

Uma avaliação sobre o desempenho de alunos do IFPB em olimpíadas de informática

Katyusco de Farias Santos

katyusco@ifpb.edu.br. Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - Campus Campina Grande.

RESUMO

Como forma de angariar fundos e treinar alunos para participar de Olimpíadas de Informática (OI), o grupo do Projeto Olímpico de Programação, do IFPB, Campus Campina Grande (POP-CG), passou a realizar simulados de OI. Para tanto, utilizou-se o serviço de um julgador online para criação de competições de programação, o *ForCode* (<https://github.com/LADOSSIFPB/POP-Judge>). Minerar dados de desempenho de alunos (número de submissões e de acertos, linguagem de programação utilizada), a partir do repositório de código gerado pelo *ForCode*, pode revelar habilidades de programação que devem ser lapidadas. Associar a essas informações dados holísticos sobre os alunos (e.g.: ano de escolaridade, situação socioeconômica, entre outros) pode indicar quais fatores impactam, de forma benéfica ou não, no desempenho desses alunos. Neste artigo, conglomeramos dados de desempenho de 47 alunos participantes do simulado de OI, realizado em novembro de 2015, com um dado holístico coletado – o ano de escolaridade – a partir do sistema da gestão acadêmica do IFPB. Em seguida, avaliamos estatisticamente qual a relação do desempenho no simulado de OI com ano de escolaridade. Os resultados apontam que, independentemente da linguagem de programação utilizada, a estratégia de aprendizado “tentativa e erro” é a que prevalece entre os alunos com melhor desempenho. Quando considerada a escolaridade, a média dos acertos dos alunos do 2º ano é 15% maior que a média obtida pelos alunos do 1º ano; já a média de acertos dos alunos do 4º ano é 78% maior que a dos alunos do 2º ano e 105% maior que as obtidas pelos do 1º ano.

Palavras-chave: Olimpíadas de Informática. Mineração Repositório. Engenharia de Software Empírica.

ABSTRACT

In order to recruit and train students to participate in Olympiads of Informatics (OI), the Olympic Programming Project group of IFPB Campina Grande Campus (POP-CG) has started to perform OI simulations. For this purpose, an online judge was used to create programming competitions, ForCode (<https://github.com/LADOSSIFPB/POP-Judge>). Mining students' performance data (number of submissions and hits, programming language used) from the code repository generated by ForCode may reveal what programming skills should be improved. Associating this information with holistic data about students (e.g.: schooling, socioeconomic status, and so on) may indicate which factors impact (positively or negatively) on the performance of these students. In this article, we joined performance data of 47 students participating in the OI simulation, conducted in November 2015, to holistic data collected from the IFPB academic management system: the year of schooling. Then, we evaluated statistically the relation between the performance's simulated OI with year of schooling. The results point out that, regardless of programming language used, the "trial and error" learning strategy is the one that prevails among the students with the best performance. When considering schooling, the average score of the students in the 2nd year is 15% higher than the average of the students in the 1st year, while the average score of the students in the 4th year is 78% higher than those of the students of the 2nd year 105% higher than those obtained in the 1st year.

Keywords: *Computer Olympics. Mining of Repository. Empirical Software Engineering.*

1 Introdução

Nos últimos anos, alunos talentosos do Instituto Federal da Paraíba-IFPB, Campus Campina Grande, têm conquistado várias medalhas, inclusive de ouro, na Olimpíada Paraibana de Informática-OPI (Olimpíada Paraibana de Informática, n.d.) bem como têm obtido posições de destaque na versão nacional, a Olimpíada Brasileira de Informática-OBI (Olimpíada Brasileira de Informática, n.d.). Em 2014, a medalha de prata obtida na OBI por um aluno do Campus Campina Grande credenciou, pela primeira vez, a participação do IFPB na Olimpíada Internacional de Informática-IOI.

Conquistas como essas revelam o potencial intelectual dos alunos e demonstram a qualidade do ensino tecnológico oferecido pelo IFPB-CG. Além disso, tais conquistas geram mídia espontânea positiva para a Rede Federal de Ensino Tecnológico-REDETEC.

Para que novas conquistas ocorram, alunos da área de informática e afins devem ser incentivados a participarem de olimpíadas (CGEE, 2011). Precisam também ser submetidos a treinos. E, para que talentos intelectuais sejam descobertos e lapidados, é preciso identificar tanto os fatores que propiciam quanto aqueles que dificultam o bom desempenho dos estudantes em Olimpíadas de Informática-OI.

O grupo do Projeto Olímpico de Programação do IFPB, Campus Campina Grande (POP-CG), faz uso de um simulador de OI para que alunos pudessem ter contato com a dinâmica e os tipos de desafios de programação com os quais eles se deparariam durante as OI reais. O uso desse tipo de ferramenta gera um repositório com dados de desempenho de alunos, composto de códigos fontes por eles submetidos, número de tentativas de submissões por desafio, linguagem de programação utilizada etc. Minerar esse repositório para observar as estratégias de participação do aluno pode nos revelar que competências devem ser aperfeiçoadas, em prol da melhoria de seu desempenho (MORAIS, 2015; ROBINS, 2010).

