

SUBMETIDO 15/05/2021

APROVADO 01/09/2021

PUBLICADO ON-LINE 29/09/2021

PUBLICADO 30/03/2023

EDITORA ASSOCIADA
Crishane Azevedo Freire

DOI: <http://dx.doi.org/10.18265/1517-0306a2021id5942>

ARTIGO ORIGINAL

C073: ferramenta para apoio ao ensino de programação usando a metodologia de aprendizagem baseada em problemas

 Thiago Gouveia ^{[1]*}

 Kerven Maciel
Monteiro Albuquerque ^[2]

 Júlia Davet Oliveira ^[3]

 Valéria Maria Bezerra
Cavalcanti Maciel ^[4]

[1] thiago.gouveia@ifpb.edu.br

[2] kerven.albuquerque@gmail.com

[3] julia.davet.oliveira@gmail.com

[4] valeria.cavalcanti@ifpb.edu.br

Instituto Federal de Educação, Ciência e
Tecnologia da Paraíba (IFPB), Brasil

RESUMO: Atualmente, o ensino de disciplinas ligadas à Computação é fundamental nas mais diversas áreas do conhecimento. É possível observar, no entanto, que os estudantes apresentam muitas dificuldades, especialmente em Programação. Entre as tentativas para superá-las, podem-se destacar a utilização de competições de programação, estratégias de gamificação e Metodologias Ativas de Aprendizagem, como a Aprendizagem Baseada em Problemas (ABP). Este trabalho apresenta a ferramenta C073, desenvolvida para auxiliar tanto docentes quanto discentes na utilização da metodologia ABP em disciplinas de Programação de computadores. O C073 é um Juiz Online (JO), uma plataforma que permite aos usuários submeter programas para tentar solucionar um dos problemas do banco, corrigindo-os automaticamente. Um dos seus diferenciais é o módulo de cursos, no qual os docentes podem adicionar conteúdos e exercícios de sua autoria ou da própria plataforma. Além disso, o C073 foi concebido explicitamente como ferramenta de apoio à ABP e oferece opções eficazes para desafiar os problemas mais comuns enfrentados no processo de ensino e aprendizagem em Programação, propiciando um acompanhamento mais eficiente e próximo do desempenho dos estudantes, uma maior contextualização para os exercícios propostos e um *feedback* rápido e preciso.

Palavras-chave: aprendizagem baseada em problemas; ensino de programação; juiz on-line.

C073: Programming teaching tool using Problem-Based Learning methodology

ABSTRACT: Nowadays, teaching subjects related to Computer Science is fundamental for many areas of knowledge. However, it is possible to observe that students present many difficulties, especially with programming. Among the attempts to overcome these difficulties, using programming contests, gamification strategies, and Active Learning Methodologies, such as Problem-Based Learning (PBL), can be highlighted. This paper presents

*Autor para correspondência.

C073, a tool developed to assist both teachers and students in using PBL in Computer Programming courses. C073 is an online judge, a platform that allows users to submit code to solve problems from its database and automatically evaluates it. One of its differentials is the classes module, in which teachers can add content and exercises of their own or from the platform. In addition, C073 was conceived explicitly as a tool to support PBL, offering effective options to challenge the most common problems faced while teaching or learning Programming. It allows monitoring the student's performance more closely and efficiently, provides better contextualization for the problems, and offers fast and accurate feedback.

Keywords: *online judge; problem-based learning; programming teaching.*

1 Introdução

A evolução tecnológica que temos vivenciado nos últimos anos tem facilitado cada vez mais o acesso a computadores e *smartphones*, o que, junto ao fenômeno da democratização da Internet, tem tornado o mundo cada vez mais conectado. De fato, de acordo com o CGI.br (2020), 71% dos domicílios brasileiros possuem acesso à Internet, sendo que 99% das pessoas acessam a rede através do telefone celular, enquanto 42% utilizam o computador. Esse fenômeno abre espaço para o surgimento de novos serviços e tecnologias, como a computação nas nuvens, a Internet das Coisas e as casas e cidades inteligentes.

Nesse contexto, o ensino de disciplinas ligadas à Computação alcança um papel fundamental nos currículos das mais diversas áreas do conhecimento, dado o extenso ferramental que a informática provê para facilitação e/ou automatização de tarefas. Muitas vezes, contudo, explorar o máximo potencial desse conjunto de instrumentos requer um conhecimento básico sobre Lógica e Linguagens de Programação.

A título de exemplo, no campus João Pessoa do Instituto Federal da Paraíba (IFPB), em todos os cursos de nível médio e superior se oferta pelo menos uma disciplina ligada à informática; em aproximadamente 60% desses cursos há disciplinas ligadas à programação de computadores (IFPB, 2021). Vale ressaltar que essas disciplinas são ainda mais importantes em cursos da área da Computação, visto que seu conteúdo serve como base para boa parte das demais disciplinas trabalhadas.

Apesar de tal abrangência, o que lhes garante relevância, é possível observar que os estudantes apresentam muitas dificuldades em disciplinas da área de Programação. Diversos estudos abordam esse fenômeno, como os trabalhos de Jenkins (2002), Milne e Rowe (2002), Bergin e Reilly (2005), Bennedsen e Caspersen (2007), Gomes e Mendes (2007), Mow (2008), Bosse e Gerosa (2015), entre muitos outros. Esse fenômeno levanta algumas perguntas, dentre as quais se destaca a seguinte: quais motivos levam a tamanha dificuldade do processo de ensino e aprendizagem de Programação?

Na revisão sistemática de Souza, Batista e Barbosa (2016) são identificadas seis causas para o grande número de reprovações em disciplinas de Programação, todas relativas a dificuldades: (s.i) na apreensão de conceitos dessas disciplinas; (s.ii) na aplicação desses conceitos; (s.iii) na compreensão de códigos de programas; (s.iv) na divisão dos problemas em módulos; (s.v) na resolução dos problemas de programação, possivelmente por desinteresse já resultante das dificuldades anteriores; (s.vi) dos professores em ensinar

os conceitos de Programação, desenvolver materiais e exercícios de apoio, bem como avaliar os trabalhos da disciplina.

Por sua vez, Cadavid e Corcho (2018) agruparam as causas elencadas por diversos autores em quatro áreas: (a) metodologias de ensino; (b) métodos de estudo; (c) natureza da disciplina; e (d) efeitos psicológicos e sociais. O Quadro 1 estrutura alguns pontos apresentados pelos autores em cada área.

Quadro 1 ►

Dificuldades no processo de ensino-aprendizagem de disciplinas de Programação.

