

# PROGRAMAÇÃO DE COMPUTADORES NO ENSINO MÉDIO: O ESTÍMULO DA OLIMPÍADA BRASILEIRA DE INFORMÁTICA

**Área Temática:** Educação

Ana Elisa Tozetto Piekarski<sup>1</sup>; Tony Alexander Hild<sup>1</sup>; Mauro Miazaki<sup>1</sup>; Elena Mariele Bini<sup>2</sup>; Higor Gardin<sup>3</sup>; João Vitor Mas Urtado<sup>3</sup>; Moisés Alonso Prestes<sup>3</sup>

**Instituição:** Universidade Estadual do Centro-Oeste (UNICENTRO)

## Resumo

Este trabalho descreve as atividades de treinamento dos alunos do ensino médio para a Olimpíada Brasileira de Informática (OBI) no biênio 2016-2017. Essas atividades fazem parte do projeto de extensão Ensino Extracurricular de Programação de Computadores (E2PC), vinculado ao Departamento de Ciência da Computação da Unicentro. A equipe executora envolve professores e acadêmicos do Bacharelado em Ciência da Computação (monitores). Utilizando a metodologia das competições de programação - mais especificamente da OBI, o projeto envolveu ações presenciais (como a aplicação de provas simuladas) e à distância, por meio da plataforma Moodle. Os conteúdos trabalhados envolvem o formato dos problemas e as estruturas iniciais de programação de computadores. As atividades são baseadas na resolução de problemas de provas anteriores, disponibilizados no site da OBI. Com a continuidade do projeto, foi possível aumentar o número de participantes no projeto e na competição, bem como melhorar os resultados nas fases da OBI. Por parte dos monitores, o projeto contribui para que desenvolvam habilidades didático-pedagógicas e de relacionamento interpessoal, além de permitir o aprofundamento em conteúdos de programação de computadores necessários ao treinamento.

**Palavras-chave:** ensino de programação; programação de computadores; ensino técnico; Olimpíada Brasileira de Informática.

## Introdução

---

<sup>1</sup> Universidade Estadual do Centro-Oeste (Unicentro), Departamento de Ciência da Computação (DECOMP).

<sup>2</sup> Colégio Estadual Francisco Carneiro Martins (CEFCM), Coordenação Curso Técnico em Informática.

<sup>3</sup> Universidade Estadual do Centro-Oeste, Bacharelado em Ciência da Computação, Fundação Araucária.



“Vivemos em um mundo dominado por software. Nossas chamadas telefônicas ocorrem sobre redes controladas por software; nosso sinal de televisão é entregue pela internet; pessoas não compram mais mapas, elas usam a web; nós compramos online. As próximas gerações serão cada vez mais online e digitais. Em breve, sua casa será controlada por software, alguns dos seus cuidados médicos serão entregues pela web e seu carro poderá se guiar sozinho. (...) Software (...) é a linguagem do nosso mundo. No futuro, não conhecer a linguagem dos computadores será um desafio como ser analfabeto ou analfabeto matemático é hoje” (CROW, 2014).

Sob essa perspectiva, há várias iniciativas em todo o mundo para disseminar os conhecimentos sobre programação de computadores para crianças e jovens. Assim, o projeto de extensão Ensino Extracurricular de Programação de Computadores (E2PC), vinculado ao Departamento de Ciência da Computação da Unicentro, tem desenvolvido ações junto aos cursos Técnicos de Informática em Guarapuava/PR, a fim de contribuir na formação dos alunos em tópicos relacionados à Ciência da Computação e, mais especificamente, à programação de computadores.

Uma das metodologias pedagógicas que têm se mostrado eficazes são as competições de programação, pois permitem aos alunos aplicar conceitos de desenvolvimento de software, desenvolver habilidades de resolução de problemas e adquirir experiência em programação de computadores, incluindo linguagens, técnicas e ferramentas (ambientes). Assim, as competições podem ser vistas como uma metodologia lúdica, em que conceitos de resolução de problemas, lógica, algoritmos, linguagens de programação, desenvolvimento de software, entre outros, são abordados e testados (ICPC, 2018; TROTMAN; HANDLEY, 2008; PIEKARSKI et al., 2015).

Este resumo descreve o treinamento destinado aos alunos do curso Técnico em Informática do Colégio Estadual Francisco Carneiro Martins (CEFCM), em Guarapuava/PR, no biênio 2016-2017, visando melhorar seus conhecimentos em programação de computadores. Especificamente, este treinamento foi destinado a incentivar e preparar os alunos para a Olimpíada Brasileira de Informática, atividade que vem sendo estimulada pela coordenação do curso técnico desde 2015.