O fato é que o bom desempenho dos alunos em OPI reais não depende apenas dos treinos a que são submetidos em OI simuladas. É preciso avaliar o estudante holisticamente para identificar os reais fatores que catalisam ou inibem o desenvolvimento de suas habilidades em programação (DEHNADI, BORNAT, 2006; MORAIS, 2015). Formação educacional, conhecimentos técnico-artísticos anteriores, situação socioeconômica, estado de saúde clínico-psicológico,

prática esportiva, por exemplo, devem também ser considerados como fatores que podem influenciar no desempenho dos estudantes que participam de OI.

Nesse contexto, este artigo avalia o desempenho dos alunos participantes do simulado da OI, realizada em novembro de 2015, no Campus acima citado. Esse simulado contou com a participação de 47 alunos, que geraram 317 submissões de códigos fontes para 10 questões desafios. Além das medidas extraídas do repositório de código, coletamos também um dado externo à OI, mas holístico a cada participante: sua escolaridade. Sobre essa massa de dados, realizamos uma exploração estatística.

As evidências apontam que, independentemente da linguagem de programação, os estudantes que mais tentativa por desafio submetem são os que obtêm o melhor desempenho. E, à medida que aumenta a escolaridade em anos do aluno do ensino técnico melhor também é seu desempenho. Por exemplo, a média de acertos dos alunos do 4º ano é 105% maior que a obtida pelos do 1º ano.

Este artigo está dividido da seguinte forma: após a Introdução, já discutida, a seção 2, de Fundamentação teórica, em que apresentamos conceitos sobre simulação de olimpíadas, mineração de repositório e sobre fatores que impactam no desempenho dos alunos. Na Seção 3 apresentamos as questões pertinentes a esta pesquisa. Todas as etapas que compõem a metodologia para realizar a Avaliação do desempenho dos alunos estão descritas na Seção 4. Nas Seções 5 e 6, respectivamente, apresentamos nossas Conclusões e relatamos nossos trabalhos futuros. Indicamos, na Seção 7 como este estudo pode ser reproduzido e evoluído.

2 Fundamentação teórica

2.1 Simulação de olimpíadas de informática

OIs são competições organizadas nos moldes das outras olimpíadas científicas, como Matemática, Física, entre outras. O objetivo das OIs é despertar nos alunos o interesse por uma ciência importante na formação básica da era da informação (no caso, ciência da computação), por meio de uma atividade que envolve desafio, engenhosidade e uma saudável dose de competição. A organização da versão brasileira, denominada de Olimpíada Brasileira de Informática, OBI, está a cargo do Instituto de Computação

da UNICAMP, sob a égide da Sociedade Brasileira de Computação (SBC).

Assim como nas demais olimpíadas científicas, para participar, é preciso ser aluno de uma instituição de ensino, logo o desempenho dos alunos traz consigo a chancela da instituição da qual o aluno faz parte. E, para sentirem-se bem representados, universidades e institutos constituem grupos internos que se dedicam ao treinamento de suas equipes para que alcancem resultados de destaque.

Com o objetivo de angariar melhores resultados na OBI, o grupo POP-CG concebeu e desenvolveu um sistema software que desempenha o serviço de um julgador online para criação de competições de programação, o *ForCode* (<https://github.com/LADOS-SIFPB/POP-Judge>). Este serviço simula um ambiente real de olimpíada de informática em que alunos competidores, com diferentes níveis de conhecimento de acordo com o ano escolar em curso, submetem implementações de código, em diferentes linguagens (Java, C, C++, Pascal e Python), na tentativa de solucionar desafios de programação gerados pelo serviço. A cada submissão do aluno competidor, o serviço julgador relata, de forma automática e quase instantânea, se o desafio foi ou não resolvido. Similar ao regulamento das olimpíadas reais de informática, os competidores que resolvem o maior número de desafios, com menor número de submissões e no menor tempo são os que apresentam a melhor performance e, conseqüentemente, são os vencedores.

2.2 Mineração de repositório de OI

Pesquisas em Engenharia de Software têm se utilizado do empirismo, observação e monitoração sobre projetos de software, para investigar características inerentes tanto aos códigos fontes como a seus autores (SANTOS, 2015).

Uma das técnicas utilizadas é o processamento dos dados coletados a partir da mineração dos repositórios de código fontes. Minerar significa coletar os registros de todas as modificações realizadas sobre os códigos fontes que compõem um programa de computador. O processamento, por sua vez, consiste em extrair medidas e aferir métricas de uma dada versão do programa. Os resultados obtidos nos possibilitam, por exemplo: avaliar a qualidade do código construído (MCCABE, 1976), compreender o processo de desenvolvimento aplicado (XIE; CHEN; NEAMTIU, 2009), identificar especialistas técnicos de cada parte do código (SANTOS, GUERRERO,

FIGUEIREDO, 2015; VENKATARAMANI *et al.*, 2013). Além disso, da perspectiva dos desenvolvedores analisar históricos ajuda a evitar erros anteriormente cometidos e endossa a repetição dos acertos, em prol do processo ou do produto software.