Fonte: elaborado pelos autores

Área	Problema
(a) 1	Turmas grandes
(a) 2	Ausência de personalização do ensino
(a) 3	Ausência de <i>feedback</i> para os alunos
(a) 4	Materiais estáticos
(a) 5	Ensino da linguagem de Programação em detrimento do de resolução de problemas
(b) 1	Metodologia de estudo inadequada
(b) 2	Dificuldade para resolução de problemas
(b) 3	Ausência do <i>background</i> necessário
(b) 4	Ausência de conhecimento prévio do assunto
(b) 5	Pouco tempo praticando (pouca dedicação)
(c) 1	Falta da necessária capacidade de abstração
(c) 2	Sintaxe complicada das linguagens
(d) 1	Estudantes desmotivados
(d) 2	Sentimento de anonimidade do estudante
(d) 3	Pouca ou nenhuma interação entre estudantes

De modo complementar, são adicionados aos elencados por Cadavid e Corcho (2018) alguns aspectos que foram julgados importantes pelos autores deste trabalho, postos no Quadro 2 a seguir.

Quadro 2 ►

Dificuldades adicionais no processo de ensino-aprendizagem de disciplinas de Programação.

Fonte: elaborado pelos autores

Área	Problema
(a) 6	Ausência de contextualização do conteúdo
(a) 7	Ausência de aplicabilidade do conteúdo
(b) 6	Excesso de atividades extracurso
(c) 3	Ausência de pensamento lógico
(d) 4	Falta de identificação com o curso
(d) 5	Estudante não possui papel ativo no processo ensino-aprendizagem

Observando que o estudante de hoje não deve ser comparado àqueles de dez ou vinte anos atrás, tendo em vista todas as mudanças sócio-históricas e tecnológicas já ocorridas ao longo do tempo, especialmente em relação à velocidade com que essas mudanças aconteceram, diversas estratégias têm sido utilizadas para tentar superar essas dificuldades, entre as quais podemos citar: competições de programação; gamificação;

Metodologias Ativas de Aprendizagem, como a sala de aula invertida, o *design thinking* e a Aprendizagem Baseada em Problemas (ABP). Neste trabalho, o interesse está, especialmente, no ensino de Programação por meio da ABP, metodologia de ensino e aprendizagem que se configura por utilizar problemas da vida real a fim de despertar maior interesse pelo assunto, ampliar o pensamento crítico e a capacidade de solução de problemas (RIBEIRO, 2008).

[1] Lê-se code, palavra inglesa que pode ser traduzida como código, uma referência ao código-fonte dos programas de computador.

Este artigo apresenta a ferramenta C073¹, desenvolvida com o objetivo de auxiliar tanto docentes quanto discentes na utilização da metodologia ABP em disciplinas de Programação de computadores. O C073 oferece opções eficazes para o enfrentamento dos problemas listados nos Quadros 1 e 2, assim como aqueles levantados por Souza, Batista e Barbosa (2016), propiciando ao professor um acompanhamento mais próximo do desempenho dos estudantes, maior contextualização dos exercícios propostos e um *feedback* rápido e preciso.

O restante deste artigo está organizado como segue: a seção 2 introduz o referencial teórico que norteou o desenvolvimento do C073; a seção 3 discute alguns trabalhos relacionados ao tema; a seção 4 aborda o processo de desenvolvimento da ferramenta, as tecnologias escolhidas e sua arquitetura; a seção 5 trata dos resultados obtidos, apresentando a versão final do C073, suas principais funcionalidades, além de fazer um paralelo direto com a metodologia ABP; por fim, a seção 6 traz as considerações finais sobre o trabalho realizado.

2 Referencial teórico

Esta seção apresenta os fundamentos teóricos que nortearam o desenvolvimento do C073. São discutidos conceitos como o modelo tradicional de ensino, as Metodologias Ativas de Ensino e a Aprendizagem Baseada em Problemas.

Segundo Mbodila e Muhandji (2012), o modelo tradicional de ensino, também referenciado como pedagogia tradicional, envolve o fluxo direto de informação do professor para o estudante, no qual o professor atua como portador do conhecimento enquanto o estudante atua apenas como recipiente. Ademais, algumas características marcantes dessa metodologia residem no uso de materiais prontos, prazos fixos e tarefas definidas unicamente pelo professor.

Ainda de acordo com Mbodila e Muhandji (2012), estas são algumas limitações inerentes à pedagogia tradicional:

- O processo de ensino e aprendizagem é concentrado em métodos teóricos em vez de aspectos práticos;
- Não há interação suficiente com os alunos em sala de aula;
- Há menos atividades na sala de aula, ambiente onde o professor decide o que fazer, quando e como;
- Não há criatividade, os estudantes apenas reproduzem o que o professor define;
- Há menos integração entre os estudantes. O professor é o único indivíduo que fala e comanda a aula.

Em contraponto ao modelo tradicional de ensino, as metodologias ativas, cujo âmago é a transformação do estudante em agente ativo do processo de ensino e aprendizagem em detrimento do papel único de receptáculo do saber, têm recebido atenção considerável do meio acadêmico nos últimos anos. As metodologias ativas (também referidas como estratégias ativas ou aprendizagem ativa) podem ser definidas como qualquer

método educacional capaz de engajar os estudantes no seu processo de aprendizagem, requerendo-lhes a realização de tarefas significativas e a reflexão sobre o que estão fazendo (PRINCE, 2004).

Segundo Bonwell e Eison (1991), as seguintes características são geralmente associadas às estratégias ativas de ensino e aprendizagem:

- Estudantes estão envolvidos em atividades que vão além de ouvir;
- A ênfase é dada ao desenvolvimento das habilidades dos estudantes, em detrimento da mera transferência de informação;
- Os estudantes estão envolvidos em atividades que requerem reflexão, como análise, síntese e avaliação;
- Estudantes estão engajados nas atividades, e.g., lendo, discutindo, escrevendo;
- Muita ênfase é dada à exploração das atitudes e valores dos próprios estudantes.

Diversas Metodologias Ativas de Aprendizagem têm sido aplicadas com sucesso em disciplinas das mais diversas áreas. Entre as estratégias mais bem sucedidas, podem ser citados a sala de aula invertida, o *design thinking*, a gamificação, a aprendizagem baseada em projetos e a aprendizagem baseada em problemas.

Neste trabalho, o interesse especial é na aplicação da ABP, que, segundo Prince (2004), é uma metodologia educacional na qual problemas relevantes são introduzidos no início do ciclo de aprendizado com o objetivo de prover os sujeitos de contexto e motivação. A ABP tem sido uma das metodologias de ensino mais adotadas entre todas as faixas etárias, tanto em cursos profissionalizantes quanto acadêmicos, e em todos os lugares do globo, sendo também uma das metodologias educacionais mais pesquisadas empiricamente (HUNG, 2009).

A origem da ABP remonta ao ensino da medicina nas décadas de 1960 e 1970, motivada pela explosão da quantidade de informação médica disponível, pelo surgimento de novas tecnologias e pela mudança rápida nas demandas futuras dessa prática (SAVERY, 2006). Vale observar que tais características se repetem hoje em relação à área da Computação em geral, que vive em revolução tecnológica constante e disponibiliza um grande conjunto de informações através da Internet.