## **Metodologia**

A metodologia pedagógica adotada no projeto aqui descrito foi a competição de programação, mais especificamente a metodologia da Olimpíada Brasileira de Informática. Portanto, a seguir, é descrita a OBI, o formato dos problemas e a estrutura de treinamento disponibilizada. São listadas algumas iniciativas similares a este projeto. São apresentadas as ações dos treinamentos realizados no CEFCM em 2016 e em 2017.



## *Olimpíada Brasileira de Informática*

A Olimpíada Brasileira de Informática (OBI), realizada pela Sociedade Brasileira de Computação (SBC), é uma competição organizada nos moldes das outras olimpíadas científicas brasileiras, como Matemática (OBMEP), Física (OBF) e Astronomia (OBA). A OBI tem como objetivo despertar nos alunos o interesse por Ciência da Computação por meio de atividades que envolvem problemas, desafios e competição. A OBI é organizada pelo Instituto de Computação da UNICAMP (OBI, 2018a).

Os objetivos da OBI são (OBI, 2018b):

- Estimular o interesse pela Computação e por Ciências em geral;
- Promover a introdução de disciplinas de raciocínio computacional e técnicas de programação de computadores nas escolas de ensino médio e fundamental;
- Proporcionar novos desafios aos estudantes;
- Identificar talentos e vocações em Ciência da Computação de forma a melhor instruí-los e incentivá-los a seguir carreiras nas áreas de ciência e tecnologia.

Em 2016 e 2017, a OBI foi organizada em três modalidades: Iniciação, Programação e Universitária (OBI, 2016; OBI, 2017)<sup>4</sup>. A modalidade em que os alunos do ensino médio podiam competir é a de Programação, em dois níveis:

- Nível 1: para alunos até o primeiro ano do ensino médio;
- Nível 2: para alunos até o terceiro ano do ensino médio.

O site da OBI<sup>5</sup> disponibiliza, além de informações sobre as provas, ambiente para que os interessados possam se preparar para a competição. Os problemas das provas anteriores está disponível, e é possível submeter o código de um problema para que ele seja avaliado (por um mecanismo de correção automática). Considerando que o foco das ações descritas neste trabalho é voltada para a modalidade Programação, a metodologia dessa modalidade é detalhada.

Os enunciados dos problemas da OBI seguem um padrão próprio das competições.

Para exemplificar, pode-se observar o enunciado da Figura 1, intitulado *Overflow*, da prova da Fase 1 da OBI em 2009. O problema é contextualizado, e algumas vezes nem todas as informações que constam do enunciado são necessárias para a resolução do problema. Portanto, é necessário que os competidores exercitem a capacidade de

---

<sup>4</sup> Para a prova em 2018, houve uma readequação das modalidades.

<sup>5</sup> <https://olimpiada.ic.unicamp.br/>



extrair as informações pertinentes à resolução do problema, diferenciando-as daquelas que simplesmente descrevem o cenário (PIEKARSKI et al., 2015).

As restrições, que são informações sobre o limite das instâncias de teste, também são apresentadas no enunciado. Esse é outro aspecto que causa estranheza aos iniciantes, que com frequência implementam essas restrições para filtrar os dados de entrada. Essas informações são disponibilizadas para dimensionar adequadamente as variáveis (tipo de dados que comportem os limites apresentados) e a eficiência da solução algorítmica. O enunciado contém também exemplos: algumas instâncias de teste para que a solução possa ser testada antes de ser submetida para avaliação.

O código-fonte da Figura 2, em linguagem C, é uma das possíveis soluções para o problema *Overflow*. Para submeter a solução desenvolvida para avaliação, é preciso acessar a página do problema, selecionar a linguagem utilizada no início da página e anexar o arquivo fonte, como pode ser visto na Figura 3.



## Overflow

Os computadores foram inventados para realizar cálculos muito rapidamente, e atendem a esse requisito de maneira extraordinária. Porém, nem toda conta pode ser feita num computador, pois ele não consegue representar todos os números dentro de sua memória. Em um computador pessoal atual, por exemplo, o maior inteiro que é possível representar em sua memória é 4.294.967.295. Caso alguma conta executada pelo computador dê um resultado acima desse número, ocorrerá o que chamamos de *overflow*, que é quando o computador faz uma conta e o resultado não pode ser representado, por ser maior do que o valor máximo permitido (em inglês *overflow* significa transbordar). Por exemplo, se um computador só pode representar números menores do que 1023 e mandamos ele executar a conta  $1022 + 5$ , vai ocorrer *overflow*.

