

# Um relato de experiência sobre o ensino de programação de computadores no Ensino Básico por meio da Olimpíada Brasileira de Informática

Alexandre F. Lopes<sup>1</sup>, Thalia S. Santana<sup>1</sup>, Adriano H. Braga<sup>1</sup>

<sup>1</sup>Instituto Federal Goiano – Campus Ceres (IF GOIANO)  
Caixa Postal 51 – 76.300-000 – Ceres – GO – Brasil

{alexandre9999lopes, thaliassantana15}@gmail.com,  
adriano.braga@ifgoiano.edu.br

**Abstract.** *This work constitutes the experience report of a preparatory course for the Brazilian Olympiad of Informatics (OBI) and algorithm solutions, with the purpose of stimulating the students for the computation, in addition to opportunizing the participation of students in scientific competitions. Thus, it was possible to verify the methodology used and to suggest a new teaching-learning approach. As a result, it was possible to obtain an improvement of the logical reasoning, to instruct the computational thinking and to encourage the interest for the learning in programming languages of the involved ones, having as consequence a good classification in the phases of the OBI.*

**Resumo.** *Este trabalho constitui do relato de experiência de um curso preparatório para a Olimpíada Brasileira de Informática (OBI) e soluções de algoritmos, com o intuito de estimular os estudantes para a computação, além de oportunizar a participação de discentes em competições científicas. Dessa forma, foi possível verificar a metodologia utilizada e sugerir nova abordagem de ensino-aprendizagem. Como resultado foi possível obter um aprimoramento do raciocínio lógico, instruir o pensamento computacional e estimular o interesse pelo aprendizado em linguagens de programação dos envolvidos, tendo como consequência uma boa classificação nas etapas da OBI.*

## 1. Introdução

Com o aumento da procura por profissionais de computação, especificamente de desenvolvedores de *softwares*, faz-se necessário uma maior quantidade de estudantes neste campo. Consequentemente, para se tornar um bom profissional, são necessárias habilidades específicas tais como: raciocínio lógico, interpretação de problemas, além do conhecimento técnico das linguagens de programação (HOED, 2016). Devido a estes requisitos, alguns alunos acabam desistindo de cursos que ensinam programação de computadores dado o elevado nível de exigência e assim, não dirime a situação da demanda por profissionais capacitados na área (DE CASTRO, 2003).

Uma estratégia para tentar resolver os problemas supracitados é promover novas abordagens de ensino-aprendizagem. Gomes et al. (2008) aponta que os métodos de ensino e as estratégias docentes relacionadas a programação de computadores nem sempre tornam-se adequadas a real carência de seus estudantes. Várias metodologias vêm sendo aplicadas para sanar tal deficiência, como no caso de Maciel et al. (2012) e Campana Filho, Oliveira e Oliveira (2016), que utilizaram respectivamente em seus projetos a análise e classificação de códigos como base de apoio para o ensino da programação.

Outro exemplo trata-se das olimpíadas científicas de conhecimento como iniciativa de estímulo e aprendizado. No contexto da computação, a Olimpíada Brasileira de Informática (OBI) é realizada anualmente desde 1999, promovida pela Sociedade Brasileira de Computação (SBC) seguindo os mesmos padrões das demais, como a Olimpíada Brasileira de Matemática (OBM) e a Olimpíada Brasileira de Física (OBF) (SBC, 2019). Tal competição tem como objetivo promover a computação, incluindo até os mais básicos níveis de ensino, despertando o interesse de discentes pelas ciências da computação, estudo indispensável na formação básica atual (OBI, 2019).