Savery (2006) sintetiza as características essenciais da ABP, detalhadas por Barrows na página da PBL Initiative²:

- O estudante deve ser responsável pelo seu próprio aprendizado;
- Os problemas trabalhados devem ter defeitos estruturais, como metas difusas e informação incompleta, assemelhando-se aos problemas do mundo real;
- O aprendizado deve integrar diversas disciplinas e/ou assuntos;
- A colaboração é essencial;
- O aprendizado individual deve ser compartilhado com o grupo, de modo que tal aprendizado seja analisado novamente e utilizado para a resolução do problema;
- É essencial que haja discussão sobre o que foi aprendido com a resolução do problema, assim como sobre os conceitos e princípios que foram adquiridos;
- Autoavaliações e avaliações dos pares devem ser feitas ao final de cada problema e de cada unidade;
- As atividades desenvolvidas durante a ABP devem ter valor no mundo real;
- As avaliações devem aferir o progresso em direção aos objetivos da ABP;

[2] Segundo Savery (2015), Barrows descreve em detalhes os pontos essenciais da ABP em http://www.pbli.org/pbl/generic_pbl.htm. Essa página, no entanto, não está mais ativa no momento da escrita deste artigo.

- A ABP deve ser a base pedagógica do currículo em vez de uma de suas partes didáticas.

Alguns modelos foram propostos para implementação da ABP tomando por base as características acima citadas. Dentre estes, destacamos o *framework* 3C3R (HUNG, 2009), que pode ser descrito como um modelo conceitual e sistemático para o projeto de problemas efetivos e confiáveis para ABP. O 3C3R, cujo nome provém de *Content* (conteúdo), *Context* (contexto), *Connection* (conexão), *Researching* (investigação), *Reflecting* (reflexão) e *Reasoning* (raciocínio), introduz um processo com nove passos para o planejamento dos problemas a serem aplicados na ABP.

3 Trabalhos relacionados

Esta seção trata de trabalhos que se relacionam de forma direta com as propostas da ferramenta C073. Em um primeiro momento, são discutidos artigos que investigam a metodologia ABP no contexto da Computação, especialmente em disciplinas de Programação. Em sequência, são apresentadas algumas ferramentas que apresentam objetivos que se alinham aos do C073. Para uma visão mais geral sobre ABP aplicada a diferentes áreas educacionais, é sugerida a leitura do estado da arte levantado por Alves, Medeiros e Melo (2020), assim como dos artigos com revisões sistemáticas referenciados na introdução do mesmo trabalho.

3.1 ABP aplicada em Computação

Oliveira, Santos e Garcia (2013) realizaram um mapeamento sistemático para identificar estudos publicados entre 1997 e 2011 sobre as melhores práticas da ABP aplicadas às disciplinas da área da Computação. Foram levantados 2.464 artigos, dos quais foram selecionados os 52 mais relevantes. Vale ressaltar alguns dos resultados obtidos pelos autores: (o.i) a ABP é mais efetiva quando combinada com estratégias colaborativas e aplicações motivantes, como o desenvolvimento de jogos, competições ou problemas relacionados ao mercado; (o.ii) é muito importante que seja disponibilizado um ambiente rico, que ofereça diferentes modos para aquisição de conhecimento; (o.iii) a ABP aprimora nos estudantes a capacidade de desenvolvimento de habilidades. Por fim, os autores concluem que a ABP tem sido largamente aceita por estudantes e professores da área da Computação, assim como tem melhorado a educação ao oferecer uma visão mais ampla sobre os problemas reais da área.

Oliveira e Santos (2016) apresentaram o PBLMaestro, um ambiente virtual de ensino e aprendizagem projetado para dar suporte a todas as fases da implementação da ABP em disciplinas da área da Ciência da Computação. Foram incluídas estratégias de gamificação para amplificar o engajamento, a retenção e a motivação dos estudantes, paralelas ao envio de mensagens para um aplicativo para celular. Por fim, os autores conduziram entrevistas semiestruturadas, nas quais foi constatado um alto grau de satisfação de todos os envolvidos.

Em sua tese de doutorado, Rodrigues (2018) parte da premissa de que não é trivial implementar o processo de ensino com a metodologia ABP, visto que depende de esforços não apenas dos estudantes, tutores e professores, mas também de uma equipe pedagógica com papéis específicos. Diante do exposto, a autora propõe o *framework* conceitual *by-cycles*, com objetivo de sistematizar tanto a implementação quanto a gestão do processo de ensino-aprendizagem utilizando a ABP para o ensino em Computação. O *framework* foi aplicado em uma disciplina de um curso de graduação em Sistemas de Informação. Além do *by-cycles*, o trabalho traz uma rica discussão sobre a ABP em geral.

3.2 ABP aplicada em disciplinas de Programação

Kumar (2003) discute o uso de *problets* (sistemas tutores inteligentes) de resolução de problemas para o ensino-aprendizagem de conceitos de Programação. Esses *problets* são capazes de gerar problemas automaticamente, dar notas às atividades, enviar *feedback* imediato e detalhado sobre respostas corretas e determinar se o estudante apresentou resultados satisfatórios de aprendizagem com o material. De acordo com o autor, a ABP melhora a retenção de longo prazo e é melhor que o modelo tradicional com respeito ao desenvolvimento da habilidade de resolver problemas da vida real.

Os problemas gerados pelos *problets* oferecem aos estudantes dois tipos distintos de atividades: resolver (leia-se depurar) os problemas de um programa apresentado; ou prever o resultado – no caso, a saída – de um dado programa. A proposta foi avaliada em três experimentos. No primeiro, não controlado, verificou-se que o uso da ferramenta melhora de forma significativa o desempenho dos alunos. Em sequência, um grupo de controle utilizou apenas o livro-texto enquanto um segundo grupo utilizou os *problets*. O desempenho do grupo de controle foi menos proveitoso do que o do outro grupo. Por fim, comparando-se o uso de estratégias simples e detalhadas de *feedback*, constatou-se um maior aprendizado por parte daqueles que receberam *feedbacks* detalhados da ferramenta.

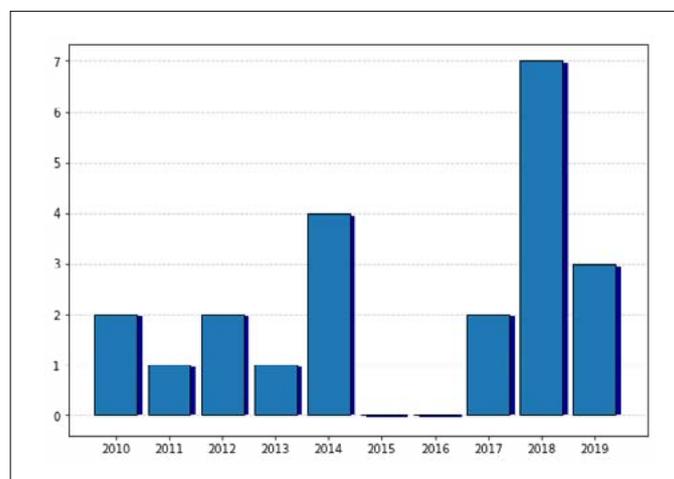
Costa e Rocha (2018) realizaram um mapeamento sistemático que levantou 39 trabalhos (454 antes de aplicar os critérios de exclusão) sobre o ensino de disciplinas de Programação sob a abordagem da ABP e a utilização de jogos como ferramenta para auxiliar esse processo. Os autores constataram que boa parte das iniciativas utiliza ambientes de apoio à submissão e avaliação automática de programas, em especial os Juízes Online (JO), também usados para apoiar os professores nas atividades de avaliação dos estudantes. Foram identificadas algumas lacunas importantes, principalmente no que diz respeito às fases de entendimento do problema, do planejamento da solução e do *feedback*, não sendo apresentados, por exemplo, *feedbacks* durante a fase de elaboração da solução.