### Tarefa

Dados o maior número que um computador consegue representar e uma expressão de soma ou multiplicação entre dois inteiros, determine se ocorrerá *overflow*.

### Entrada

A primeira linha da entrada contém um inteiro  $N$  representando o maior número que o computador consegue representar. A segunda linha contém um inteiro  $P$ , seguido de um espaço em branco, seguido de um caractere  $C$  (que pode ser '+' ou '\*', representando os operadores de adição e multiplicação, respectivamente), seguido de um espaço em branco, seguido de um outro inteiro  $Q$ . Essa linha representa a expressão  $P + Q$ , se o caractere  $C$  for '+', ou  $P \times Q$ , se o caractere  $C$  for '\*'.

### Saída

Se programa deve imprimir uma única linha, contendo a palavra 'OVERFLOW' se o resultado da expressão causar um *overflow*, ou a palavra 'OK' caso contrário. Ambas as palavras devem ser escritas com letras maiúsculas.

### Restrições

- $1 \leq N \leq 500.000$
- $0 \leq P \leq 1000$
- $0 \leq Q \leq 1000$

### Informações sobre a pontuação

- Em um conjunto de casos de teste que totaliza 30 pontos,  $N \leq 10$ ,  $P \leq 10$  e  $Q \leq 10$ .
- Em um conjunto de casos de teste que totaliza 80 pontos,  $N \leq 100$ ,  $P \leq 100$  e  $Q \leq 100$ .

### Exemplos

Entrada	Saída
10 5 + 5	OK
44 23 * 2	OVERFLOW
323500 42 * 35	OK

Figura 1 - Enunciado do problema *Overflow*.

Fonte: <https://olimpiada.ic.unicamp.br/pratique/p1/2009/f1/overflow/>

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5     int N, P, Q;
6     char C;
7
8     scanf("%d%d %c", &N, &P, &C, &Q);
9     if(C == '+'){
10        printf("%s", (((P + Q) > N) ? "OVERFLOW" : "OK"));
11    }else{
12        if(C == '*'){
13            printf("%s", (((P * Q) > N) ? "OVERFLOW" : "OK"));
14        }
15    }
16
17    return 0;
18 }
```

Figura 2 - Resolução em linguagem C do problema *Overflow*.

Submeta sua solução

Linguagem:  Arquivo:

### Overflow

Os computadores foram inventados para realizar cálculos muito rapidamente, e atendem a esse requisito de maneira extraordinária. Porém, nem toda conta pode ser feita num computador, pois ele não consegue representar todos os números dentro de sua memória. Em um computador pessoal atual, por exemplo, o maior inteiro que é possível representar em sua memória é 4.294.967.295. Caso alguma conta executada pelo computador dê um resultado acima desse número, ocorrerá o que chamamos de *overflow*, que é quando o computador faz uma conta e o resultado não pode ser representado, por ser maior do que

Tarefas Programação Nível Júnior

- 2017
  - Fase 1
    - Bondinho
    - Drone de Entrega
  - Fase 2
    - Cartas
    - Jogo de Tabuleiro
    - Montanha
  - Fase 3
    - Gomoku

Figura 3 - Submissão da solução do problema *Overflow*.

O problema anexado será julgado automaticamente por um software de autojulgamento usado no site. A avaliação automática das soluções fornece os dados de entrada ao programa e compara os dados de saída gerados com os dados de saída corretos. Esse padrão de avaliação (por comparação de entradas e saídas) é utilizado por diferentes



juizadores automáticos, tanto em competições (como por exemplo a Maratona de Programação (MARATONA, 2018)) como em sites de treinamento (URI Online Judge<sup>6</sup>, Sphere Online Judge<sup>7</sup>). Após a avaliação da solução, o usuário é redirecionado a uma página (Figura 4), na qual pode-se visualizar o resultado de sua submissão.

