemente semelhantes aos de competições de programação, em contraste com os sistemas que apresentam enfoque em enunciados análogos a exercícios de fixação.

A qualidade de apelo motivacional, com aspectos relacionados à competitividade, e o *feedback* contínuo e imediato estiveram presentes na discussão.

Com base na revisão bibliográfica, atuação efetiva e relato de caso, vislumbrou-se o potencial ganho no processo de ensino e aprendizagem de programação de computadores com o uso da tecnologia Juízes *On-line*, considerando suas características direcionadas à abordagem de ensino comportamentalista.

Por fim, para o aproveitamento mais prático, os autores elaboraram um instrumento orientador de como utilizar os sistemas em turmas de graduação tecnológica profissional na área de computação.

Notas

¹ Em 2021, o ambiente teve sua identidade alterada para beecrowd. Link oficial: <https://www.beecloud.com.br/>.

² Link oficial: <http://acm.timus.ru>.

³ Link oficial: <http://codebench.icomp.ufam.edu.br/>.

⁴ Link oficial: <https://br.spoj.com/>.

⁵ Link oficial: <https://www.thehuxley.com/>.

Referências

ALVES, Arthur Bindá *et al.* Análise comportamental em juízes online para predição do desempenho final de alunos em disciplinas de computação. *In: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO; 7.; SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 30., 2019, [s. l.]. Anais [...]. [s. l.: s. n.], 2019. p. 1906-1915. Disponível em: <http://www.br-ie.org/pub/index.php/sbie/article/view/8924/6477>. Acesso em: 26 out. 2021.*

ALVES, Fábio P.; JAQUES, Patrícia. Um ambiente virtual com feedback personalizado para apoio a disciplinas de programação. *In: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 3.; SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 25., 2014, [s. l.]. Anais [...]. [s. l.: s. n.], 2019. Disponível em: <https://www.br-ie.org/pub/index.php/sbie/article/download/3051/2715>. Acesso em: 8 nov. 2021.*

BEZ, Jean Luca; FERREIRA, C. E.; TONIN, Neilor A. URI Online Judge Academic: a tool for professors. *In: INTERNATIONAL CONFERENCE ON ADVANCED INFORMATION AND COMMUNICATION TECHNOLOGY FOR EDUCATION, 2013., [s. l.]. Proceedings [...]. [S. l.: s. n.], 2013.*

BEZ, Jean Luca; TONIN, Neilor A.; RODEGHERI, Paulo. R. URI Online Judge Academic: a tool for algorithms and programming classes. *In: INTERNATIONAL CONFERENCE ON COMPUTER SCIENCE & EDUCATION, 9., 2014. [s. l.]. Proceedings [...]. [S. l.: s. n.], 2014, p. 149-152.*

BOSSE, Yoram; GEROSA, Marco Aurelio. Reprovações e trancamentos nas disciplinas de introdução à programação da Universidade de São Paulo: um estudo preliminar. *In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO, 23., 2015. Porto Alegre. Anais [...]. Porto Alegre, Sociedade Brasileira de Computação, 2015. p. 426-435.*

CAMPOS, Cassio P.; FERREIRA, Carlos. E. BOCA: um sistema de apoio a competições de programação. *In: WORKSHOP DE EDUCAÇÃO EM COMPUTAÇÃO, 2004, [s. l.]. Anais [...]. [S. l.: s. n.], 2004. p. 885-895.*

CHAVES, José Oswaldo Mesquita. **Uma ferramenta de apoio ao processo de ensino-aprendizagem em disciplinas de programação de computadores por meio da integração dos Juízes Online ao Moodle.** Dissertação (Mestrado em Ciência da Computação) – Universidade do Estado do Rio Grande do Norte, Universidade Federal Rural do Semi-Árido, Mossoró, 2014. Disponível em: <https://ppgcc.ufersa.edu.br/wp-content/uploads/sites/42/2014/09/jose-osvaldo-mesquita-chaves.pdf>. Acesso em: 20 out. 2021.