Coelho e Guedes (2020) realizaram um mapeamento sistemático sobre ABP aplicada a disciplinas de Programação de Computadores, classificando-a como uma metodologia ativa de ensino capaz de elevar a capacidade de aprendizagem nessas disciplinas. Foram encontrados 316 estudos, dos quais 22 foram selecionados para uma análise mais detalhada. O gráfico da Figura 1 demonstra a distribuição dos artigos selecionados entre os anos de 2010 e 2019.

Figura 1 ►

Estudos primários entre 2010 e 2019.

Fonte: adaptada de Coelho e Guedes (2020)



Duas questões se destacam entre aquelas que o trabalho se propôs a responder:

- 1) Quais as vantagens e benefícios da aplicação da ABP no ensino de Programação?
- 2) Quais são os principais desafios e dificuldades para a utilização da ABP? Em relação à primeira pergunta, a contribuição de maior relevância da ABP foi (g.1) uma ampliação na capacidade de promover habilidades relacionadas à resolução de problemas. Outras contribuições realçadas foram (g.2) a capacidade para promover o trabalho em equipe, (g.3) o incentivo à autoaprendizagem, i.e., a aprendizagem autônoma, e (g.4) uma maior motivação propiciada pela metodologia. Por outro lado, os desafios considerados mais relevantes para o sucesso da ABP no ensino de Programação foram (g.5) a capacitação dos docentes e discentes, (g.6) a exigência de maior esforço de todos os atores envolvidos no processo e (g.7) o nível de qualidade e complexidade dos problemas utilizados.

3.3 Ferramentas com problemas de programação

Aplicar a ABP no contexto de disciplinas relacionadas à Programação de computadores requer o apoio de ferramentas que permitam uma mínima estruturação dos problemas abordados, assim como a organização de materiais, estudantes, professores, grupos e avaliações. Apesar da utilização de Ambientes Virtuais de Aprendizagem (AVA) e Sistemas de Tutores Inteligentes (STI), a maioria das experiências da ABP em disciplinas de Programação faz uso de sistemas de Juizes Online (JO).

Wasik *et al.* (2018) realizaram um mapeamento detalhado sobre JO, definindo-os como sistemas seguros, confiáveis e contínuos, cujo objetivo geral é a avaliação de algoritmos submetidos por usuários distribuídos por todo o globo. Os autores listaram e avaliaram mais de 96 JO, classificando-os, de acordo com seus principais objetivos, em: sistemas de apoio à organização de competições de programação; sistemas para o aprimoramento dos processos de educação e recrutamento; sistemas para realização de grandes desafios; compiladores *on-line* ou plataformas de desenvolvimento integrado.

Por fim, entre os sistemas de JO que consideramos mais bem sucedidos, gostaríamos de ressaltar o UVA Online Judge (REVILLA; MANZOOR; LIU, 2008), o URI Online Judge (BEZ; TONIN; RODEGHERI, 2014), o CodeBench (GALVÃO; FERNANDES; GADELHA, 2016), e o CodeForces (MIRZAYANOV *et al.*, 2020). O URI e o CodeBench apresentam problemas em língua portuguesa, sendo ambos ferramentas bem maduras. Por sua vez, o URI, o UVA e o CodeForces têm sido usados em treinamentos para maratonas de programação no Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB) por muitos anos.

4 Análise, projeto e desenvolvimento do C073

Apesar de existir um grande número de sistemas de JO, estes não foram concebidos explicitamente como ferramentas de apoio à ABP, como é o caso do C073. Nesse contexto, esta seção aborda o processo de desenvolvimento dessa plataforma, as tecnologias escolhidas e sua arquitetura, além de justificar as decisões que nortearam esse processo.

Seguindo as boas práticas da Engenharia de Software, a primeira etapa do processo de desenvolvimento do C073 foi a análise de requisitos. Entre os requisitos levantados, podemos destacar a divisão dos usuários de acordo com os papéis de administrador, professor, instrutor e estudante; a simplicidade no cadastro de problemas; a possibilidade de criação de turmas com acompanhamento detalhado; o gerenciamento de materiais de

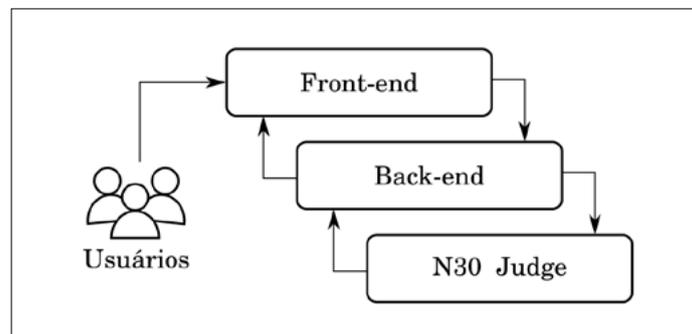
apoio, como apostilas, livros e vídeos; e a garantia de segurança e desempenho durante o julgamento dos códigos enviados, visto que estes não são confiáveis.

Em sequência, foram levantados os casos de uso do sistema. Dentre estes, ressaltamos a possibilidade de qualquer usuário da ferramenta criar os próprios problemas. Em relação aos professores, isso lhes permite a personalização de problemas, de modo a adequá-los ao grau cognitivo da classe, além de possibilitar uma maior contextualização. Em relação aos estudantes, essa característica suscita a criatividade e apresenta um grande potencial de aumentar tanto a motivação quanto o engajamento, permitindo, por exemplo, que usuários desafiem uns aos outros. Vale notar que tais problemas devem ser homologados antes de entrar no banco de dados geral do C073.

O próximo passo foi a definição da arquitetura do projeto. Optou-se por uma estrutura modular em três camadas, como exibido na Figura 2, facilitando tanto o desenvolvimento quanto a manutenção, além de possibilitar que qualquer um dos módulos seja substituído, paralelizado, ou mesmo atualizado, com um mínimo impacto às outras camadas.