```
Resultado da submissão

TAREFA: Overflow
LINGUAGEM: C

Compilação correta

Fase de testes -- Tempo Limite para cada execução: 1000 ms
Teste 1: ... (5 pontos)
Teste 2: . (5 pontos)
Teste 3: . (5 pontos)
Teste 4: . (5 pontos)
Teste 5: . (5 pontos)
Teste 6: . (5 pontos)
Teste 7: . (5 pontos)
Teste 8: . (5 pontos)
Teste 9: . (5 pontos)
Teste 10: . (5 pontos)
Teste 11: . (5 pontos)
Teste 12: . (5 pontos)
Teste 13: . (5 pontos)
Teste 14: . (5 pontos)
Teste 15: . (5 pontos)
Teste 16: . (5 pontos)
Teste 17: . (5 pontos)
Teste 18: . (5 pontos)
Teste 19: . (5 pontos)
Teste 20: . (5 pontos)

Total: 100 pontos (de 100 possíveis)

Legenda:
.: resultado correto X: resultado incorreto T: tempo limite excedido
R: erro em tempo de execução V: programa não gerou saída
S: violação de recursos, interrompido pelo sistema
```

Figura 4 - Tela com o resultado da avaliação.

Analisar o resultado da solução é de grande importância para verificar se a solução submetida está correta ou se o algoritmo apresentou algum problema antes, durante ou depois da execução. Uma solução bem sucedida é compilada com sucesso e alcança a totalidade dos pontos possíveis. O resultado indica, para cada caso de teste, se o algoritmo apresentou erros: conforme a legenda, X significa que a saída produzida pela solução submetida não coincide com o resultado esperado (saídas pré-definidas). Outras falhas podem ser:

<sup>6</sup> URI Online Judge - [www.urionlinejudge.com.br](http://www.urionlinejudge.com.br)

<sup>7</sup> Sphere Online Judge - <http://br.spoj.com/>



- T: significa que o programa não foi executado dentro do limite de tempo estabelecido; extrapolações de tempo podem indicar que a solução deve ser mais eficiente (poupar tempo de execução);
- R: significa que a execução do programa foi interrompida antes de terminar; ocorre por exemplo quando há divisão por zero, estouro de pilha ou falha de segmentação;
- V: significa que o programa não gerou saída; isso ocorre quando esquecemos de utilizar funções de saída de dados no algoritmo;
- S: significa violação de recursos; nesse caso o programa é interrompido pelo ambiente, para garantir a integridade dos outros processos que estão em funcionamento.

Caso a solução não alcance a totalidade dos pontos, é possível, a partir da avaliação disponível, corrigir os erros do código. A disponibilidade de avaliar uma solução a qualquer momento contribui com aqueles alunos que querem treinar de forma autônoma. No caso do treinamento aqui descrito, nos valemos desse recurso para incentivar os participantes a resolver os exercícios propostos de forma complementar às atividades presenciais.

### *Iniciativas similares*

No Departamento de Ciência da Computação da UFRJ (Universidade Federal do Rio de Janeiro), o treinamento preparatório é destinado a alunos do primeiro ano do curso, que podem participar da modalidade Universitária. Todos os alunos são encorajados a participar do treinamento, mesmo que não tenham a intenção de disputar a prova. O treinamento é dividido em duas partes. A primeira é destinada à linguagem C, desde o processo de compilação até funções e *strings*. A segunda parte é voltada para a competição: inclui o formato da prova e as técnicas para a resolução dos problemas (estruturas de dados, recursão, ordenação, programação dinâmica, etc) (UFRJ, 2012).

Maydana et al. (2009) relatam um treinamento disponibilizado no Campus Tecnológico de Alegrete (Unipampa) para os alunos dos últimos anos do ensino médio do município. Tendo como pano de fundo a participação na OBI, o treinamento visa desenvolver o raciocínio lógico e abstrato dos participantes por meio dos conceitos de programação de computadores, linguagens de programação e resolução de problemas. Os instrutores, graduandos em Ciência da Computação, foram capacitados para o treinamento, em especial quanto ao formato dos problemas das competições.



Para os alunos de um curso Técnico de Informática (nível médio) e os acadêmicos do primeiro ano de Engenharia de Computação da UEPG, foi oferecido um treinamento preparatório para a OBI por meio de um projeto de extensão (SANTOS; FERRASA; SOUZA, 2017). O treinamento consistiu de uma palestra de divulgação para os alunos do curso técnico e quatro aulas. Nas aulas, foram abordados os diversos temas e técnicas recorrentes nas provas da OBI, bem como o modelo de questões da competição. Os autores comentam sobre os entraves, relacionados à dificuldade de locomoção (por parte dos alunos do ensino médio), escassez de tempo por excesso de outras atividades (por parte dos acadêmicos da UEPG) e falta de interesse (por parte dos dois públicos).