Assim, vários trabalhos vêm sendo realizados no Brasil a fim de difundir a informática por meio desta olimpíada científica. Barros, Ribeiro e Oeiras (2009) e Barros e Oeiras (2010), descreveram experiências de extensão universitária nas modalidades Iniciação e Programação. Para a última, os treinamentos abordaram a linguagem de programação Pascal, com extensionistas graduandos ministrantes do curso. Lucena et al. (2018) realizaram treinamentos para a OBI na educação básica por meio do Programa de Educação Tutorial (PET), sendo a linguagem de programação C escolhida para exercícios e resoluções de provas anteriores. Já Meneses et al. (2015) ofereceram um curso de introdução à programação também para alunos de ensino médio, dos quais os concluintes dedicaram-se, em uma segunda etapa do projeto, à competição com treinamentos presenciais.

Deste modo, contribuindo aos demais trabalhos relacionados, este artigo relata a experiência de um projeto conjunto de pesquisa e ensino que oferta um treinamento de solução de algoritmos, auxiliando no pensamento computacional e lógica de programação, bem como na preparação dos estudantes para olimpíadas científicas, como a OBI. O trabalho apresenta as metodologias de ensino utilizadas, as formas de identificação das dificuldades encontradas pelos participantes, além de índices de aprendizado e a classificação na Olimpíada, visando minimizar a evasão de discentes ingressantes em cursos de informática e atraindo-os cada vez mais para a Ciência da Computação.

## **2. Material e Métodos**

A execução do curso abordou temáticas como lógica e linguagens de programação, incluindo o contexto de preparação para competição na OBI. O público-alvo foram os acadêmicos do curso Técnico em Informática para Internet Integrado ao Ensino Médio, possuindo faixa etária entre 14 e 16 anos, sendo em maioria dos ingressantes da 1ª série.

Além de estratégias para introdução e inserção de conceitos computacionais, também realizou-se uma breve análise dos algoritmos produzidos na plataforma *URI Online Judge* (URI, 2019), visando a apuração da eficácia deste processo de ensino presente no nível de aprendizagem dos participantes.

## 2.1 Treinamentos

O curso preparatório teve início no primeiro semestre de 2018, se prolongando até a data da última fase da competição, sucedida no final de agosto do mesmo ano, sendo finalizando o treinamento com 48 horas/aula, sendo 12 encontros com carga horária de 4 horas para cada aula. Com frequência semanal, foi oferecido às quartas-feiras no horário livre dos estudantes como forma extracurricular, em um laboratório de informática da instituição portando a distribuição Linux Ubuntu 16.04, entre o período das 13h às 17h (Figura 1).

As aulas foram na maioria práticas por meio da solução de algoritmos nos computadores, mesclando com conteúdos teóricos, utilizando-se de listas de exercícios pré-definidas no URI conforme o conteúdo era repassado. Ao todo, quatro listas foram preparadas, sendo: a) operações matemáticas, com 10 problemas; b) seleção com *if*, com 15 problemas; c) repetição com *for/while*, com 15 problemas; e, d) vetores e matrizes, com 10 problemas, compondo um total de 50 exercícios. A equipe foi composta por um professor orientador e 3 bolsistas, sendo dois destes do curso de Graduação da mesma Instituição e 1 da 3ª Série do Ensino Médio, mesmo curso dos participantes.



**Figura 1. Curso preparatório ministrado aos estudantes: a) treinamento, b) aplicação da prova fase local da OBI. Fonte: Própria (2018).**

Além de indiretamente auxiliar os discentes em disciplinas presentes na matriz curricular de seus cursos, como Lógica de Programação, um dos intuits também tratou-se da oportunização da participação dos estudantes na OBI, bem como capacitá-los para a competição. Dessa forma os treinamentos tiveram como base principal a ementa oficial da Olimpíada, combinada a exercícios complementares para solução algorítmica. Também foram efetuadas simulações das provas da OBI com questões de edições anteriores, a fim de preparar os estudantes e familiarizá-los com a competição. Em relação a OBI, os discentes correspondiam-se aptos a participarem nas modalidades de programação níveis 1 e 2 (Ensino Médio).