Figura 2 ►

Arquitetura modular do C073.
Fonte: elaborada pelos autores



A camada mais alta da arquitetura é o *Front-End* (FE), responsável por intermediar a interação dos usuários com o sistema, caracterizando-se por ser processado no lado do cliente. Optou-se inicialmente por uma interface *web*, oferecendo acesso aos usuários através do navegador. O *framework* escolhido para o desenvolvimento do FE foi o Vue³, por conta de sua leveza, segurança, versatilidade, suporte da comunidade e licença *Open Source* (MIT License).

[3] <https://vuejs.org/>

Por sua vez, a funcionalidade mais importante a ser ofertada pelo Back-End (BE) é o gerenciamento dos bancos de dados da aplicação e da base de problemas. Outras atribuições do BE são a autenticação e autorização de usuários, a oferta de uma API (do inglês *Application Programming Interface*) com todos os serviços que podem ser utilizados pelo FE e o gerenciamento do julgamento dos códigos enviados para o sistema. Vale observar que o julgamento em si é realizado pelo módulo N30. O BE foi desenvolvido utilizando Django⁴ e Django Rest Framework⁵, enquanto JSON⁶ foi escolhida como linguagem de transferência de dados da API do BE.

[4] <https://www.djangoproject.com/>

[5] <https://www.django-rest-framework.org/>

[6] <https://www.json.org/>

Dado o baixo acoplamento do BE com os outros módulos, torna-se possível que sejam desenvolvidas novas aplicações para utilizar sua API, como aplicativos para celular (ou mesmo videogames) e interfaces de linha de comando. Por outro lado, a arquitetura modular do C073 possibilita que o BE lance várias instâncias do N30 paralelamente, que podem ser executadas no mesmo servidor ou em computadores acessíveis pela rede. Essa característica garante que o C073 seja escalável e elástico, dado que as instâncias do N30 podem ser iniciadas e finalizadas sob demanda.

[7] N30 é uma abreviação para Núcleo, com números substituindo as letras “e” e “o”.

Por fim, o módulo N30⁷ é responsável por compilar o código enviado (quando necessário), executá-lo e comparar o resultado com as saídas consideradas corretas. O N30 ainda precisa garantir que o programa não exceda os limites de memória, tempo de execução e escrita em disco, não acesse arquivos indevidos, não crie processos extras

[8] https://github.com/ifpb/C0D3_core

[9] <https://www.gnu.org/software/bash/>

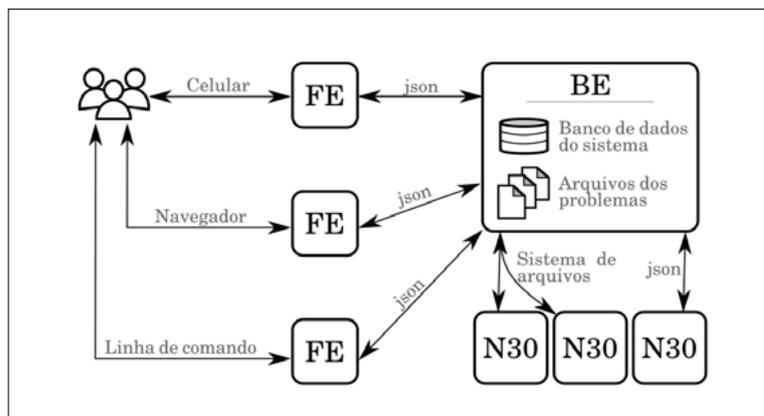
[10] <https://www.docker.com/>

[11] <https://hub.docker.com/>

e não permita acesso remoto ao servidor. O início do desenvolvimento do N30 se deu antes do projeto C073, sob a alcunha C0D3_core⁸ e licença *Open Source* MIT. A linguagem escolhida para o desenvolvimento do N30 foi o *Shell Script*, especificamente Bash⁹, por conta da sua íntima ligação com o núcleo do sistema operacional, que permite um maior controle sobre a utilização indevida de recursos do computador. De forma complementar, uma segunda camada de segurança foi adicionada pela utilização da tecnologia de contêineres Docker¹⁰, que possibilita que o código enviado seja executado em um ambiente isolado e com recursos limitados, o que aumenta a dificuldade de um programa malicioso prejudicar o sistema C073 como um todo. Adicionalmente, o uso do Docker facilita a criação e adição de novas linguagens ao N30 por meio dos contêineres prontos, disponibilizados no Docker Hub¹¹. A respeito dessa facilidade, foi desenvolvida e adicionada ao C073 a linguagem SIM (ALBUQUERQUE, 2021), criada com o intuito de facilitar o ensino de disciplinas de Circuitos Lógicos e composta de uma sintaxe mais simples em comparação às alternativas similares.

A Figura 3 detalha a arquitetura modular em três camadas do C073, evidenciando a possibilidade de múltiplos FE, a linguagem de comunicação da API do BE, as responsabilidades do BE e a possibilidade de instanciamento de vários N30, que podem utilizar o mesmo servidor ou serem instalados sob demanda em outros computadores, desde que sejam acessíveis através da rede.

Figura 3 ►
Arquitetura detalhada do C073.
Fonte: elaborada pelos autores



5 Resultados da pesquisa

Esta seção discute os resultados alcançados, apresentando a versão final do C073 e suas principais funcionalidades, além de traçar um paralelo direto destas com a metodologia ABP, relacionando-as com os pontos elencados nos Quadros 1 e 2 e com os pontos discutidos no referencial teórico e nos trabalhos relacionados.

Após a autenticação, o usuário é direcionado à tela inicial da plataforma C073, exibida na Figura 4. Além de apresentar um menu para acesso às principais funcionalidades do sistema, essa tela apresenta um carrossel de fotos de estudantes que obtiveram notório sucesso profissional e/ou acadêmico por meio da programação. Essa característica tende a criar uma maior identificação do estudante com a disciplina e com o curso, estimulando-lhe a motivação e o engajamento. Desse modo, essa característica atua diretamente contra as dificuldades (s.v), d(1), d(2), d(4) e (g.4).

Outra característica presente na tela inicial do sistema é um conjunto de artigos e tutoriais sobre o uso da ferramenta, tanto para professores quanto para estudantes, explicando o processo de busca e resolução de problemas, assim como o processo de criação destes. A oferta de exemplos e tutoriais em lugares visíveis e de simples acesso

objetiva a mitigação das dificuldades de adaptação a uma nova ferramenta, atuando nos pontos (s.vi), (g.5) e (g.6).

A funcionalidade mais importante para qualquer ferramenta de apoio à ABP no ensino de programação é que esta apresente os problemas aos usuários e os proveja de meios que possibilitem a sua resolução. Nesse ponto, deve-se salientar a importância de haver um robusto banco de problemas prontos, diversificados e bem classificados em relação ao conteúdo, ao contexto e também à dificuldade. Isso mantém o estudante por mais tempo na plataforma, dado que não o limita aos problemas selecionados pelo professor, contrapondo os pontos b(5), d(5) e (o.ii) e fomentando (g.3).