No Instituto de Informática da UFG o treinamento é oferecido aos alunos do 8º e 9º anos do ensino fundamental, a fim de prepará-los para a OBI. O foco do treinamento é para a modalidade de Programação, em linguagem C++. A plataforma Moodle é usada para apoio ao treinamento (UFG, 2017).

Também estão crescendo, por iniciativas de ex-participantes de olimpíadas e maratonas científicas, plataformas de treinamento e aconselhamento na web, abertas a todos os interessados. O Núcleo Olímpico de Incentivo ao Conhecimento (Noic)<sup>8</sup> e o GuaraBots<sup>9</sup> são exemplos desses ambientes.

#### *Treinamento para a OBI no CEFCM em 2016*

Desde 2011, quando teve início o projeto de extensão E2PC, havia a proposta de desenvolver ações para apoiar os alunos de ensino médio interessados em participar da OBI. No Colégio Estadual Francisco Carneiro Martins (CEFCM), a coordenação do curso Técnico em Informática tem estimulado os alunos a participar da OBI desde 2015. Em 2016, graças à parceria estabelecida em outro projeto, o E2PC iniciou as atividades de treinamento de alunos do curso técnico interessados em participar da OBI.

Os treinamentos visam fornecer subsídios quanto aos conteúdos e metodologia própria da OBI. A ementa da OBI para a modalidade Programação (OBI, 2018c) é bastante extensa e complexa, até mesmo para estudantes de graduação. Como a disponibilidade de tempo para o

treinamento é restrita, a estratégia adotada foi sedimentar os conhecimentos dos alunos do curso técnico, reforçando os conceitos iniciais de programação, detalhando o formato dos problemas da competição e fazendo-os praticar.

---

<sup>8</sup> <http://noic.com.br>

<sup>9</sup> <https://guarabots.wordpress.com>

Quanto à linguagem de programação, considerando a escassez de tempo, foi mantida a linguagem Pascal, que os participantes já conheciam (linguagem adotada nas disciplinas de programação do curso técnico).

Na primeira edição, em 2016, o treinamento foi constituído por algumas atividades presenciais e outras à distância, possibilitando aos interessados treinar conforme sua disponibilidade. As atividades presenciais ocorreram no Colégio, a fim de motivar para a participação na OBI, instruir sobre o treinamento, sondar as habilidades e dúvidas dos participantes e discutir exercícios propostos.

Para as atividades à distância, foi desenvolvida a “Plataforma de apoio ao treinamento para a Olimpíada Brasileira de Informática”, utilizando o ambiente virtual de aprendizagem institucional da Unicentro<sup>10</sup>, que usa o Moodle (*Modular Object-Oriented Dynamic Learning Environment*) para implementar o ambiente virtual (PADILHA et al., 2016). Moodle é um sistema de gerenciamento de aprendizagem gratuito e *open-source*. A plataforma foi organizada em tópicos (ver Figura 5), a saber:

- Primeiro tópico: contém uma descrição do site da OBI, detalhando o formato dos problemas e o processo de correção automática (como encontrar os exercícios indicados, como salvar os programas desenvolvidos no formato para submissão, como submeter uma solução para avaliação automática no site);
- Segundo tópico: o conteúdo é destinado ao formato dos dados de entrada e saída e quais funções (na linguagem Pascal) devem ser utilizadas;
- Terceiro tópico: sobre as estruturas de seleção;
- Quarto tópico: sobre as estruturas de repetição;
- Quinto tópico: destinado a vetores (estruturas de armazenamento homogêneas).

---

<sup>10</sup> <https://moodle.unicentro.br>





Figura 5 - Tópicos da Plataforma de treinamento no Moodle da Unicentro.

Em cada tópico, há uma videoaula explicando o tema abordado e um problema é proposto. Os participantes devem submeter a solução para o problema, podendo utilizar o fórum para tirar dúvidas. Somente depois que a submissão da solução é feita, outra videoaula explicando a solução do problema é disponibilizada e são indicados exercícios de fixação (PADILHA et al., 2016).

Semanalmente um novo tópico foi disponibilizado no Moodle. Todas as resoluções postadas foram avaliadas pela equipe de coordenação do projeto, que testou e deu retorno sobre possíveis melhorias e/ou falhas da solução. Além das atividades à distância, foi disponibilizado um horário semanal para atendimento dos interessados.

