

Caracterização de Grupos de Estudantes de Programação: Uma Abordagem Baseada em Modelagem de Redes

João Pedro M. Sena¹, Carlos Eduardo P. Silva^{1,2}, André G. Santos¹, Julio C. S. Reis¹

¹Departamento de Informática - Universidade Federal de Viçosa (UFV)

²Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais (IFMG)

{joao.sena, carlos.e.silva, andresantos, jreis}@ufv.br

Abstract. *Automatic exercise grading systems for programming exercises or Online Judges (OJ) are widely used in the educational context. In addition to allowing teachers to apply exercises on a large scale, they can also capture relevant data about student interaction during this process. In this work, we explore complex network techniques to propose a methodology for clustering and characterizing groups of programming students from OJ data. Overall, the results reveal interesting patterns within the different groups of students, highlighting the potential of the proposed approach to be used as a tool to support teachers in programming classes.*

Resumo. *Sistemas de correção automática de exercícios de programação ou Juízes Online (JO) são amplamente utilizados no contexto educacional. Além de permitirem ao professor a aplicação de exercícios em larga escala, também podem capturar dados relevantes acerca da interação dos alunos durante este processo. Neste trabalho, exploramos técnicas de redes complexas para propor uma metodologia para agrupamento e caracterização de grupos de estudantes de programação a partir de dados de um JO. De forma geral, os resultados revelam padrões interessantes dentro dos diferentes grupos de estudantes, evidenciando o potencial da abordagem proposta para ser utilizada como ferramenta de suporte aos professores em turmas de programação.*

1. Introdução

Juízes Online (JO) são sistemas criados para viabilizar a verificação automática da correção de algoritmos [Barbosa et al. 2023]. Na educação, esses sistemas têm sido usados há pelo menos 60 anos, permitindo a correção automática de exercícios realizados por alunos de programação em diferentes linguagens [Wasik et al. 2018]. Nesse tipo de sistema, um servidor previamente configurado avalia o código dos usuários por meio de diversos casos de teste e retorna uma resposta, que pode estar relacionada à correção da implementação ou ainda incluir o tipo ou o nível do erro, se for o caso. Cada código enviado é considerado como uma tentativa de resolução de um exercício. Exemplos desses sistemas incluem o *BOCA Online Contest Administrator*, o *CodeBench*¹, o *beecrowd*², entre outros.

A interação dos estudantes com JOs produz dados que podem ser valiosos neste contexto educacional. Além de funcionar como um corretor automático, esse tipo de

¹<https://codebench.icomp.ufam.edu.br/>

²<https://judge.beecrowd.com/>

sistema também armazena informações como o número de tentativas e seus respectivos tempos em um banco de dados desacoplado. Esses dados podem ser explorados para diferentes análises, como previsão de reprovação ou das preferências de aprendizagem dos alunos [Silva et al. 2023, Mohamad and Tasir 2013]. Embora existam alguns esforços na literatura com o objetivo de propor abordagens para suporte ao docente neste contexto [Lopes 2003, Garin et al. 2006, de Oliveira et al. 2015], ainda há uma carência em formas de representar e extrair informações desse tipo de dado diretamente, caracterizando os estudantes por meio de suas interações e não a partir de previsões, por exemplo.

Neste trabalho, exploramos dados de um JO coletados durante aulas práticas de uma disciplina de programação ministrada em uma universidade federal de ensino superior do Brasil ao longo de cinco anos. Neste contexto, é importante destacar alguns desafios: 1) o volume de interações geradas pelos estudantes nestes sistemas é alto, o que, na maioria das vezes, inviabiliza qualquer tipo de análise manual por parte do docente; 2) existe uma grande heterogeneidade do conhecimento dos alunos sobre programação [Capovilla et al. 2015], o que pode exigir do professor um alto nível de personalização durante o ensino. Por isso, dada a variedade de perfis dos estudantes, é importante que o professor tenha algum indicativo do desempenho dos alunos e de suas dificuldades de forma estratificada. Assim, medidas podem ser tomadas de maneira específica, permitindo uma análise mais efetiva do aprendizado desses estudantes, algo imprescindível para promover algum tipo de reforço educacional especializado [Silva et al. 2023], o que pode contribuir, potencialmente, para a diminuição da evasão estudantil [Ícaro Alvim et al. 2024].