Figura 4 ►
Página inicial da plataforma C073.
Fonte: elaborada pelos autores



Por outro lado, o banco de problemas também oferece ao professor uma base para construção de atividades, permitindo-lhe usar os problemas prontos, ou mesmo copiá-los e editá-los alterando apenas a sua descrição, o que possibilita a ampliação da personalização e contextualização do ensino, atuando nos pontos (s.vi), a(2), a(6) e (g.7). A Figura 5 ilustra a base de problemas do C073¹² e seu mecanismo de busca avançada.

[12] A base de dados conta atualmente com 150 questões, mas está em rápido crescimento, pois estão sendo portados todos os problemas utilizados em competições de programação do IFPB.

Figura 5 ►
Banco de dados de problemas.
Fonte: arquivo dos autores

Problemas	
Termo de Busca	Buscar
Problema P0001 (programação) Olá Mundo	algoritmo construtivo strings Elo 1000 (razoável)
Problema P0003 (programação) Ida e Volta	Elo 1000 (razoável)
Problema P0002 (programação) Sequência de Números	implementação Elo 1000 (razoável)
Problema P0004 (programação) Simulador de Comandos	strings ordenação Elo 1500 (perigo)

Ao selecionar um problema, o estudante é levado à sua página de resolução, como exemplificado na Figura 6. Nessa página, são apresentados o nome do problema, sua descrição, a definição detalhada das entradas e saídas esperadas e casos de testes para serem tomados como exemplo. Sobre a descrição do problema, esta pode ser feita em linguagem *markdown*¹³, permitindo, de forma simples, não apenas o uso de texto formatado e realce de sintaxe de códigos, mas ainda a inclusão de tabelas, imagens,

[13] <https://www.markdownguide.org/>

vídeos ou mesmo *links* para arquivos ou explicações extras. Trata-se, portanto, de uma forma de responder às limitações apresentadas no ponto a(4).

Figura 6 ►
Página de resolução
de problemas.
Fonte: arquivo
dos autores

Problema P0001 (programação)
Olá Mundo

Olá, bem vindo ao C073. Pra começar, apenas leia uma string e imprima ela na saída padrão. =-)

Entrada
Apenas uma palavra

Saída
A palavra lida na entrada

Testes

Exemplo de Entrada 1
olah

Exemplo de Saída 1
olah

```
#!/bin/python3
entrada=input()
saida=entrada
print(saida)
```

Python3

Adicionalmente, a tela de resolução de problemas apresenta um campo para a implementação da resposta, que pode ser codificada em uma das doze linguagens aceitas inicialmente pelo C073, como Python, Java e C++, por exemplo. No momento, está sendo desenvolvida uma funcionalidade para oferta de *feedback* em tempo real durante a programação em linguagem Python, evitando pequenos erros sintáticos e oferecendo materiais com conteúdos direcionados ao erro encontrado. Essa é uma tentativa de fechar a maior lacuna detectada no mapeamento de Costa e Rocha (2018), discutido na Seção 3.2.

Uma vez que o estudante escreva a solução para o problema e esteja confiante de que esta está correta, ele deve submetê-la para que o C073 (especificamente o N30) a avalie. O usuário será direcionado à página de *feedback* da submissão, apresentada na Figura 7. Essa página apresenta dados sobre a execução da solução submetida – tais como o código-fonte, o tempo de execução do programa e o seu uso de memória –, assim como o resultado do seu julgamento, que pode ser “Resposta errada”, “Tempo limite alcançado”, “Erro em tempo de execução” ou “Aceito”, que representa a resposta correta.

Figura 7 ►
Página de *feedback* da
submissão.
Fonte: arquivo
dos autores

Submissão 202111487888414

Questão: P0002 - Usuário: maria Data: 13/05/2021 20:04
Sequência de Números
Linguagem: Python3 Tempo: 468 ms Memória: 7 MB
Status: Aceito

```
# Entrada
ini=int(input())
fim=int(input())

# Saída
for x in range(ini, fim+1):
    print(x)
```

Comentários

Normalmente, o julgamento das soluções submetidas não leva mais que um minuto, possibilitando que o estudante receba um *feedback* quase automático a respeito da sua solução. De forma suplementar, o professor pode adicionar comentários à submissão, aumentando consideravelmente a qualidade do *feedback* oferecido. O julgamento automático e a possibilidade de comentar as soluções propostas têm potencial atenuante sobre os pontos (s.vi), a(1), a(2) e a(3).

Para propiciar uma melhor organização de estudantes, conteúdos e turmas, o C073 permite aos professores, instrutores e estudantes experientes a criação de cursos. Nesses cursos, é possível realizar o agendamento e gerenciamento de atividades, como, por exemplo, exercícios de fixação ou avaliações, compostos por um conjunto de problemas sobre determinado assunto. Em especial, diferentemente da maioria das ferramentas de apoio à ABP para o ensino de Programação, as atividades podem englobar diversos conteúdos além dos problemas, como vídeos, apresentações, apostilas, livros, *links*, exemplos de código ou arquivos em geral. Dessa forma, o estudante tende a encontrar um ambiente mais rico, que oferece diferentes caminhos, cada vez mais dinâmicos, para obtenção do conhecimento, conforme elencado no ponto (o.ii). Essa característica reforça o enfrentamento aos problemas a(4), b(2), b(3), b(4), c(2) e d(5).

Os cursos também possibilitam a criação de times, que podem ser usados para realização de tarefas em grupo, promovendo uma maior colaboração (alinhada aos pontos (o.i) e (g.2)). Observamos que o conhecimento que cada usuário (ou time) venha a obter durante a resolução de uma atividade pode retornar para a turma utilizando a função “*bag*”, que permite ao estudante enviar arquivos para que sejam anexados à atividade, tornando-os acessíveis a todos. Dessa forma, a *bag* suscita discussão e reflexão sobre os conhecimentos adquiridos, em alinhamento direto com os pontos 5, 6 e 7 elencados por Barrows e Savery (apresentados na Seção 2). A Figura 8 exibe a página de um curso.

Figura 8 ►
Página do curso
“Introdução ao C073”.
Fonte: arquivo
dos autores

The screenshot shows a course interface for 'Curso RMXZ4Z'. At the top, it lists the course name, the responsible person (@admin), and the course title (Introdução ao C073). Below this, there are navigation links for 'Ferramentas: Nova Atividade | Editar o Curso | Desempenho'. The main content is divided into two sections: 'Atividades' and 'Problemas para começar'. The 'Atividades' section includes a 'Leitura Básica' activity with a date range of 2021-05-12 to 2021-05-15, described as 'Bases para iniciar o C073'. It lists three items: 'Página: 1 - Começando bem!', 'Vídeo: Instalando o Python3', and 'Arquivo: Arquivos com exemplos de Entrada e Saída'. The 'Problemas para começar' section also has a date range of 2021-05-12 to 2021-05-15 and lists three problems: 'Problema: P0001 - Olá Mundo', 'Problema: T0002 - Ida e Volta', and 'Problema: T0003 - Sequência de Números'.