Em 2016, toda a turma do terceiro ano do curso técnico foi convidada a participar do treinamento. Dos 47 alunos que se inscreveram para participar, 36 responderam a primeira atividade do Moodle. À medida que a complexidade dos problemas aumentava, o número de participantes foi diminuindo. Na segunda atividade, 15 alunos postaram a resolução dos exercícios; na terceira, foram oito alunos; e na última, apenas cinco participantes enviaram a solução do problema proposto para avaliação.

Foi realizada uma competição simulada, como atividade presencial do treinamento. A prova consistiu de quatro problemas envolvendo os conceitos que haviam sido disponibilizados na plataforma do Moodle. Foram 16 participantes. Nenhum conseguiu resolver integralmente nenhum dos problemas. Quatro participantes conseguiram atingir 35 pontos da solução de um problema. Em média, foram duas tentativas de solução por participante. Um dos participantes não implementou nenhuma solução. Os erros mais comuns foram lógica incorreta na solução dos problemas (incluindo a não utilização das estruturas adequadas) e uso inadequado das instruções de leitura de dados.

Para corrigir essas falhas, foi realizada uma aula presencial de resolução de exercícios, para comentar os erros mais frequentes e como isso deve ser evitado nas competições. Dos sete alunos que prestaram a prova da primeira fase, um foi selecionado para segunda fase. Foi disponibilizado atendimento, a fim de apoiar esse aluno na resolução de problemas de provas antigas, melhorando suas habilidades para a segunda fase.

#### *Treinamento para a OBI no CEFCM em 2017*

Em 2017, no CEFCM, apenas os alunos do terceiro ano estavam aptos a participar, sendo enquadrados na Modalidade Programação, Nível 2.

A fim de identificar o nível de conhecimento e as dificuldades dos participantes do treinamento (toda a turma do terceiro ano foi convidada), a primeira atividade realizada foi a aplicação de uma prova simulada como forma de sondagem. Os exercícios foram selecionados de provas anteriores, envolvendo conceitos básicos de programação (programação sequencial, estrutura condicional, estrutura de repetição e vetores). Os exercícios, do nível Júnior, foram: OBI<sup>11</sup>, Sinuca<sup>12</sup>, Notas<sup>13</sup> e Volume da TV<sup>14</sup>.

O exercício OBI envolve estruturas repetição e seleção. Dado o número de competidores, o número mínimo de pontos requeridos para ganhar o prêmio da OBI e a pontuação da primeira e segunda fase de cada competidor, o programa deve apresentar o número de competidores que ganharão o prêmio. Para isso, é preciso que os pontos da primeira e da segunda fase de cada competidor seja maior ou igual ao mínimo de pontos requerido.

O problema Sinuca envolve vetores (e, obviamente, estruturas de seleção e repetição). Trata-se de um passatempo envolvendo bolas de sinuca pretas e brancas. A distribuição

---

<sup>11</sup> <https://olimpiada.ic.unicamp.br/pratique/pj/2008/f1/obi/>

<sup>12</sup> <https://olimpiada.ic.unicamp.br/pratique/pj/2014/f2/sinuca/>

<sup>13</sup> <https://olimpiada.ic.unicamp.br/pratique/p2/2014/f2/notas/>

<sup>14</sup> <https://olimpiada.ic.unicamp.br/pratique/pj/2013/f2/volume/>



das cores das bolas segue uma regra. Identificada essa regra, a tarefa é apresentar qual será a cor do último elemento.

O exercício Notas também envolve vetores, estruturas de seleção e repetição. O exercício pede um programa que, dado o número de alunos e a nota de cada um, informe qual é a nota mais recorrente.

O problema Volume da TV envolve estruturas de seleção e repetição. O exercício pede o desenvolvimento de um programa que, dado o volume inicial, o número de trocas de volume e as modificações do volume, mostre qual o volume da TV após todas essas trocas.

Apenas seis alunos participaram dessa atividade, o que possibilitou que os monitores fizessem um acompanhamento bem personalizado, observando os participantes e identificando suas dificuldades.

Identificadas as deficiências, foi reativada e reutilizada a “Plataforma de apoio ao treinamento para a Olimpíada Brasileira de Informática” no Moodle. Foi incluído um novo tópico à plataforma, incluindo exercícios envolvendo as estruturas abordadas.