A Engenharia de Software se utiliza da mineração de repositórios de projetos de software para aprimorar seu conhecimento (PARNAS, 1994; SANTOS, 2015; XIE, CHEN, NEAMTIU, 2009). Da mesma forma, é possível extrair do repositório do *ForCode* informações que nos revelam características predominantes nos códigos desenvolvidos tanto por alunos que se destacam nas simulações quanto por aqueles que precisam de aprimoramentos na sua formação em programação.

As conclusões extraídas da análise sobre essas informações devem fomentar o aprimoramento do ensino de programação e potencializar a performance dos alunos tanto em OI como também no mundo produtivo da Tecnologia da Informação. Além disso a melhoria no desempenho dos alunos pode impactar, indiretamente, na qualificação técnica de professores bem como na valorização e visibilidade mercadológica das instituições das quais estudantes e docentes fazem parte.

2.3 Fatores impactantes no desempenho

A preocupação com a qualidade da educação não é recente. Em todo o mundo são realizadas pesquisas com a finalidade de elevar a qualidade da educação em função dos resultados positivos sobre o crescimento econômico e redução das desigualdades sociais (PINTO; TENÓRIO, 2014).

Pesquisas educacionais em vários países apontam a existência de grande correlação entre o desempenho dos alunos e fatores externos. Entre esses fatores destacam-se situação socioeconômica de suas famílias, escolaridade e nível da escola (ANDRADE; LAROS, 2007), o envolvimento com drogas (TAVARES; BÉRIA; LIMA, 2004), conhecimentos técnicos prévios (ROBINS, 2010).

É necessário, então, extrair ou coletar os fatores externos que caracterizem os alunos de forma holística, em seguida materializá-los em métricas, para que finalmente possam ser relacionados, esses fatores, ao desempenho acadêmico dos alunos.

2.4 Integração de dados

É consenso entre professores e pedagogos que os exercícios resolvidos por alunos é fonte de informação para se identificarem possíveis dificuldades de aprendizagem (ANDRADE, LAROS, 2007; ROBINS, 2010).

Repositórios de códigos fontes gerados por simulações de OI, por meio do serviço *ForCode* por exemplo, representam, por definição, resultados de exercícios desafiadores. Estes tanto podem ser analisados de forma isolada como integrados a dados externos, holísticos, em prol da melhoria do processo de ensino e aprendizagem.

Reunir dados dos alunos sob diferentes perspectivas, mesmo dentro de uma única instituição, é uma tarefa árdua e desafiadora. Entre as principais causas está a inexistência de um sistema de informação único que conglomere todos os fluxos e bases de informações de uma organização educacional. O que se encontra na realidade é uma diversidade de sistemas, desde os implementados ainda em fichários manuais, passando por planilhas, até sistemas de informação de propósito específico que automatiza um ou poucos departamentos da organização educacional.

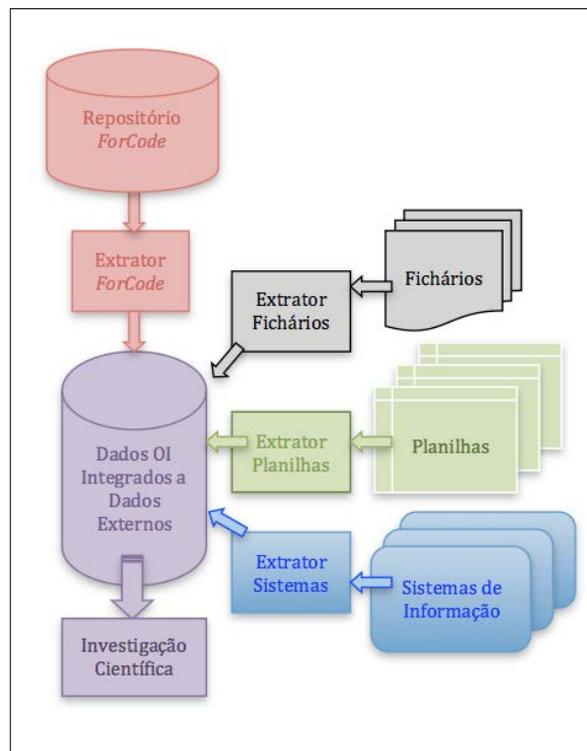
Assim, o custo de tempo e de esforço tomado do pesquisador para obtenção de dados de diversas e diferentes fontes consome boa parte dos recursos de projetos. Essa dificuldade rotineira desestimula a realização de pesquisas que considerem uma ampla quantidade de fatores a serem investigados.

Uma das formas de mitigar esse custo é fazer uso da automatização. A seguir, na Figura 1, está diagramada a arquitetura do sistema de software utilizado neste trabalho como uma solução para integrar dados de OI a dados externos.

Além dos dados de desempenho obtidos em OI, por exemplo, do **Repositório ForCode**, devem se fazer presentes dados que caracterizem os estudantes holisticamente, de diferentes perspectivas. Esses dados podem ser extraídos de diferentes departamentos da organização educacional na forma de fichários, planilhas e ou sistemas de informação. Os componentes extratores, **Extrator Fichários**, **Extrator Planilhas**, e **Extrator Sistemas** podem ser implementados de forma incremental em função da necessidade e da natureza do dado. Com os **Dados de OI Integrados a Dados Externos**, uma infinidade de avaliações e análises podem ser realizadas por diferentes métodos de **Investigação Científica**. Entre os objetivos das avaliações destaca-se a identificação

de fatores impactantes no desempenho discente em simulados de OI.