Por fim, como exposto na Figura 9, o C073 permite um acompanhamento em tempo real dos estudantes enquanto estes resolvem as atividades, possibilitando ao professor, a partir de uma tela única, acessar as submissões e julgamentos dados pela plataforma. A seleção de uma coluna da tabela de acompanhamento leva o professor às submissões em questão, permitindo a leitura do código e a possível adição de comentários. A possibilidade de realizar um acompanhamento mais eficiente dos estudantes tende a mitigar os problemas levantados nos pontos a(1), a(2) e a(3).

Figura 9 ►
Página de acompanhamento
de atividades.
Fonte: arquivo dos autores

Problemas para começar [2021-05-12 - 2021-05-15]			
User	P0001	T0002	T0003
gov	1	1	1
maria		1	1
jose		3	1

6 Considerações finais

Este trabalho apresenta a ferramenta C073, planejada, desde a sua concepção, como uma ferramenta de apoio para o ensino de programação usando a metodologia de aprendizagem baseada em problemas. O C073 oferece opções eficazes para desafiar os problemas mais comuns enfrentados no processo de ensino-aprendizagem de disciplinas ligadas à programação de computadores, propiciando um acompanhamento mais eficiente e próximo do desempenho dos estudantes, uma maior contextualização para os exercícios propostos e um *feedback* rápido e preciso.

Entre as contribuições deste artigo, destacamos o levantamento bibliográfico e a estruturação das dificuldades enfrentadas no tocante ao ensino de Programação; as discussões sobre o modelo tradicional de ensino em contraste com as Metodologias Ativas e com a Aprendizagem Baseada em Problemas; o mapeamento dos trabalhos relacionados à aplicação da ABP tanto na área da Computação quanto nas disciplinas de Programação, que levam a diversos *surveys* capazes de agrupar o conhecimento da área; e, mais importante, a apresentação da ferramenta C073, da sua arquitetura e de suas características, traçando um paralelo direto com a metodologia ABP.

É importante pontuar, contudo, que a plataforma C073 se encontra na versão de testes alfa, estando suscetível a problemas e erros, que serão corrigidos tão logo detectados. Em contrapartida, o C073 está em desenvolvimento contínuo, tendo novas funcionalidades e novos problemas adicionados periodicamente.

Partindo das discussões fomentadas, pode-se afirmar que a plataforma C073 tem forte potencial para trazer um pouco da evolução tecnológica que foi vivenciada pelos autores para a sala de aula, auxiliando tanto os professores quanto os estudantes na aplicação da Aprendizagem Baseada em Problemas.

Por fim, como proposta para trabalhos futuros, pretende-se que os impactos da ferramenta apresentada sejam avaliados de forma mais detalhada quanto ao ensino de disciplinas da área de Programação. Tal avaliação pode ser feita pela comparação do desempenho dos alunos que usaram o C073 com aqueles que não o fizeram.

Financiamento

Esta pesquisa foi parcialmente financiada pelo IFPB por meio do Edital 01/2020 - Chamada Interconecta. Mais informações disponíveis em: <https://www.ifpb.edu.br/prpipg/editais/ano-2020/edital-no-01-2020-pesquisa>.

Conflito de interesses

Os autores declaram não haver conflito de interesses.

Referências

ALBUQUERQUE, K. M. M. **S1M e C073**: Ferramentas de apoio ao ensino de circuitos lógicos. 2021. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Elétrica) – Instituto Federal de Educação Ciência e Tecnologia da Paraíba, João Pessoa, 2021. Disponível em: https://www.researchgate.net/publication/368503537_S1M_e_C073_FERRAMENTAS_DE_APOIO_AO_ENSINO_DE_CIRCUITOS_LOGICOS. Acesso em: 14 fev. 2023.

ALVES, M. O.; MEDEIROS, F. P. A.; MELO, L. B. Levantamento do estado da arte sobre aprendizagem baseada em problemas na educação a distância e híbrida. *In*: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 31., 2020, Natal. **Anais [...]**. Natal: SBC, 2020. p. 61-71. DOI: <https://doi.org/10.5753/cbie.sbie.2020.61>.

BENNEDSEN, J.; CASPERSEN, M. E. Failure rates in introductory programming. **ACM SIGCSE Bulletin**, v. 39, n. 2, p. 32-36, 2007. DOI: <http://dx.doi.org/10.1145/1272848.1272879>.

BERGIN, S.; REILLY, R. The influence of motivation and comfort-level on learning to program. *In*: ANNUAL WORKSHOP OF THE PSYCHOLOGY OF PROGRAMMING INTEREST GROUP, 17., 2005, Brighton. **Proceedings [...]**. Brighton: University of Sussex, 2005. p. 293-304. Disponível em: <https://mural.maynoothuniversity.ie/8685/>. Acesso em: 10 out. 2022.

BEZ, J. L.; TONIN, N. A.; RODEGHERI, P. R. URI Online Judge Academic: a tool for algorithms and programming classes. *In*: INTERNATIONAL CONFERENCE ON COMPUTER SCIENCE & EDUCATION, 9., 2014, Vancouver. **Proceedings [...]**. Vancouver: IEEE, 2014. p. 149-152. DOI: <https://doi.org/10.1109/ICCSE.2014.6926445>.

BONWELL, C. C.; EISON, J. A. **Active learning**: creating excitement in the classroom. 1991 ASHE-ERIC Higher Education Reports. Washington, D.C.: ERIC Clearinghouse on Higher Education, The George Washington University, 1991. Disponível em: <https://eric.ed.gov/?id=ED336049>. Acesso em: 10 out. 2022.

BOSSE, Y.; GEROSA, M. A. 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. DOI: <https://doi.org/10.5753/wei.2015.10259>.

CADAVID, J. M.; CORCHO, A. F. P. Competitive programming and gamification as strategy to engage students in computer science courses. **Revista ESPACIOS**, v. 39, n. 35, p. 11-23, 2018. Disponível em: <https://www.revistaespacios.com/a18v39n35/a18v39n35p11.pdf>. Acesso em: 10 out. 2022.

CGI.br – COMITÊ GESTOR DA INTERNET NO BRASIL. **TIC Domicílios 2019**: pesquisa sobre o uso das Tecnologias de Informação e Comunicação nos domicílios brasileiros. São Paulo: Comitê Gestor da Internet no Brasil, 2020. Disponível em: https://cetic.br/media/analises/tic_domicilios_2019_coletiva_imprensa.pdf. Acesso em: 10 out. 2022.

COELHO, M. C.; GUEDES, A. M. A. Aprendizagem baseada em problemas aplicada à programação de computadores: um mapeamento sistemático. **RENOTE**, v. 18, n. 2, p. 570-580, 2020. DOI: <https://doi.org/10.22456/1679-1916.110298>.

COSTA, E. B.; ROCHA, H. J. B. Programação numa abordagem de aprendizagem baseada em resolução de problemas e jogos: um mapeamento sistemático. *In*: SBGAMES, 17., 2018, Foz do Iguaçu. **Proceedings** [...]. Foz do Iguaçu: SBC, 2018. p. 1216-1223. Disponível em: <http://www.sbgames.org/sbgames2018/files/papers/EducacaoFull/188378.pdf>. Acesso em: 10 out. 2022.