Diante deste contexto, propusemos uma abordagem onde estudantes são modelados em uma rede (i.e., grafo), criada a partir das suas tentativas de solução para exercícios de programação submetidas em um JO. Nessa rede, os estudantes foram agrupados a partir dos exercícios resolvidos em comum e caracterizados a partir de diferentes métricas. Em resumo, a abordagem proposta teve como objetivo modelar os alunos e suas relações, e então extrair informações relevantes. Com isso, considerando principalmente as aplicações práticas, espera-se que a modelagem proposta possa ser usada por professores para avaliação de seus alunos diretamente dos dados, diferentemente de outros trabalhos, que focam, por exemplo, na previsão de evasão ou de desempenho.

O restante do trabalho está organizado da seguinte forma: na Seção 2 são destacadas as principais diferenças em relação a outras abordagens recentes ou semelhantes. O formato dos dados, modelagem, métricas e detalhes relativos à etapa de caracterização, são apresentados nas Seções 3 e 4. Por fim, a Seção 5 apresenta as conclusões e direções para trabalhos futuros neste contexto.

2. Trabalhos Relacionados

É crescente o número de esforços que fazem uso de técnicas de computação para criação de mecanismos de suporte no contexto educacional [Lopes 2003, Garin et al. 2006, de Oliveira et al. 2015], em especial a partir de dados oriundos de JO [Fernandes et al. 2023]. Em [Silva et al. 2023], por exemplo, são exploradas técnicas de mineração de dados para extração de informações pertinentes de estudantes de um JO. Nesse estudo, foram usados dados da interação de alunos de uma turma de programação com um JO para prever as reprovações desses estudantes. Em resumo, os resultados evidenciam que dados gerados naturalmente pelo uso do sistema podem ser aplicáveis para

auxiliar o professor na compreensão de seus alunos.

Além disso, a aplicação de técnicas de redes complexas na educação tem proporcionado novas formas de análise dos impactos das relações entre os estudantes ou entre as atividades realizadas por eles. Neste cenário, o estudo apresentado em [Weber et al. 2020] avalia a evolução da relação entre alunos em uma universidade a partir de fatores como gênero, ano de ingresso e educação dos pais. Nesse trabalho, foi considerada a influência da semelhança dos indivíduos no surgimento de laços, conhecida como homofilia. Em suma, os resultados indicam que estudantes semelhantes tendem a ter um desempenho similar.

Já em [Aguiar et al. 2023], foi considerada a avaliação do desempenho de alunos a partir da modelagem de uma rede de amizades entre eles, agrupando os alunos em comunidades e verificando as relações com as notas. Nesse estudo, os autores aplicaram um questionário para estudantes de uma turma de Ensino Médio para identificar até cinco amigos mais próximos. A partir da rede modelada, os autores dividiram os estudantes em comunidades e verificaram a relação entre os grupos formados e as notas dos alunos nos grupos. Os resultados mostram uma correlação importante entre as amizades dos estudantes e seu desempenho acadêmico.

Diferente dos trabalhos anteriores, o suporte ao professor neste trabalho é realizado por meio de informações que são extraídas diretamente dos dados de histórico de tentativas dos alunos, e não por meio de sistemas de recomendação, previsão ou decisão a partir desses dados. Além disso, apesar de alguns trabalhos focarem também na análise de relações dos estudantes a partir do uso de técnicas de redes complexas, o estudo aqui apresentado não se concentra na esfera de relações sociais, mas sim no desempenho acadêmico avaliado por meio das atividades realizadas pelos estudantes, que pode ser extremamente útil para suportar decisões dos docentes neste contexto.