Inicialmente, 25 discentes compareceram a aula inaugural do curso preparatório. Percebeu-se que conforme as atividades acadêmicas e avaliativas aumentavam, um menor número de acadêmicos faziam-se presentes no treinamento, já que o mesmo não possuía caráter obrigatório. Assim, com o intuito de manter a participação dos estudantes durante o mesmo, foram utilizadas diversificadas formas de integração, como: ensino de conceitos por meio de atividades lúdicas, sorteios de brindes conforme frequência e/ou resolução das listas e trabalhos de cooperação em grupo. Outro ponto de considerável incentivo foi a certificação de horas complementares ao final do curso aos participantes.

## 2.2 Python

Dentre todas as linguagens de programação que são aceitas para submissão de códigos na competição (Pascal, C, C++, Python, Java e Javascript), a linguagem selecionada para o treinamento foi Python. Conforme a *Python Software Foundation* (2019), foi desenvolvida de acordo com licença de código aberto, com aprovação da *Open Source Initiative* (OSI), a fim de manter-se como livremente utilizável e distribuível, até mesmo em fins comerciais, e, atualmente, possui sua licença sob administração da *Python Software Foundation*. Ademais, como *Integrated Development Environment* (IDE) foi utilizado o Python IDLE (PYTHON SOFTWARE FOUNDATION, 2019), ambiente padrão distribuído gratuitamente pelos desenvolvedores da linguagem e já instalado em todas as máquinas do laboratório de informática.

A linguagem Python é considerada como simples e de fácil aprendizado, inclusive para os iniciantes em programação (PYTHON SOFTWARE FOUNDATION, 2019). Pesente et al. (2016), defende o uso de linguagens menos complexas, de fácil entendimento e sintaxe simplificada, como o Python, para o início do ensino de programação. Rebouças et al. (2010), também aponta outros fatores que justificam a utilização desta no ensino de programação, como: módulos descomplicados de uso, *feedback* imediato e a exigência de indentação constante. Vale ainda ressaltar a opção pela linguagem devido ao fato de ser do conhecimento de todos os membros da equipe e ser utilizada nas disciplinas introdutórias de programação da Instituição.

## 2.3 URI Online Judge

O *URI Online Judge* é uma plataforma online desenvolvida pela Universidade Regional Integrada, que visa auxiliar professores e alunos de algoritmos ou linguagens de programação (BEZ et al., 2014). O mesmo oferece aos estudantes uma enorme quantidade de problemas separados em níveis de complexidades e categorias, correção em tempo real da submissão, permitindo a visualização do código-fonte enviado e apresentação detalhada dos erros, caso sejam encontrados. A plataforma está disponível nos seguintes idiomas: português, espanhol e inglês (URI, 2019).

A plataforma possui a funcionalidade “*Academic*” que foi amplamente utilizada neste projeto e para se ter o acesso é necessário enviar um e-mail para a equipe de suporte da plataforma. Após isso, é criada uma conta em que se torna possível elaborar listas de exercícios com prazos determinados, adicionar membros em turmas, monitorar o desempenho e visualizar a submissão dos problemas solicitados por grupo. Durante o curso preparatório foi criada uma turma na plataforma e assim a inclusão dos

competidores, que ao adentrarem autorizaram via termo de permissão liberar seus códigos para uso e compartilhamento. Ao decorrer dos treinamentos foram preparadas listas de exercícios referentes a cada conteúdo abordado no treinamento, por meio do URI *Online Judge*, e então solicitada a resolução dessas aos competidores, possibilitando obter os algoritmos e acompanhar o desempenho individual de todos, bem como uma visão geral dos temas de maior dificuldade com base na correteza dos exercícios.