GALVÃO, L. S. G; FERNANDES, D. B. F.; GADELHA, B. F. Juiz online como ferramenta de apoio a uma metodologia de ensino híbrido em programação. *In*: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 26., 2016, Uberlândia. **Anais** [...]. Uberlândia: SBC, 2016. p. 140-149. DOI: <http://dx.doi.org/10.5753/cbie.sbie.2016.140>.

GOMES, A.; MENDES, A. J. Learning to program - difficulties and solutions. *In*: INTERNATIONAL CONFERENCE ON ENGINEERING EDUCATION (ICEE), 2007, Coimbra. **Proceedings** [...]. Coimbra: Faculdade de Ciências e Tecnologia da Universidade de Coimbra, 2007. Disponível em: <http://icee2007.dei.uc.pt/proceedings/papers/411.pdf>. Acesso em: 10 out. 2022.

HUNG, W. The 9-step problem design process for problem-based learning: Application of the 3C3R model. **Educational Research Review**, v. 4, n. 2, p. 118-141, 2009. DOI: <https://doi.org/10.1016/j.edurev.2008.12.001>.

IFPB – INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA. **Portal do Estudante**. 2021. Disponível em: <https://estudante.ifpb.edu.br/>. Acesso em: 30 abr. 2021.

JENKINS, T. On the difficulty of learning to program. *In*: ANNUAL CONFERENCE OF THE LTSN CENTRE FOR INFORMATION AND COMPUTER SCIENCES, 3., 2002, Leeds. **Proceedings** [...]. Loughborough: Loughborough University, 2002. v. 4, p. 53-58. Disponível em: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.596.9994&rep=rep1&type=pdf>. Acesso em: 10 out. 2022.

KUMAR, A. N. Learning programming by solving problems. *In*: CASSEL, L.; REIS, R. A. (ed.). **Informatics curricula and teaching methods**. Boston: Springer, 2003. p. 29-39. DOI: https://doi.org/10.1007/978-0-387-35619-8_4.

MBODILA, M.; MUHANDJI, K. The use of ICT in Education: a comparison of traditional pedagogy and emerging pedagogy enabled by ICT's. *In: INTERNATIONAL CONFERENCE ON FRONTIERS IN EDUCATION: COMPUTER SCIENCE AND COMPUTER ENGINEERING (FECS)*, 2012, Las Vegas. **Proceedings** [...]. Las Vegas: CSREA Press, 2012. p. 590-593. Disponível em: <http://worldcomp-proceedings.com/proc/p2012/FEC2651.pdf>. Acesso em: 10 out. 2022.

MILNE, I.; ROWE, G. Difficulties in learning and teaching programming: views of students and tutors. **Education and Information Technologies**, v. 7, n. 1, p. 55-66, 2002. DOI: <https://doi.org/10.1023/A:1015362608943>.

MIRZAYANOV, M.; PAVLOVA, O.; MAVRIN, P.; MELNIKOV, R.; PLOTNIKOV, A.; PARFENOV, V.; STANKEVICH, A. Codeforces as an educational platform for learning programming in digitalization. **Olympiads in Informatics**, v. 14, p. 133-142, 2020. DOI: <http://dx.doi.org/10.15388/oi.2020.10>.

MOW, I. C. Issues and difficulties in teaching novice computer programming. *In: ISKANDER, M. (ed.) Innovative techniques in instruction technology, e-learning, e-assessment, and education*. Dordrecht: Springer, 2008. p. 199-204. DOI: https://doi.org/10.1007/978-1-4020-8739-4_36.

OLIVEIRA, A. M. C. A.; SANTOS, S. C.; GARCIA, V. C. PBL in teaching computing: An overview of the last 15 years. *In: 2013 IEEE FRONTIERS IN EDUCATION CONFERENCE (FIE)*, 2013, Oklahoma City. **Proceedings** [...]. Oklahoma City: IEEE, 2013. p. 267-272. DOI: <https://doi.org/10.1109/FIE.2013.6684830>.

OLIVEIRA, F. S.; SANTOS, S. PBLMaestro: a virtual learning environment for the implementation of problem-based learning approach in computer education. *In: 2016 IEEE FRONTIERS IN EDUCATION CONFERENCE (FIE)*, 2016, Erie. **Proceedings** [...]. Erie: IEEE, 2016. p. 1-9. DOI: <https://doi.org/10.1109/FIE.2016.7757388>.

PRINCE, M. Does active learning work? A review of the research. **Journal of Engineering Education**, v. 93, n. 3, p. 223-231, 2004. DOI: <https://doi.org/10.1002/j.2168-9830.2004.tb00809.x>.

REVILLA, M. A.; MANZOOR, S.; LIU, R. Competitive learning in informatics: the UVa online judge experience. **Olympiads in Informatics**, v. 2, n. 10, p. 131-148, 2008. Disponível em: <https://ioinformatics.org/journal/INFOL035.pdf>. Acesso em: 10 out. 2022.

RIBEIRO, L. R. C. **Aprendizagem baseada em problemas (PBL): uma experiência no ensino superior**. São Carlos: EdUFSCar, 2008. DOI: <https://doi.org/10.7476/9788576002970>.

RODRIGUES, A. N. **Um framework conceitual para implementação e gestão da abordagem PBL no ensino de Computação**. 2018. Tese (Doutorado em Ciência da Computação) – Universidade Federal de Pernambuco, Recife, 2018. Disponível em: <https://repositorio.ufpe.br/handle/123456789/33637>. Acesso em: 10 out. 2022.

SAVERY, J. R. Overview of problem-based learning: definitions and distinctions. **Interdisciplinary Journal of Problem-Based Learning (IJPBL)**, v. 1, n. 1, p. 9-20, 2006. DOI: <https://doi.org/10.7771/1541-5015.1002>.

SAVERY, J. R. Overview of problem-based learning: Definitions and distinctions. *In*: WALKER, A.; LEARY, H.; HMELO-SILVER, C. E.; ERTMER, P. A. **Essential readings in problem-based learning**: Exploring and extending the legacy of Howard S. Barrows. West Lafayette: Purdue University Press, 2015. p. 5-15.

SOUZA, D. M.; BATISTA, M. H. S.; BARBOSA, E. F. Problemas e dificuldades no ensino de programação: um mapeamento sistemático. **Revista Brasileira de Informática na Educação**, v. 24, n. 1, p. 39-52, 2016. DOI: <http://dx.doi.org/10.5753/rbie.2016.24.1.39>.

WASIK, S.; ANTCZAK, M.; BADURA, J.; LASKOWSKI, A.; STERNAL, T. A survey on online judge systems and their applications. **ACM Computing Surveys (CSUR)**, v. 51, n. 1, p. 1-34, 2018. DOI: <https://doi.org/10.1145/3143560>.