### **Resultados e Discussão**

Para os participantes do treinamento em 2016 foi aplicada uma avaliação para 44 dos alunos que haviam se inscrito para o treinamento. Desses, 19 não participaram de nenhuma das atividades (disseram não ter disponibilidade de tempo ou interesse). Dos que participaram, cinco avaliaram o treinamento como “muito bom”, 18 avaliaram como “bom” e dois avaliaram como “ruim”.

Em 2017, na primeira atividade (prova simulada) ficou em evidência a falta de prática com a linguagem de programação, especialmente no que se trata da sintaxe das instruções. Os alunos participantes não estavam mais em contato com a linguagem, pois não tinham mais disciplinas de programação naquele ano. Outra dificuldade dos participantes que se repete é o uso adequado das funções de entrada e saída. Tomando isso como base, foram disponibilizados no Moodle problemas similares aos do simulado, para que os participantes praticassem. Os monitores acompanharam os acessos, postagens e dúvidas por meio da plataforma de treinamento (no Moodle).

Posteriormente, às vésperas da realização da prova da OBI, foi realizada uma atividade presencial, destinada a elucidar dúvidas. Infelizmente, poucos participantes se dedicaram (a maioria nem sequer havia acessado a plataforma Moodle) e puderam aproveitar a presença dos monitores para auxiliá-los em problemas que não haviam conseguido resolver.



Dos dez participantes que prestaram a primeira fase da OBI, três foram selecionados para a segunda fase. A coordenação do curso técnico solicitou, então, apoio para o treinamento desses selecionados, com prazo bastante restrito. Foram realizadas duas monitorias nas dependências do DECOMP, destinadas a melhorar as habilidades dos alunos na resolução dos problemas, com conteúdos que eles já dominavam. Em cada uma dessas monitorias presenciais, foram resolvidos os problemas que já haviam sido propostos no Moodle.

Ainda, um dos participantes foi selecionado para a fase nacional da prova. Com um pouco mais de tempo, novamente foram realizadas atividades presenciais. Na primeira monitoria, foi explicado o conceito de matrizes (estruturas bidimensionais que armazenam elemento de um mesmo tipo de dado, acessíveis pelos índices das linhas e colunas) (MANZANO; OLIVEIRA, 2016), que o aluno ainda não conhecia. A partir disso, o treinamento foi destinado aos exercícios envolvendo matrizes, tais como Torre<sup>15</sup>, Tabuleiro Esburacado<sup>16</sup> e Quadrado Mágico<sup>17</sup>.

O exercício Torre pede a identificação e apresentação do maior valor, em uma matriz, referente à combinação de cada cruzamento de linhas e colunas, excluindo o valor pertencente à intersecção (a resolução desse problema de acordo com o tempo de execução esperado requer técnica mais avançada de programação, que não é trabalhada no curso técnico).

O exercício Tabuleiro esburacado apresenta uma estrutura em matriz simulando um tabuleiro de xadrez, porém quatro casas são buracos. Dada a sequência de movimentos que o cavalo deve executar, o desafio é simular um passeio da peça pelo tabuleiro e determinar quantos movimentos o cavalo faz até chegar ao seu destino ou até cair em um buraco.

O exercício Quadrado Mágico apresenta o conceito que dá nome ao exercício. Um quadrado mágico é uma matriz cuja soma dos elementos de uma das linhas, colunas ou diagonais principais sempre será o mesmo. O objetivo do programa é identificar se uma matriz inserida é um quadrado mágico.

As dificuldades enfrentadas pelo aluno nessa última prova eram esperadas, pois a maior parte dos exercícios envolve técnicas que não são contempladas na grade do ensino técnico. Ainda, o tempo disponível para o treinamento não foi suficiente para abordar esses conteúdos.

---

<sup>15</sup> <https://olimpiada.ic.unicamp.br/pratique/p1/2015/f2/torre/>

<sup>16</sup> <https://olimpiada.ic.unicamp.br/pratique/p2/2012/f1/cavalo/>

<sup>17</sup> <https://olimpiada.ic.unicamp.br/pratique/p2/2011/f2/magico/>



## Considerações Finais

O treinamento realizado possibilitou aos participantes praticar as habilidades em programação de computadores, pois a partir do terceiro ano, a grade do curso técnico que frequentam não possui disciplina com esse conteúdo específico. Vale destacar a classificação de um dos participantes para a fase nacional da OBI. Esse treinamento, bem como outras ações do projeto E2PC, pretendem motivar os participantes para que continuem na área da computação.