A plataforma foi indispensável para a execução deste trabalho, tornando fácil e ágil várias etapas do ensino de programação. Como exemplo, a correção dos exercícios, que antes tinha de ser realizada manualmente pelos ministrantes, tarefa que durava horas e que se tornava inviável em tempo hábil para todas as submissões. Com o auxílio da ferramenta, a verificação foi executada de forma instantânea e contribuiu para a familiarização dos participantes com a correção automática, a qual é utilizada nas olimpíadas, poupando assim o tempo da equipe executora e proporcionando um maior enfoque no ensino de conteúdos técnicos, na sintaxe da linguagem, bem como interpretação de textos de acordo com enunciado dos problemas propostos. Por meio do ambiente, também foi possível analisar o desempenho geral, determinando o compasso das aulas, adiantando ou permanecendo no previsto em plano de ensino.

#### **2.4 MOSS (*Measure Of Software Similarity*)**

Após obter as resoluções finais dos estudantes via listas de exercícios aplicadas no URI, os algoritmos foram submetidos a uma análise feita no MOSS (*Measure Of Software Similarity*), um serviço de detecção de plágio amplamente utilizado na internet desde 1997 (SCHLEIMER et al., 2003). Esse serviço é concedido pela Universidade da Califórnia, na qual é necessária a requisição para a utilização do mesmo (AIKEN, 2018). Após o envio de arquivos para análise, é retornada uma página eletrônica com o resultado da análise de similaridade entre os arquivos. Atualmente o MOSS fornece análise de códigos escritos em várias linguagens como C, C++, Java, C# e Python.

Na nova versão do URI *Academic* é oferecido aos professores a integração da tecnologia MOSS, verificando nas listas de exercícios a ocorrência de casos de plágio das submissões enviadas (AIKEN, 2018). Contudo, durante a execução dos treinamentos e da coleta dos algoritmos, a ferramenta não foi utilizada. Logo, para o andamento da pesquisa, as resoluções das listas tiveram de ser baixadas do próprio URI *Academic*, em seguida enviadas diretamente ao serviço *web*.

A aplicação do MOSS de forma isolada foi justificada pelo fato de proporcionar a verificação de similaridade entre os códigos feitos pelos estudantes em relação às resoluções apresentadas em sala. Com os resultados obtidos foi possível constatar a proporção de códigos que assemelhavam-se ao ensinado em sala ou, caso houvesse, algoritmos desenvolvidos utilizando lógicas e técnicas não abordadas. Diferindo bastante do propósito principal da ferramenta que se fundamenta na detecção de plágio, utilizada neste trabalho para a mensuração do conhecimento alcançado.

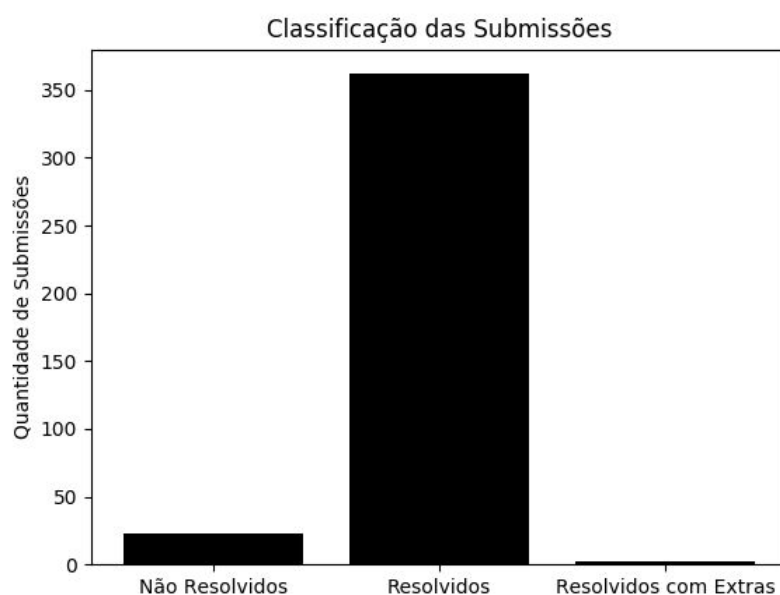
Os códigos foram agrupados por questão, em seguida submetidos ao MOSS, que retornou a análise com o grau de similaridade entre os algoritmos. Em seguida os códigos foram novamente agrupados, dessa vez conforme a porcentagem de semelhança entre eles, sendo a solução proposta considerada similar se o código obteve índice igual

ou superior a 70%, viabilizando inclusive estudos comparativos entre os algoritmos. Após a similaridade, verificou-se aqueles que possuíam resolução correta e os que não foram resolvidos. Posteriormente, uma análise mais minuciosa dos algoritmos resolvidos, possibilitou a verificação de um grupo sem similaridade com o que foi ensinado nas aulas, sendo considerados resolvidos com extras.