3. Metodologia

3.1. Base de Dados

A base de dados explorada neste estudo é composta por tentativas de alunos na resolução de exercícios em um JO. Esses dados foram coletados durante diferentes ofertas de uma disciplina introdutória de programação do curso de Ciência da Computação ministrado na Universidade Federal de Viçosa (UFV), no Brasil. Durante a referida disciplina, o professor utiliza o sistema (i.e., JO) como mecanismo de suporte para controle das atividades práticas propostas durante o período das aulas a partir de exercícios que podem variar entre os anos. Por outro lado, o aluno usa esse mesmo sistema para envio de suas tentativas para cada exercício. Os dados foram utilizados de forma anônima, respeitando os princípios éticos de acordo com projeto de número CAAE 75327023.0.0000.5153.

Cada tentativa realizada no sistema é composta pela data e horário de envio (`timestamp`), por um número que identifica o estudante (`student_id`), por uma letra que identifica o exercício (`exercise_id`), pelo resultado da tentativa (`is_correct`), que pode assumir o valor `True` para uma tentativa bem-sucedida (correta) e `False`, caso contrário (incorreta). Nesse caso, o resultado é expresso com valores binários, desconsiderando, por exemplo, a porcentagem de casos de testes corretos ou os tipos de erros. Além disso, foi acrescentado em cada tentativa o tempo gasto em minutos

Tabela 1. Exemplo das informações contidas na base de dados.

timestamp	student_id	exercise_id	is_correct	spent_time
10	1	A	False	10
14	2	A	True	14
20	3	C	True	20
25	1	C	False	15
35	1	A	True	10
40	3	D	True	20
49	2	C	True	35
50	1	B	True	15
59	2	B	True	10

(`spent_time`). Esse tempo foi calculado considerando a diferença entre o horário de envio de cada tentativa em relação ao horário da tentativa anterior enviada pelo mesmo estudante³ e, assim, obtendo uma aproximação do tempo dedicado em cada tentativa. Um exemplo dos dados é apresentado na Tabela 1, com os estudantes 1, 2 e 3, e os exercícios A, B, C e D. Por fim, considera-se a soma dos tempos gastos em cada uma das tentativas realizadas por um estudante em um exercício como o tempo de resolução. Por exemplo, o tempo de resolução do exercício A pelo estudante 1 foi $10 + 10 = 20$.

No total, foram coletadas 11.226 tentativas realizadas ao longo dos seguintes anos: 2018, 2019, 2021, 2022 e 2023. Em 2020, devido à pandemia da COVID-19, as aulas presenciais foram suspensas, e por isso o uso do JO foi comprometido na dinâmica da disciplina nesse ano. Como a disciplina é ofertada apenas uma vez por ano, cada ano corresponde a um período acadêmico. A disciplina possui cerca de 12 aulas práticas (uma por semana), cada uma delas com um conjunto de exercícios e com duração de 110 minutos. A Tabela 2 sumariza as informações, com a última coluna apresentando a quantidade de exercícios resolvidos em média pelos estudantes, além do quanto esse valor representa em relação ao total de exercícios. Pode-se observar um valor de $\approx 50\%$, indicando que, em geral, os estudantes resolvem uma parte significativa dos exercícios da disciplina.

Nota-se que os números de alunos e exercícios apresentam valores semelhantes ao longo dos anos, exceto em 2021, com uma queda significativa no número de estudantes. Tanto o número de tentativas como o número de estudantes em 2021 podem ter sido influenciados pelas condições de ensino remoto provocadas pela pandemia da COVID-19, impactando, conseqüentemente, na queda desses valores. Contudo, vale observar que a média de tentativas por estudante se mantém regular.

Tabela 2. Estatísticas gerais da base de dados.

Ano	# Estudantes	# Exercícios	# Tentativas	(# Tent. / # Est.)	(# Ex. Resolvidos / # Est.)
2018	57	49	2.129	37,35	21,09 (43%)
2019	59	54	2.486	42,14	25,68 (48%)
2021	46	51	1.873	40,72	23,35 (46%)
2022	52	48	2.137	41,10	26,92 (56%)
2023	60	53	2.601	43,35	27,25 (51%)

³Para a primeira tentativa, foi considerado o horário de *login* do aluno depois do início da aula.