Figura 1 – Arquitetura da Solução



Fonte: Autoria própria.

3 Questões de pesquisa

Temos conhecimento de que este nosso trabalho é pioneiro na iniciativa de investigação sistematizada sobre os repositórios gerados pelo simulador de OI *ForCode*. Por isso, julgamos fundamental averiguar se os dados contidos nesses repositórios são de fato fonte de informação para identificar possíveis dificuldades de aprendizado, assim como são os exercícios regulares resolvidos por alunos (ROBINS, 2010). Assim, definimos nossa primeira Questão de Pesquisa (QP1):

QP1 – É possível identificar, no repositório de soluções de código fonte dos simulados de informática do IFPB, estratégias de resolução (quantidade de submissões, linguagem de programação) utilizadas pelos alunos-competidores que influenciam no seu desempenho?

A resposta parece ser obviamente afirmativa, no entanto apenas com análise sobre os dados, com foco para responder a esta questão, é que teremos evidências que comprovem se a extração de informações relevantes contribuem na identificação de

estratégias, benéficas ou não, no desempenho dos estudantes nesses simulados.

Além das estratégias utilizadas pelos alunos durante os simulados de OI, fatores externos (formação educacional e nível de escolaridade, conhecimentos técnico-artísticos anteriores, situação socioeconômica, estado de saúde clínico-psicológico, assiduidade de prática esportiva, entre outros) que caracterizam os alunos de forma holística podem também influenciar nos seus respectivos desempenhos. Identificar quais desses fatores são os mais impactantes, positiva ou negativamente, no desempenho nos possibilita indicar o que os alunos devem priorizar ou melhorar. Diante desse contexto, definimos uma segunda Questão de Pesquisa (QP2):

QP2 – É possível correlacionar o desempenho dos alunos em simulados de OI com os fatores holísticos aos quais esses alunos se encontram imersos na suas rotinas diárias?

Explorando o repositório de códigos fontes do serviço *ForCode*, é possível obter conclusões capazes de responder à QP1. Já para se responder à QP2, se faz necessário ir além dos dados coletados do repositório de código do *ForCode*. É preciso associar dados oriundos de fontes que caracterizam alunos de diferentes perspectivas. Por exemplo, é bem provável que o desempenho em simulados de OI esteja relacionado com a maturidade do aluno nesse tipo de competição ou até com o nível de conhecimento técnico que possua.

4 Avaliação do desempenho dos alunos

Para demonstrar o potencial informacional contido no repositório de OI simuladas pelo *ForCode*, selecionamos uma para ser avaliada sobre a égide das QP de interesse.

A execução da avaliação, metodologicamente, se deu por meio das etapas de Seleção da amostra, Formulação de hipóteses, Coleta de dados, Critérios de descarte, Análise dos dados e resultados e Ameaças à validade. Cada uma dessas etapas detalharemos nas subseções a seguir.

4.1 Seleção amostra

Entre as 3(três) OI realizadas no Campus Campina Grande, a realizada nos dias 27 e 28 de novembro de 2015 foi a que obteve a maior adesão por parte dos alunos. Nela, cadastraram-se 47 alunos de cur-

sos técnicos integrados ao Ensino Médio e cursos de Informática, e de Manutenção e Suporte à Informática subsequentes ao Ensino Médio.

As duas outras, de 2014 e 2016, obtiveram respectivamente 7 e 11 participantes apenas, fato que as inviabilizava para fins de análise estatística. Outro motivo que justifica a seleção única da OI de 2015 foi de que, nessa edição, os alunos participantes eram de diferentes anos de escolaridade. Nas de 2014 e 2016, existiu uma forte concentração de alunos apenas do último ano de cursos integrados ao Ensino Médio.

Na OI de 2015, para cada um dos 10 (dez) problemas-desafios, o aluno poderia submeter diversas tentativas e fazer uso da linguagem de programação que julgasse ser a mais conveniente para resolver qualquer um dos problemas, limitadas, contudo, às seguintes: C, C++, Python, Java ou Pascal.

4.2 Formulação de hipóteses

Derivadas da QP1, definimos duas hipóteses. A primeira delas averigua se o número de tentativas submetidas por aluno influencia no seu desempenho. Segue sua definição:

HI-1: Quanto maior o número de tentativas submetidas por um aluno maior sua quantidade de acertos.

HI-0 (nula): Embora haja maior o número de tentativas submetidas por um aluno não há aumento na sua quantidade de acertos.

Já a segunda hipótese avalia se a linguagem de programação utilizada influencia no desempenho do aluno. Segue:

HII-2: Há predominância de uso de um tipo de linguagem de programação pelos alunos com melhor desempenho.

HII-0 (nula): Não há predominância de uso de nenhum dos tipos de linguagem de programação pelos alunos com melhor desempenho.

As hipóteses **HI** e **HII**, claramente, buscam evidências sobre o potencial dos simulados de OI como fonte de informação para identificar possíveis dificuldades de aprendizado (ROBINS, 2010).