### 3. Resultados e Discussão

Visando aferir o nível de aprendizado alcançado no final do curso preparatório, o estudo limitou-se a analisar somente as submissões finais dos estudantes, ou seja, levando em questão apenas a última resolução para cada um dos problemas. Dessa forma, avaliou-se o conhecimento conclusivo adquirido pelos estudantes, desconsiderando as submissões elaboradas durante o processo de aprendizado, mas que ao mesmo tempo colaboraram para perguntas dos alunos e posterior eliminação de dificuldades durante as aulas ministradas, sendo as listas de exercícios atividade fim para despertar a lógica nos estudantes.

Dos algoritmos submetidos, que totalizaram 387 envios finais, foi possível a classificação em três grupos distintos: não resolvidos, resolvidos e resolvidos com extras, conforme demonstrado no Gráfico 1. Os exercícios resolvidos referem-se aos códigos com resolução assertiva para os problemas propostos utilizando os conteúdos adquiridos durante as aulas. Dos resolvidos com extras, foram aqueles que não possuíam similaridade com o ensinado, então foram verificados código a código para entender as soluções empregadas, as quais utilizaram de conceitos ainda não tratados no treinamento. Por fim, os não resolvidos tratam-se de todo o percentual de exercícios sem resposta algorítmica aceita dos cursistas cadastrados na turma do URI *Online Judge*.



**Gráfico 1. Classificação das submissões finais ao URI dos problemas algorítmicos. Fonte: Própria (2018).**

Do percentual de exercícios resolvidos (aproximadamente 93,5%), obteve-se uma boa taxa de resolubilidade. Tal índice foi alcançado mediante tentativa dos estudantes em responder a questão e em casos de dificuldade, os conteúdos necessários para a elaboração do código foram reforçados em sala de aula, o que garantiu elevada taxa de acertos, bem como o comprometimento dos discentes com as listas. Nos casos em que os problemas não foram resolvidos, correspondendo cerca de 6%, tal número se justifica pelos estudantes que evadiram do curso preparatório, não obtiveram todo o conteúdo programático para a solução dos mesmos e conseqüentemente não classificaram nas etapas da olimpíada. Nas situações em que as proposições não utilizaram dos conteúdos ministrados e que continham diferentes técnicas de programação, somou-se um total de aproximadamente 0,5% das submissões. Supõe-se que essa porcentagem foi gerada pelos participantes que possuíam conhecimentos prévios ao curso na área e/ou que aprofundaram no estudo além do ministrado nas aulas.

Sob o ponto de vista da classificação na OBI, o desempenho dos estudantes na edição de 2018 apresentou resultados satisfatórios. A olimpíada possui três etapas classificatórias, divididas em: fase local, fase regional e fase nacional. Os competidores que alcançam as maiores pontuações em cada etapa são classificados para a seguinte. Dessa forma, ressalta-se a eficácia de aplicação do curso em questão, tendo 18 selecionados na fase local para a regional, sendo todos estes participantes do treinamento ofertado. De um total de 111 classificados em todo o Brasil para a realização da etapa nacional, apenas 4 são do estado da Instituição e destes, 3 participaram do curso preparatório.