3.2. Modelagem das Redes

A modelagem das redes exploradas neste estudo foi inspirada em esforços anteriores [Araujo et al. 2023], que também fazem uso de técnicas de redes complexas para capturar relações relevantes e não triviais entre elementos distintos. Além disso, a modelagem também foi dividida em algumas etapas, as quais são detalhadas a seguir.

Inicialmente, pode-se modelar uma rede considerando as informações sem nenhuma modificação. Com isso, obtém-se um grafo de arestas múltiplas (multigrafo) $\mathcal{A} = (\mathcal{V}_A, \mathcal{E}_A)$, sendo \mathcal{V}_A o conjunto de nós que simbolizam estudantes e exercícios, e \mathcal{E}_A o conjunto de arestas, cada uma ligando dois elementos presentes em \mathcal{V}_A . Dessa maneira, \mathcal{V}_A pode ser dividido nos dois subconjuntos disjuntos de nós que representam os conjuntos de estudantes e exercícios. Além disso, cada aresta sempre conecta um estudante a um exercício, tornando \mathcal{A} um grafo bipartido. Essa aresta é ponderada pelo tempo gasto na tentativa e pode ser dos tipos `True` ou `False`, representando uma tentativa correta ou incorreta, respectivamente. Essa representação pode ser considerada como uma representação direta, uma vez que mantém os dados integralmente.

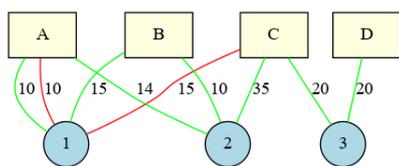


Figura 1. Exemplo do grafo \mathcal{A} .

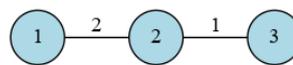


Figura 2. Exemplo do grafo \mathcal{S} .

Na Figura 1, pode-se observar um exemplo, com os dados da Tabela 1, onde os nós na cor azul e formato circular representam os estudantes 1, 2 e 3, e os nós na cor amarela e formato retangular representam os exercícios A, B, C e D. As arestas do tipo `True` foram representadas na cor verde, e as do tipo `False` na cor vermelha. Nesse caso, o estudante 1 realizou duas tentativas para o exercício A (uma incorreta e outra correta), e uma para os exercícios B e C. Por outro lado, o estudante 2 realizou uma tentativa para os exercícios A, B e C, e o estudante 3, uma tentativa para os exercícios C e D.

Contudo, essa representação mantém dois tipos de nós em um multigrafo, com isso, análises voltadas especificamente aos estudantes podem ser dificultadas. De forma mais específica, como observado no grafo exemplificado na Figura 1, não existem arestas entre estudantes, e por isso se considerarmos os exercícios resolvidos, por exemplo a relação entre eles é indireta. Dessa forma, uma possibilidade interessante é desconsiderar os nós de exercícios embutindo a informação contida na rede analisada.

Em vista desse fator, uma alternativa à primeira modelagem consiste em conectar os estudantes a partir dos exercícios resolvidos comuns entre eles. A intuição é que exercícios em comum podem indicar ao professor níveis de habilidades ou dificuldades compartilhadas por um grupo de estudantes.

Por isso, foi proposta a modelagem da rede de estudantes como um grafo simples não-direcionado $\mathcal{S} = (\mathcal{V}_S, \mathcal{E}_S)$. \mathcal{V}_S representa agora o conjunto de estudantes apenas, sendo removidos os nós de exercícios e \mathcal{E}_S o conjunto de arestas. Nessa representação, uma aresta é definida entre dois nós (estudantes) se existe ao menos um exercício em comum resolvido por eles. Além disso, a aresta é ponderada pelo número de exercícios

resolvidos em comum. Um exemplo pode ser observado na Figura 2, considerando o mesmo exemplo da Figura 1. Neste caso, o estudante 1 resolveu os exercícios A e B e o estudante 2, os exercícios A, B e C, resultando em dois exercícios em comum. Além disso, o estudante 3 resolveu os exercícios C e D, e tem um exercício em comum com 2.