Considerando-se, contudo, uma visão holística sobre o aluno, fatores externos são também possíveis causas da deficiência no aprendizado (ANDRADE; LAROS, 2007). O nível de conhecimento indicado pelo ano de escolaridade do aluno é um dado externo ao simulado de OI, logo pode ser um fator de influência direta no desempenho dos alunos. Aplicando o

ano de escolaridade sobre a QP2, derivamos a nossa terceira hipótese:

HIII-1: Quanto maior o ano de escolaridade do aluno melhor seu desempenho.

HIII-0 (nula): Embora seja maior o ano de escolaridade do aluno, não há melhoria no seu desempenho.

4.3 Coleta de dados

A avaliação das hipóteses definidas depende da extração e coleta tanto dos dados de desempenho do aluno como também do seu respectivo ano de escolaridade. Para tanto, seguindo a Arquitetura da Solução da Figura 1, foram desenvolvidos dois componentes de software:

Extrator ForCode – constituído de um script em linguagem SQL-*Structure Query Language* para o banco de dados MySQL do Repositório *ForCode*. A execução desse script gera uma tabela em que cada linha representa um aluno participante, e as colunas representam o desempenho por ele obtido;

Extrator Sistema Q-Acadêmico – consiste em obter a escolaridade, ou ano escolar, que cada aluno cursava na época em que participou do simulado. A extração dessa informação foi realizada de forma manual, consultando-se o histórico escolar registrado no sistema de gestão acadêmica do IFPB, o Q-Acadêmico.

Com os resultados gerados pelos dois componentes de software, geramos a **Base de Dados OI Integrada a Dados Externos**. Para fins de pesquisa deste trabalho, essa base é composta do histórico de submissões de código associado ao ano de escolaridade de cada aluno.

4.4 Critérios de descarte

Inicialmente, foram coletados o desempenho de todos os 47 alunos que se cadastraram para participar da simulação de OI de 2015, no Campus citado. Desses, 34 (72%) realizaram alguma submissão, isto é, tentaram responder a algum dos 10 problemas-desafio. Os 13 (27%) restantes não conseguiram ou não quiseram submeter resposta alguma a quaisquer dos desafios do simulado.

Esses 13 alunos foram descartados da nossa amostra de análise por entendermos que a desistência de um aluno em participar não faz parte do escopo das nossas QP. Em trabalhos futuros, é possível que se tente pesquisar sobre os fatores que inibem alu-

nos de sequer tentarem responder a quaisquer dos desafios.

4.5 Análise dos dados e resultados

Para os 10 desafios, os 34 alunos participantes efetivos utilizaram 317 códigos fontes como respostas às submissões. Estratificando as submissões por aluno, por linguagem de programação, e pelo veredito sobre cada submissão realizada, obtivemos a Tabela 1. A coluna **Id** identifica unicamente cada aluno, a coluna **Ling.** indica a linguagem de programação escolhida pelo aluno para submeter suas respectivas soluções. Entre os vereditos possíveis para cada submissão, temos as que obtiveram respostas **Corretas**, as com repostas **Erradas**, as que continham **Erros de Compilação** e o **Total** de submissões, respectivamente identificadas, na Tabela 1, da terceira à quinta coluna.

Apenas 1 aluno submeteu o maior número de submissões: 19 vezes; 2 alunos realizaram submissões únicas. A média de submissões por aluno foi de 9.32.

Sobre a Tabela 1, geramos o histograma, Gráfico 1, com frequência de alunos que realizaram o mesmo número de submissões. Nesse histograma observa-se, inclusive visualmente, que temos dois grupos bem definidos em termos de números de submissões: 1) 17 alunos (50%) cujas submissões foram acima da média e em que prevalece a mediana de 11 submissões por aluno; e, 2) os 17 restantes cujas submissões foram abaixo da média e em que prevalece a mediana de 4 submissões por aluno.

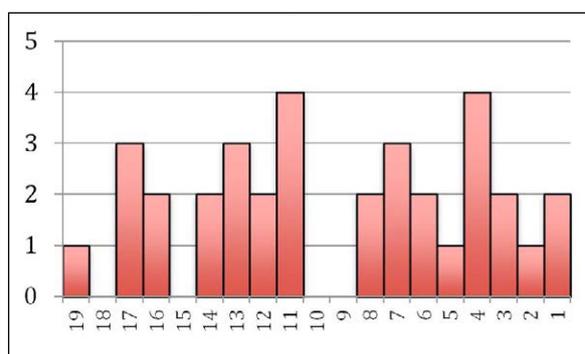
Entre os que obtiveram as quatro maiores quantidades de respostas corretas – alunos com Ids 47, 2, 26, 4, 28 e 33 –, todos estão no grupo cujo total de submissões está acima da média de 9.32. Já entre aqueles que obtiveram no máximo uma resposta correta – alunos com Ids 6, 7, 14, 25, 35, 41, 51, 3, 8, 13, 23, 42, 44, 45, e 48 –, apenas 3 não estão no grupo cujo total de submissões está acima da média de 9.32. Essa constatação é uma primeira evidência de que a quantidade de submissões tem influência positiva no desempenho dos alunos em OI.