CHAVES, José Osvaldo *et al.* Uma ferramenta baseada em juízes online para apoio às atividades de programação de computadores no Moodle. **RENOTE**, Porto Alegre, v. 11, n. 3, p. 1-10, 2013.

FALKEMBACH, Gilse Antoninha Morgental *et al.* Aprendizagem de algoritmos: uso da estratégia ascendente de resolução de problemas. *In*: TALLER INTERNACIONAL DE SOFTWARE EDUCATIVO, 8. 2003, Santiago. **Anais [...]**. Santiago: [s. n.], 2003.
FORBELLONE, Andre Luis Villar; EBERSPACHER, Henri Frederico. **Lógica de programação**: a construção de algoritmos e estruturas de dados. 3. ed. São Paulo: Prentice Hall, 2005.

GOMES, Anabela. **Dificuldades de aprendizagem de programação de computadores**: contributos para a sua compreensão e resolução. 2010. Tese (Doutorado em Engenharia Informática) – Universidade de Coimbra, Coimbra, 2010.

GOMES, Anabela; HENRIQUES, Joana; MENDES, António José. Uma proposta para ajudar alunos com dificuldades na aprendizagem inicial de programação de computadores. **Educação, Formação & Tecnologias**, [s. l.], v. 1, n. 1, p. [93-103], 2008.

GOMES, Anabela; MENDES, Antonio Jose. Learning to program-difficulties and solutions. *In*: INTERNATIONAL CONFERENCE ON ENGINEERING EDUCATION, 2007, [Coimbra]. **Proceedings [...]**. [Coimbra: s. n.], 2007, p. 1-6.

IHANTOLA, Petri *et al.* Review of recent systems for automatic assessment of programming assignments. *In*: KOLI CALLING INTERNATIONAL CONFERENCE ON COMPUTING EDUCATION RESEARCH, 10., 2010, [s. l.]. **Proceedings [...]**. [S. l.: s. n.], 2010. p. 86-93.

IRION, Crishna; PELEGRINO, Dâmi Henrique; BOTELHO, Marcus Paulo da Silva. A motivação através da competitividade: a busca pela qualidade da educação no estudo da computação. *In*: SIMPÓSIO EDUCACIONAL DE EDUCAÇÃO A DISTÂNCIA, 2016; ENCONTRO DE PESQUISADORES EM EDUCAÇÃO A DISTÂNCIA, 2016, São Carlos, SP. **Anais [...]**. São Carlos, SP: UFSCar, 2016. p. 1-9.

JESUS, Andreia de; BRITO, Gláucia Silva. Concepção de ensino-aprendizagem de algoritmos e programação de computadores: a prática docente. **Varia Scientia**, [s. l.], v. 9, n. 16, p. 149-158, 2009.

MACHADO, Leonardo Davi Pereira *et al.* Uma ferramenta colaborativa para apoiar a aprendizagem de programação de computadores. **Revista Brasileira de Computação Aplicada**, Passo Fundo, v. 10, n. 1, p. 23-29, 2018.

MIZUKAMI, Maria da Graça Nicoletti. **Ensino**: as abordagens do processo. São Paulo: Editora Pedagógica e Universitária, 1986.

PAES, Rodrigo de Barros *et al.* Ferramenta para a avaliação de aprendizado de alunos em programação de computadores. *In*: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 2., 2013, [s. l.]. **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**. [S. l.: s. n.], 2013.

PEARS, Arnold *et al.* A survey of literature on the teaching of introductory programming. **ACM SIGCSE Bulletin**, [s. l.], v. 39, n. 4, p. 204-223, dez. 2007.

PEREIRA, Filipe Dwan *et al.* Otimização e automação da predição precoce do desempenho de alunos que utilizam juízes online: uma abordagem com algoritmo genético. *In*: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO; 8., 2019; SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 30., 2019, [s. l.]. **Anais [...]**. [S. l.: s. n.], 2019. p. 1451-1460. Disponível em: <http://br-ie.org/pub/index.php/sbie/article/view/8877/6431>. Acesso em: 26 out. 2021.