Pode-se observar que essa representação permite um foco mais significativo nos estudantes, que agora estão ligados a partir de algo em comum, exercícios. Contudo, como visto na Tabela 2. Na prática, alguns exercícios podem ser resolvidos por quase todos os estudantes e, portanto, \mathcal{S} pode possuir diversas arestas. Por essa razão, pode ser difícil agrupar estudantes em comunidades semelhantes, já que quase todos tendem a ter ao menos um exercício resolvido em comum e, por isso, uma aresta.

Para contornar esse problema, foi utilizada uma variação da métrica *Neighborhood Overlap* (sobreposição de vizinhanças) [Easley and Kleinberg 2010], criando um subgrafo de \mathcal{S} a partir da remoção de arestas. De acordo com esta métrica, para dois determinados vértices, calcula-se um valor que representa a porcentagem de vizinhos em comum. E nessa variação, ao invés de considerar os vizinhos no grafo \mathcal{S} , foram selecionados o conjunto de exercícios resolvidos, representados por \mathcal{N} . Assim, a sobreposição de vizinhanças entre dois vértices de estudantes v_i e v_j pode ser definida pela Equação 1. Assim, o valor obtido varia no intervalo $[0, 1]$, pois o numerador é sempre menor ou igual ao denominador.

$$O_{v_i, v_j} = \frac{|\mathcal{N}(v_i) \cap \mathcal{N}(v_j)|}{|\mathcal{N}(v_i)|} \tag{1}$$

Para fins de ilustração, considere um exemplo simplificado do cálculo da métrica na Figura 3, onde os vizinhos dos nós de estudantes foram representados como os exercícios resolvidos por eles. Por exemplo, $\mathcal{N}(1) = \{A, B\}$ e $\mathcal{N}(2) = \{A, B, C\}$ e, portanto, $O_{(1,2)} = \frac{2}{2}$ e $O_{(2,1)} = \frac{2}{3}$. Note que $O_{(1,2)}$ é diferente de $O_{(2,1)}$. Nesse sentido, os valores obtidos definem o quão semelhante são dois estudantes considerando os exercícios resolvidos e os exercícios em comum.

Dessa forma, a modelagem final, um grafo $\mathcal{G} = (\mathcal{V}_{\mathcal{G}}, \mathcal{E}_{\mathcal{G}})$, é obtida a partir dessa métrica. Cada aresta de \mathcal{S} entre os nós v_i e v_j foi mantida em \mathcal{G} apenas se $O_{v_i, v_j} \geq k$ e $O_{v_j, v_i} \geq k$, sendo k um parâmetro definido experimentalmente (mais detalhes abaixo). Por exemplo, para $k = 0.50$, a aresta entre 2 e 3 é removida, já que $O_{2,3} = \frac{1}{3} \approx 0.33 < k$, resultando no grafo da Figura 4, subgrafo da Figura 2.

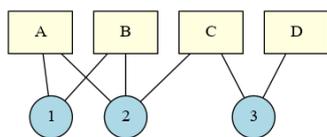


Figura 3. Grafo para cálculo da métrica *Neighborhood Overlap* – (O_{v_i, v_j}).

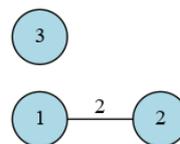


Figura 4. Grafo \mathcal{G} .

Por fim, a definição de k é feita em conjunto com a estratégia para detecção de comunidades no grafo. A ideia é que agrupar estudantes semelhantes possa evitar análises individuais dos estudantes, devido ao número de alunos. Para isso, foram construídos di-

ferentes grafos \mathcal{G} com cada k testado (0.00, 0.05, 0.10, ..., 1.00). Em cada \mathcal{G} foi realizada a detecção de comunidades, retornando um grupo para cada estudante. Nos experimentos foi usado o Algoritmo de Detecção de Comunidades de *Louvain* [Blondel et al. 2008], escolhido por seu desempenho em outros trabalhos ao formar comunidades com alta densidade de conexões [Araujo et al. 2023]. Vale ressaltar que os estudantes sem comunidade foram unidos em pseudogrupos, destacados com a indicação “s. c.” (sem comunidade) na Seção 4⁴.