Por parte dos monitores, o projeto permitiu o contato com os estudantes do ensino médio. Além do conteúdo ser extremamente pertinente com a área de formação (foram necessários os conhecimentos de programação de computadores), os monitores desenvolveram habilidades didático-pedagógicas. Ainda, os monitores tiveram que aprender a linguagem Pascal, com a qual não tinham experiência.

Utilizar a plataforma Moodle para a disponibilização dos conteúdos e exercícios foi importante por ser de fácil acesso, replicável, e constituir base para futuras edições do projeto. Outros projetos podem se beneficiar da plataforma de treinamento, tanto em ações de ensino quanto de extensão.

Sob o ponto de vista das premissas extensionistas, o projeto cumpre os requisitos de levar conhecimentos técnicos à comunidade externa e possibilitar a complementação da formação dos discentes que atuam como monitores (por permitir o contato com a comunidade e por realizar práticas distintas daquelas da área de formação de bacharel). Ainda, aproxima a comunidade atendida do espaço universitário, mostrando a possibilidade de participar dele como aluno em um futuro próximo. Vale lembrar que o projeto iniciou para atender a solicitação da coordenação do curso, que identificou a necessidade de melhorar a qualificação dos seus alunos para a competição. Como consequência, os participantes melhoraram sua formação quanto à resolução de problemas e programação de computadores.

## Referências

CROW, D. Why every child should learn to code. *The Guardian*, 2014. Disponível em: <[www.theguardian.com/technology/2014/feb/07/year-of-code-dan-crow-songkick](http://www.theguardian.com/technology/2014/feb/07/year-of-code-dan-crow-songkick)>.

Acesso em: 30 mar. 2018.

ICPC. *International Collegiate Programming Contest*. 2018. Disponível em: <<http://icpc.baylor.edu>>. Acesso em: 23 fev. 2018.

MANZANO, J. A. N. G.; OLIVEIRA, J. F. de. *Algoritmos: lógica para desenvolvimento de programação de computadores*. 28. ed. São Paulo: Érica-Saraiva, 2016. 336 p.



MARATONA. *O que é?*. 2018. Disponível em: <<http://maratona.ime.usp.br/info17.html>>. Acesso em: 30 mar. 2018.

MAYDANA, G. S.; GIDRÃO, Y. M. H.; MACHADO JUNIOR, O. R.; PINTO, M. C. Treinamento para a Olimpíada Brasileira de Informática (OBI). In: SALÃO INTERNACIONAL DE ENSINO, PESQUISA E EXTENSÃO, 1., 2009, Uruguaiana/RS. *Anais do...* Uruguaiana: Unipampa, 2009.

OBI. *OBI* 2016. 2016. Disponível em: <<https://olimpiada.ic.unicamp.br/passadas/OBI2016/>>. Acesso em: 30 mar. 2018.

OBI. *OBI* 2017. 2017. Disponível em: <<https://olimpiada.ic.unicamp.br/passadas/OBI2017/>>. Acesso em: 30 mar. 2018. OBI. *Sobre a OBI*. 2018a. Disponível em: <<https://olimpiada.ic.unicamp.br/info/>>. Acesso em: 30 mar. 2018.

OBI. *Regulamento*. 2018b. Disponível em: <<https://olimpiada.ic.unicamp.br/info/regulamento/>>. Acesso em: 30 mar. 2018.

OBI. *Ementa Modalidade Programação*. 2018c. Disponível em: <<https://olimpiada.ic.unicamp.br/info/ementa/geral/>>. Acesso em: 30 mar. 2018.

PADILHA, L.; DIDUR, L. F.; HILD, T. A.; MIAZAKI, M.; PIEKARSKI, A. E. T. Plataforma de apoio ao treinamento para a Olimpíada Brasileira de Informática. In: SALÃO DE EXTENSÃO E CULTURA, 9., 2016, Guarapuava/PR. *Anais...* Guarapuava: Unicentro, 2016.

PIEKARSKI, A. E. T.; MIAZAKI, M.; HILD, T. A.; MULATI, M. H.; KIKUTI, D. A metodologia das maratonas de programação em um projeto de extensão: um relato de experiência. In: WORKSHOP DE ENSINO EM PENSAMENTO COMPUTACIONAL, ALGORITMOS E PROGRAMAÇÃO, 1., 2015, Maceió/AL. *Anais dos Workshops do IV Congresso Brasileiro de Informática na Educação*. Porto Alegre: SBC, 2015. p. 1246-1254.

SANTOS, L. M.; FERRASA, M.; SOUZA, M. A. Preparação para a prova de programação da Olimpíada Brasileira de Informática (OBI) em escolas de ensino médio e universitário. In: ENCONTRO CONVERSANDO SOBRE EXTENSÃO NA UEPG, 15., 2017, Ponta Grossa/PR. *Anais do 15º CONEX*. Ponta Grossa: UEPG, 2017.

TROTMAN, A.; HANDLEY, C. Programming contest strategy. *Computers & Education*, N. 50, Vol 3, p. 821–837.

UFG. *Seleção para o treinamento*. 2017. Disponível em: <<https://sites.google.com/site/obiufg/inscricoes>>. Acesso em: 30 mar. 2018.

UFRJ. *Competições de algoritmos e programação*. 2012. Disponível em: <<http://maratona.dcc.ufrj.br/obi/>>. Acesso em: 30 mar. 2018.