Ademais, após a terceira fase da Olimpíada foram colhidos alguns depoimentos de discentes participantes dos treinamentos, com o objetivo de avaliar e expressar sua opinião sobre a execução do projeto. Aluno 1 afirmou:

Foi muito bom pra mim, me ajudou na disciplina lógica de programação, um aprendizado muito gratificante. Gostei muito de como foram ministradas as aulas, como as atividades foram trabalhadas e em momento algum houve dificuldade na compreensão dos conceitos apresentados. Os treinamentos conseguiram apresentar todos os conteúdos cobrados na OBI, influenciando bastante meu desempenho na olimpíada, aprimorando meu raciocínio lógico. Outra habilidade desenvolvida foi a interpretação de textos, já que todas as questões resolvidas cobravam essa aptidão, havendo a prática intensa, contribuindo para minha vida acadêmica. Aperfeiçoando até o inglês, presente na maioria dos comandos da linguagem de programação, aprendendo novas palavras e me familiarizando a língua estrangeira. (Entrevista, Cidade da Instituição, 26 de outubro de 2018).

Em algumas palavras, outro estudante descreve o projeto, destacando a importância e o impacto positivo que os treinamentos possibilitaram aos acadêmicos, afirmando:

O treinamento ajudou bastante na lógica de programação e aumentou minhas habilidades com a linguagem de programação, me colocando à frente dos demais na competição. E com as aulas semanais a facilidade de compreensão do conteúdo aumenta. Os conteúdos foram aplicados de uma forma simples e didática descomplicando os temas. Os treinamentos contribuíram bastante em meu resultado na OBI, já que com as aulas práticas pude me familiarizar com as questões e o modelo da prova. Também desenvolveram meu raciocínio lógico, ajudando em matérias como matemática e física, e até em disciplinas

que não estão diretamente relacionadas com a informática, como o português, no desenvolvimento da estrutura e na solução de uma redação. (Entrevista, Cidade da Instituição, 26 de outubro de 2018).

#### **4. Conclusão**

Houve desempenho satisfatório por parte dos estudantes cursistas do treinamento, mostrando aprendizagem, domínio da lógica e compreensão dos conceitos apresentados, expressados por meio da porcentagem de questões resolvidas e classificação dos estudantes na OBI. Ressalta-se que boa parte dos discentes envolvidos obtiveram êxito na resolução dos exercícios, de modo que a taxa de abstenção de envios de problemas não resolvidos foi menor que 6%, contrapondo-se aos cerca de 94% de submissões corretas e com códigos otimizados.

Acredita-se que a taxa de classificação da referida escola participante dentre as fases da OBI, inclusive com cursistas na etapa nacional, colabora para a efetividade da metodologia de ensino-aprendizagem utilizada. Considera-se a abordagem exitosa, cabendo destacar que desde a primeira execução do treinamento no ano de 2015 a referida instituição pelo 4º ano consecutivo destaca-se como a escola com o maior número de representantes da região da Instituição. Visto que a participação possui caráter voluntário, manter a frequência dos estudantes após as datas das provas da OBI representa um dos obstáculos enfrentados, já que nem todos os discentes são classificados.

Assim sendo, a oferta de cursos preparatórios com a abordagem metodológica utilizada nesse estudo apresenta ser uma estratégia para motivar estudantes para a ciência da computação. Espera-se que o trabalho figure como uma sugestão de metodologias de ensino-aprendizagem, ao passo que contribua para a formação de um maior número profissionais capacitados e colabore na dirimção da demanda do mercado de trabalho, bem como na permanência de estudantes em cursos ligados a computação.

Como trabalhos futuros, sugere-se a aplicação de um curso preparatório em Ambiente Virtual de Aprendizado (AVA) para propiciar o acesso a um público maior, não somente dos discentes da Instituição, e que possam estudar no momento em que for mais propício para cada estudante. Propõe-se também uma análise comparativa dos estudantes que fizeram parte deste treinamento com os que não fizeram comparando-os ao rendimento das notas escolares e do Exame Nacional do Ensino Médio (ENEM).