Definidas as comunidades, foi obtida a média ponderada do desvio-padrão do número de exercícios resolvidos entre os estudantes dos mesmos grupos, sendo o peso da média o tamanho dos grupos. O valor de k selecionado é aquele que gera um grafo com a menor média. Considere um exemplo na Tabela 3, onde a métrica é avaliada a partir dos grafos da Figura 2 ($k = 0.0$) e da Figura 4 ($k = 0.5$). Sabe-se que, em ambos os grafos, os exercícios resolvidos são: $\mathcal{N}(1) = \{A, B\}$, $\mathcal{N}(2) = \{A, B, C\}$ e $\mathcal{N}(3) = \{C, D\}$. Assim, a partir de experimentos definiu-se $k = 0.5$, pois resultou em uma menor média. Com esse cálculo, pode-se verificar o quanto cada grupo está discrepante e, com isso, favorecer grupos mais coerentes. Vale destacar que um valor de k próximo a 1.0 tende a remover a maioria das arestas, mas também um valor próximo a 0.0 pode resultar em um grafo com um número elevado de arestas, dificultando o agrupamento e consequentemente a análise.

Tabela 3. Exemplo ilustrativo da seleção de k .

k	Grupos	Desvio Padrão dos Grupos	Média Ponderada
0.0	{1, 2, 3}	$\sqrt{\frac{(2-2.33)^2+(3-2.33)^2+(2-2.33)^2}{3}} \approx 0.47$	$\frac{0.47 \cdot 3}{3} \approx 0.47$
0.5	{1, 2} {3}	$\sqrt{\frac{(2-2.5)^2+(3-2.5)^2}{2}} = 0.5$ $\sqrt{\frac{(2-2)^2}{1}} = 0.0$	$\frac{0.5 \cdot 2 + 0.0 \cdot 1}{3} \approx 0.33$

3.3. Caracterização das Comunidades

Para caracterização das comunidades de estudantes, foram selecionadas algumas métricas consideradas relevantes, principalmente em relação ao desempenho. Nessa disciplina, o número de exercícios resolvidos é a métrica usada para atribuição de nota aos estudantes. A partir de cada comunidade detectada, são calculadas a média e desvio-padrão para as métricas: número de exercícios resolvidos (“Ex. Res.”), número de tentativas (“Tent.”), peso médio das arestas (“Peso M. das Ar.”) e tempos médios de resolução dos exercícios (“Tempos M. de Res.”). Um exemplo pode ser visto na Tabela 4, usando os dados de 2023. Na tabela, além do número de exercícios (“# Ex.”), cada comunidade detectada é identificada com uma letra (“Com.”)⁵, sendo descrito o número de estudantes (“# Est.”). O objetivo é que, com esses valores de cada grupo, o professor possa avaliar tanto a média do desempenho dos estudantes como também a semelhança entre eles. Espera-se que, a partir de então, seja possível identificar medidas pedagógicas adequadas de acordo com as especificidades apontadas pelos dados nos grupos.

⁴O algoritmo de detecção de comunidades pode manter alguns nós sem grupo.

⁵Além da indicação “s. c.” para os pseudogrupos, como descrito na Seção 3.2.

Tabela 4. Métricas de caracterização das comunidades de estudantes a partir dos dados de 2023.