PEREIRA, Silvio do Lago. **Algoritmos e lógica de programação em C**: uma abordagem didática. São Paulo: Érica, 2010.

PIMENTEL, Edson Pinheiro; OMAR, Nizam. Ensino de algoritmos baseado na aprendizagem significativa utilizando o ambiente de avaliação NetEdu. *In*: CONGRESSO DA SBC, 28., 2008; WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO, Belém, PA. **Anais [...]**. Belém: SBC, 2008.

RIBEIRO, Ralph Breno Silva *et al.* Investigação empírica sobre os efeitos da gamificação de um juiz online em uma disciplina de introdução à programação. **Revista Brasileira de Informática na Educação**, [Porto Alegre], v. 28, p. 461-490, jun. 2020. Disponível em: <https://www.br-ie.org/pub/index.php/rbie/article/view/v28p461/6725>. Acesso em: 26 out. 2021.

ROBINS, Anthony; ROUNTREE, Janet; ROUNTREE, Nathan. Learning and teaching programming: a review and discussion. **Computer Science Education**, [s. l.], v. 13, n. 2, p. 137-172, 2003.

SAMPIERI, Roberto Hernández; COLLADO, Carlos Fernández; LUCIO, Maria del Pilar Baptista. **Metodologia de pesquisa**. 5. ed. Porto Alegre: Penso, 2013.

SANTOS, Ingrid Lima dos *et al.* Tempos de transição em estados de corretude e erro como indicadores de desempenho em juízes online. *In*: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 31, 2020, [s. l.]. **Anais [...]**. Porto Alegre: Sociedade Brasileira de Computação, 2020. p. 1283-1292.

SANTOS, Rodrigo Pereira dos; COSTA, Heitor Augustus Xavier. Análise de metodologias e ambientes de ensino para algoritmos, estruturas de dados e programação aos iniciantes em computação e informática. **INFOCOMP**, [s. l.], v. 5, n. 1, p. 41-50, 2006.

SELIVON, Michele; BEZ, Jean Luca; TONIN, Neilor A. URI Online Judge Academic: integração e consolidação da ferramenta no processo de ensino/aprendizagem. *In*: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO, 23., 2015, Recife. **Anais [...]**. Porto Alegre: Sociedade Brasileira de Computação, 2015.

SOUZA, Draylson Micael de. **Subsídios à integração de ferramentas de avaliação automática e sistemas de gerenciamento de aprendizagem**. 2017. Tese (Doutorado em Ciências de Computação e Matemática Computacional) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Paulo, 2017.

THIOLLENT, Michel. **Metodologia pesquisa-ação**. Cortez: São Paulo, 2018.
VALENTIM, H. Um Estudo sobre o Ensino-aprendizagem de Lógica de Programação. *In*: ENCONTRO NACIONAL DE PESQUISA EM EDUCAÇÃO EM CIÊNCIAS, 7., 2009, [s. l.]. **Anais [...]**. [S. l.: s. n.], 2009.

VIEIRA, Carlos Eduardo Costa; LIMA JUNIOR, José Augusto Teixeira de; VIEIRA, Priscila de Paula. Dificuldades no processo de aprendizagem de algoritmos: uma análise dos resultados na disciplina de AL1 do Curso de Sistemas de Informação da FAETERJ - Campus Paracambi. **Cadernos UniFOA**, Volta Redonda, n. 27, p. 5-15, abr. 2015.

WASIK, Szymon *et al.* A survey on online judge systems and their applications. **ACM Computing Surveys (CSUR)**, [s. l.], v. 51, n. 1, p. 1-34, 2018.

YOSHIDA, Winston Bonetti. Redação do relato de caso. **Jornal Vascular Brasileiro**, [s. l.], v. 6, n. 2, p. 112-113. Disponível em: <https://www.scielo.br/j/jvb/a/vnKt5ttNpdFMj6dLcmnM4Q/?format=pdf&lang=pt#:~:text=A%20estrutura%20b%C3%A1sica%20do%20relato,complementam%20este%20tipo%20de%20publica%C3%A7%C3%A3o>. Acesso em: 26 out. 2021.