Tabela 1 – Veredito de submissões por aluno

Id	Ling.	Corretas	Erradas	Erros de Compilação	Total
47	Cpp	10	2	0	12
2	Python	6	11	0	17
26	Java	6	6	2	14
4	Python	4	9	0	13
28	Java	4	12	0	16
33	Cpp	4	9	0	13
10	Pascal	3	1	0	4
12	Pascal	3	6	2	11
16	Pascal	3	3	0	6
19	C	3	11	2	16
20	Java	3	4	0	7
27	Cpp	3	10	0	13
43	Python	3	4	0	7
11	Pascal	2	10	0	12
15	Pascal	2	3	0	5
24	C	2	9	0	11
30	Java	2	9	0	11
34	C	2	15	0	17
37	Cpp	2	12	0	14
6	Pascal	1	7	0	8
7	C	1	3	0	4
14	Pascal	1	7	3	11
25	Java	1	0	0	1
35	Cpp	1	18	0	19
41	Python	1	6	0	7
51	Cpp	1	0	1	2
3	C	0	4	0	4
8	Pascal	0	6	0	6
13	Pascal	0	8	0	8
23	Java	0	17	0	17
42	Python	0	4	0	4
44	Python	0	3	0	3
45	Java	0	1	0	1
48	C	0	2	1	3

Fonte: Autoria própria

Gráfico 1– Frequência de alunos por número de submissões



Fonte: Autoria própria

Para averiguar essa influência considerando todo o universo de alunos, computamos a matriz de correlação de *Pearson*, Tabela 2, para: submissões com respostas **Corretas**, respostas **Erradas**, **Erros de Compilação** e **Total de Submissões**.

Tabela 2 – Matriz de correlação entre vereditos

Matriz de Correlação	Corretas	Erradas	Erros de Compilação	Total de Submissões
Respostas Corretas	1,00	0,03	0,07	0,45
Respostas Erradas	0,03	1,00	-0,05	0,90
Erros de Compilação	0,07	-0,05	1,00	0,13
Total Submissões	0,45	0,90	0,13	1,00

Fonte: Autoria própria

De acordo com os coeficientes de *Pearson* computados, não há correlação entre a quantidade de repostas **Corretas** e o número de repostas **Erradas**, nem entre respostas **Corretas** e aquelas com **Erros de Compilação**, mas observa-se uma forte correlação positiva, **corr = 0,90**, entre o **Total de Submissões** e respostas **Erradas**, e uma correlação mediana, **corr = 0,45**, entre o **Total de Submissões** e respostas **Corretas**.

Tal fato, na nossa interpretação, evidencia que quanto maior o número de tentativas para responder a questões maior é o número de respostas erradas. Apesar dessa constatação, entende-se que, com os erros, o aluno tende a aprender, logo é possível que não erre novamente. E, desde que tenha perseverança, o aluno tende a finalmente acertar as respostas.

Assim, baseado nas duas evidências, podemos aceitar a hipótese HI-1 de que, quanto maior a quantidade de tentativas submetidas por um aluno maior sua quantidade de acertos. Consequentemente, reafirmamos a hipótese HI-0.

Do ponto de vista de linguagem de programação por número de submissões, a Tabela 1 revela que: 79 (24,9%) das submissões foram feitas com códigos escritos em Cpp (C++); 71 (22,4%) em Pascal; 66 (20,8%) em Java; 51 (16,1%) em C; e 50 (15,8%) codificadas em Python.

Quando tipificamos linguagem de programação por aluno, contudo, a ordem de preferência da linguagem muda. Pascal foi a mais utilizada, já que 9 alunos (26,5%) a preferiram; a segunda foi Java, escolhida por 7 alunos (20,6%); e, por fim e empatadas, vem C, Cpp (C++) e Python, com cada uma sendo preferida por 6 alunos (17,6%).

Apesar de o aluno com a nota máxima de acertos, 10 (dez), fazer uso da linguagem Cpp (C++), não podemos afirmar que o uso dessa linguagem é determinante para o sucesso no simulado. Isso fica claro quando observamos a linguagem utilizada pelos top-3 – melhores alunos; cada um fez uso de uma linguagem diferente. Da mesma forma não observamos predominância de nenhuma das linguagens de programação quando consideramos os alunos top-5 ou top-10.

Diante dessas evidências, a hipótese HII-1 é refutada. Tal fato indica que não há uma linguagem de programação predominantemente utilizada pelos alunos que tenha promovido uma melhoria no seu desempenho durante o simulado de OI realizado IFPB, Campus Campina Grande.

Quanto à adição dos dados de desempenho ao ano de escolaridade (do ensino técnico) em que se encontrava o aluno na época da sua participação no simulado, geramos a Tabela 3, apresentada a seguir.

Nela, a coluna **Id** identifica o aluno, e, nas demais colunas, temos a quantidade de **Respostas Corretas** conseguidas pelo aluno do simulado, e sua respectiva **Escolaridade**.

Agrupamos os alunos em três níveis de escolaridade: grupo 1, formado pelos alunos que estavam cursando primeiro ano do ensino técnico integrado ao EM, na época do simulado; grupo 2, composto por aqueles que estavam no segundo ano; e o grupo 3 composto pelos que estavam cursando o quarto ano dessa modalidade.