#### **5. Referências**

- Aiken, A. (2019). “A System for Detecting Software Similarity”, <https://theory.stanford.edu/~aiken/moss/>, Fevereiro.
- Barros, L. G.; Ribeiro, S. P. S.; Oeiras, J. Y. Y. (2009). Projeto de Extensão Universitária para apoio e realização da Olimpíada Brasileira de Informática em Escolas, In: Anais do XXIX WEI, CSBC 2009, Bento Gonçalves/RS.



- Barros, L. D., & Oeiras, J. Y. Y. (2010). Extensão universitária em Computação por meio da Olimpíada Brasileira de Informática. In: Anais do XVIII Workshop sobre Educação em Computação (WEI), Belo Horizonte/MG.
- Bez, J. L., Tonin, N. A., & Rodegheri, P. R. (2014). URI Online Judge Academic: A tool for algorithms and programming classes. In: Computer Science & Education (ICCSE), 2014 9th International Conference on (pp. 149-152). IEEE.
- Campana Filho, J., Oliveira, M. & Oliveira, E. (2016). Classificação de Códigos C usando medidas de similaridade para apoio ao Ensino em Programação. In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE) (Vol. 27, No. 1, p. 1156).
- De Castro, T. H. C., De Castro Júnior, A. N., De Menezes, C. S., Boeres, M. C. S., & Rauber, M. C. P. V. (2003). Utilizando programação funcional em disciplinas introdutórias de computação. Anais do WEI.
- Gomes, A., Areias, C., Henriques, J., & Mendes, A. J. (2008). Aprendizagem de programação de computadores: dificuldades e ferramentas de suporte. Revista Portuguesa de Pedagogia, 161-179.
- Hoed, R. M. (2016). Análise da evasão em cursos superiores: o caso da evasão em cursos superiores da área de Computação. Brasília, DF: Universidade de Brasília.
- Lucena, L. B., Andrade, G. L. C., Soares, E. D. F. A., Oliveira, Á. G. G., Batista, L. M. D. S. D., Gomes, D. A., & Lima, R. W. (2018). AS AÇÕES DO PET NO DESENVOLVIMENTO DO CURSO DE CIÊNCIA DA COMPUTAÇÃO. In: 26º Workshop sobre Educação em Computação (WEI 2018), v. 26, n. 1/2018, SBC.
- Maciel, D. L. (2014). Sherlock n-overlap: Normalização invasiva e coeficiente de sobreposição para análise de similaridade entre códigos-fonte em disciplinas de programação. Dissertação (Mestrado em Engenharia de Teleinformática) - Centro de Tecnologia, Universidade Federal do Ceará, Fortaleza, 105 f.
- Meneses, L. F., Mai, L. F. F., Rosario, J., Oliveira, E., & Gomes, R. L. (2015). IntroComp: Atraindo alunos do ensino médio para uma instigante experiência com a programação. In: Anais do XXIII Workshop sobre Educação em Computação (WEI 2015), Recife, PE, SBC.
- Pesente et al. (2016). Uma abordagem de ensino de programação de computadores utilizando Scratch e Python, <http://www.sinect.com.br/2016/down.php?id=3723&q=1>, Novembro.
- Python Software Foundation (2019). “Python”, <https://www.python.org/>, Março.
- Rebouças et al. (2010). Aprendendo a ensinar programação combinando jogos e Python. In Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE) (Vol. 1, N. 1).
- SBC (2019). “Olimpíada Brasileira de Informática”, <http://www.sbc.org.br/educacao/312-olimpiada-brasileira-de-informatica>, Março.

Schleimer, S., Wilkerson, D. S., & Aiken, A. (2003). Winnowing: local algorithms for document fingerprinting. In: Proceedings of the 2003 ACM SIGMOD international conference on Management of data (pp. 76-85). ACM.

OBI (2019). “Olimpíada Brasileira de Informática”, <http://olimpiada.ic.unicamp.br/>, Fevereiro.

URI (2019). “URI *Online Judge*” , <https://www.urionlinejudge.com.br/>, Março