Tabela 3 – Respostas corretas por ano de escolaridade

Id.	Respostas Corretas	Escolaridade
47	10	4
2	6	1
26	6	4
4	4	1
28	4	4
33	4	2
10	3	1
12	3	1
16	3	1
19	3	1
20	3	4
27	3	2
43	3	1
11	2	1
15	2	1
24	2	2
30	2	4
34	2	2
37	2	2
6	1	1
7	1	1
14	1	1
25	1	2
35	1	2
41	1	1
51	1	2
3	0	4
8	0	1
13	0	1
23	0	1
42	0	1
44	0	1
45	0	4
48	0	1

Fonte: Autoria própria

Não houve alunos do terceiro ano dos cursos citados acima porque os alunos do grupo 3, composto pelos alunos do quarto ano, foram os últimos que cursaram a matriz curricular do integrado ao EM distribuída em quatro anos.

Computadas as médias de repostas corretas por aluno para cada um dos grupos, temos os seguintes valores: Para o grupo 1 a média foi 1.74; a média do grupo 2 aumentou 15% em relação à média obtida pelo grupo 1, passando para 2.0 questões corretas por aluno. Já no grupo 3, a média de repostas corretas foi de 3.57, que, em relação à média do grupo 2, aumentou 78%, e, em relação à média do grupo 1 aumentou 105%.

Diante do incremento absoluto e percentual da média de repostas corretas por aluno, considerando o agrupamento por escolaridade, aceitamos a hipótese HIII-1. Assim, à medida que o aluno acumula anos de escolaridade técnica, melhor é seu desempenho em OI.

4.6 Ameaças à validade

Os resultados aqui obtidos são restritos ao universo de alunos participantes do simulado de OI realizado em novembro de 2015 no IFPB, Campus Campina Grande. Além disso, como a amostra analisada é restrita a apenas uma edição de OI, as conclusões não podem ser generalizadas. Fatores como: nível de escolaridade (Fundamental, Técnico Integrado ao Ensino Médio, Técnico Subsequente ao Ensino Médio, Tecnológico, Bacharelado) dos alunos competidores, nível de dificuldade das questões, e até o entusiasmo dos participantes, podem influenciar nos resultados das análises.

5 Conclusões

O uso de sistemas de software que integram o desempenho de alunos em OI a dados holísticos coletados a partir de sistemas externos possibilita reduzir o custo de pesquisas focadas em identificar fatores que impactam no desempenho de OI. Nesse modelo, o pesquisador não desperdiça nem esforço nem tempo para criar instrumentos (e.g.: questionários) a serem submetidos a uma amostra de alunos. Pelo contrário, ao pesquisador o sistema já disponibiliza uma massa de dados sobre a qual ele testará suas hipóteses.

Como comprovação da valorosa fonte de informação contida nos repositórios de simuladores de OI,

executamos uma avaliação sobre o desempenho de 34 alunos do IFPB, dos Cursos Integrados ao Ensino Médio e Cursos Técnicos em Informática e Manutenção e Suporte a Informática Subsequentes ao Ensino Médio, na OI simulada pelo *ForCode*, realizada em novembro de 2015. Adicionamos, ainda, a essas medidas de desempenho o ano de escolaridade, coletado a partir do sistema de gestão acadêmica do IFPB. Por fim, avaliamos o impacto da escolaridade no desempenho alcançado pelos alunos, individualmente, na referida OI.

As evidências apontam que o bom desempenho dos alunos independe da linguagem de programação por eles utilizadas, mas o uso da estratégia “tentativa e erro” é um componente que propicia um maior número de acertos. Em outras palavras, alunos persistentes, que não desistem até conseguirem escrever o código correto para um dado desafio, analisam o que pode estar errado e realizam novas tentativas têm, de modo geral, melhor desempenho.

Do ponto de vista dos professores-treinadores, é importante entender que o desempenho dos alunos também aumenta com o amadurecimento dos seus conhecimentos em programação, fato que ocorre à medida que o aluno avança no seu ano de escolaridade. E, diante dessas evidências, fazer com que alunos participem de simulados de OI, de fato os estimula a aprimorarem seus conhecimentos.

6 Trabalhos futuros

Como trabalhos futuros, pretendemos construir novos extratores por meio dos quais seja possível coletar, de alunos participantes de simulados de OI, outros dados, de diferentes perspectivas. Tendo novos dados de alunos, poderemos investigar cientificamente outros fatores que causam impacto no seu desempenho em OI.

Apontar os alunos competidores que estão melhor preparados a partir da performance obtida em simulações não contribui, por exemplo, para uma melhor formação em programação dos demais alunos competidores que não conseguem evoluir na resolução dos desafios dos simulados de que participam.

Também, ainda em 2017, vamos revitalizar o projeto POP-CG e realizar novas edições de simulações de OI, com números representativos de alunos. Aumentando a nossa base de desempenho de alunos, poderemos refazer nossas análises estatisticamente significativas.

Acreditamos que, com o incremento da base de dados e com a identificação de novos fatores impactantes no desempenho de alunos, a Instituição de Ensino possa melhorar suas práticas pedagógicas. Alcançaríamos assim o objetivo maior de Olimpíadas Científicas: mais do que servirem de suporte para competições que revelam jovens talentos com conhecimentos científicos de alto nível, as OE podem e devem ser utilizadas como ferramenta educacional para auxiliar aqueles jovens com dificuldades de evoluir no seu aprendizado (CGEE, 2011; Olimpíada Brasileira de Informática, n.d.).

7 Reprodutibilidade

Para fins de reprodução dos resultados aqui alcançados bem como para possíveis contribuições para evolução, este estudo encontra-se disponível em: <http://www.softwarevocabulary.org/home/avaliacao-de-fatores-impactantes-no-desempenho-de-alunos-em-olimpiadas-de-programacao>. Estão disponíveis o repositório de código fonte gerado pelo simulador *ForCode* quando da realização simulado de OI em novembro de 2015, além de scripts e planilhas utilizadas para processá-lo e analisá-lo.

REFERÊNCIAS

Andrade, J. M.; Laros, J. A. Fatores associados ao desempenho escolar: estudo multinível com dados do SAEB/2001; **Psicologia: Teoria E Pesquisa**, v. 23 n. 1, p. 33-41, Brasília, jan./mar. 2007. Disponível em: <<http://doi.org/10.1590/S0102-37722007000100005>>.

CENTRO DE GESTÃO E ESTUDOS ESTRATÉGICOS (CGEE). **Avaliação do impacto da Olimpíada Brasileira de Matemática nas Escolas Públicas**: Série Documentos Técnicos. Brasília, 2011. n. 11. Disponível em: <<https://www.cgee.org.br/>>.

Dehnadi, S.; Bornat, R. (2006). The camel has two humps (working title). Middlesex University, UK, 1-21. Disponível em: <<http://mrss.dokoda.jp/r/http://www.eis.mdx.ac.uk/research/PhDArea/saeed/paper1.pdf>>.

McCabe, T. A complexity measure. **IEEE Transactions on Software Engineering**, v. 2 n. 4, p. 308–320, dec. 1976.

Morais, L. A. de M. (2015). **Avaliação de uma Abordagem Personalizada de Ensino no Contexto de Programação Introdutória**.

Universidade Federal de Campina Grande, Campus I, Campina Grande, Brasil, 2015.

Olimpíada Brasileira de Informática. (n.d.). Retrieved February 17, 2017, Disponível em: <<http://olimpiada.ic.unicamp.br/>>.

Olimpíada Paraibana de Informática. (n.d.). Retrieved February 16, 2017, Disponível em: <<http://www.dsc.ufcg.edu.br/~opi/>>.

Parnas, D. L. Software aging. *Proceedings of 16th International Conference on Software Engineering*, p. 279–287, 1994. Disponível em: <<http://doi.org/10.1109/ICSE.1994.296790>>.

Pinto, J. de C. A.; Tenório, R. M.. **A Influência dos Fatores Socioeconômicos no Desempenho Acadêmico dos Estudantes de Ensino Médio Integrado do IFBA-Campus Barreiras**, 2014. Disponível em: <http://www.equidade.faced.ufba.br/sites/equidade.oe.faced.ufba.br/files/jucinara_pinto_robinson_tenorio_-_a_influencia_dos_fatores_socioeconomicos_no_desempenho.pdf>.

Robins, A. Learning edge momentum: a new account of outcomes in CS1. **Computer Science Education**, v. 20, n.1, p. 37–71, jan. 2010. Disponível em: <<http://doi.org/10.1080/08993401003612167>>.

Santos, K. de F.. Investigaç o sobre Uso de Vocabul rio de C digo Fonte para Identifica o de Especialistas Katyusco de Farias Santos. Universidade Federal de Campina Grande, 2015.

Santos, K. de F.; Guerrero, D. D. S.; Figueiredo, J. C. A. de. Using Developers Contributions on Software Vocabularies to Identify Experts. In: 12TH INTERNATIONAL CONFERENCE ON INFORMATION TECHNOLOGY - NEW GENERATIONS (pp. 452–456). Las Vegas, NV, USA: IEEE Comput. Soc., 2015. Disponível em: <<http://doi.org/10.1109/ITNG.2015.149>>.

Tavares, B. F.; B ria, J. U.; Lima, M. S. de. Fatores associados ao uso de drogas entre adolescentes escolares. **Revista de Sa de P blica**, S o Paulo, v. 38, n. 6, p.787–796, dez. 2004. Disponível em: <<http://doi.org/10.1590/S0034-89102004000600006>>.

Venkataramani, R. et al. Discovery of Technical Expertise from Open Source Code Repositories. *Proceedings of the 22nd International Conference on World Wide Web Companion*, p. 97–98, 2013. Disponível em: <<http://doi.org/10.1145/2487788.2487832>>.

Xie, G.; Chen, J.; Neamtiu, I. Towards a better understanding of software evolution: An empirical study on open source software. *2009 IEEE International Conference on Software Maintenance*, p. 51–60, 2009. Disponível em: <<http://doi.org/10.1109/ICSM.2009.5306356>>.