# Ex.	Com.	# Est.	Ex. Res.	Tent.	Peso M. das Ar.	Tempos M. de Res.
53	A	14	44,93 (\pm 4,04)	62,86 (\pm 7,08)	39,53 (\pm 2,94)	14,48 (\pm 2,55)
	B	14	23,71 (\pm 6,92)	38,93 (\pm 11,68)	19,84 (\pm 5,83)	21,42 (\pm 4,05)
	C (s. c.)	13	11,69 (\pm 7,19)	23,31 (\pm 10,73)	0,00 (\pm 0,00)	38,75 (\pm 19,47)
	D	11	34,82 (\pm 2,44)	58,73 (\pm 8,20)	30,25 (\pm 2,70)	19,02 (\pm 2,78)
	E	4	15,75 (\pm 0,83)	26,25 (\pm 5,72)	12,63 (\pm 0,41)	35,32 (\pm 2,50)
	F	2	30,50 (\pm 1,50)	44,00 (\pm 4,00)	25,67 (\pm 1,67)	15,43 (\pm 4,63)
	G	2	7,50 (\pm 0,50)	17,00 (\pm 4,00)	6,00 (\pm 0,00)	40,70 (\pm 3,74)

4. Resultados

Nesta seção são apresentadas: 1) comparações estruturais entre as modelagens e 2) análise das métricas de caracterização. Os códigos dos experimentos foram implementados utilizando a linguagem *Python*, além do uso das ferramentas *dot* e *Gephi* para as visualizações dos grafos gerados. Para fins de reprodutibilidade do estudo, o código e os resultados completos da Tabela 4 podem ser acessados pelo repositório no GitHub⁶. Contudo, os dados utilizados não podem ser disponibilizados por conterem informações sigilosas.

4.1. Comparação entre Modelagens

A Tabela 5 apresenta características estruturais (i.e., topológicas) dos grafos descritas na Seção 3.2. As colunas iniciais detalham os números de vértices e arestas. Contudo, nas colunas “ Q_S ” e “ Q_G ” foi adicionado o cálculo da modularidade de \mathcal{S} e \mathcal{G} . A modularidade, descrita na Equação 2, compara os agrupamentos do grafo com os de um grafo aleatório, com valores altos indicando uma melhor repartição [Borba 2013]. Neste caso, m_p é o número de arestas e d_p é a soma dos graus dos vértices, considerando uma partição p . Neste caso, nota-se que o valor da modularidade aumenta, indicando que o algoritmo de *Louvain* revela comunidades mais estruturadas em \mathcal{G} em comparação a \mathcal{S} , por exemplo.

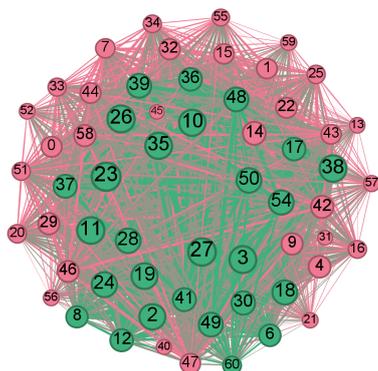
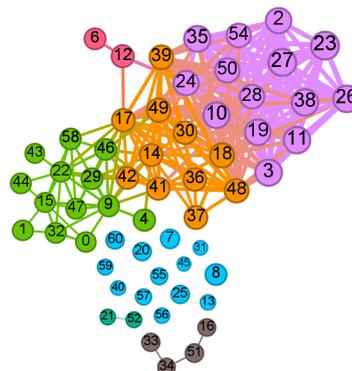
$$Q = \sum_p \left(\frac{m_p}{m} - \left(\frac{d_p}{2m} \right)^2 \right) \quad (2)$$

Além disso, observa-se que o número de vértices e arestas reduz significativamente de \mathcal{A} para \mathcal{S} e \mathcal{G} (\approx 25%), isso porque em \mathcal{S} e \mathcal{G} foram removidos os nós de exercícios. Em suma, essa remoção permite um foco nos estudantes, além de novas análises e diferentes métricas, já que \mathcal{A} é um multigrafo, enquanto \mathcal{S} e \mathcal{G} são grafos simples.

Tabela 5. Métricas estruturais dos grafos \mathcal{A} , \mathcal{S} e \mathcal{G} em cada ano.

Ano	$ \mathcal{V}_A $	$ \mathcal{E}_A $	$ \mathcal{V}_S $ e $ \mathcal{V}_G $	$ \mathcal{E}_S $	$ \mathcal{E}_G $	Q_S	Q_G	k
2018	73	2129	56	1481	298	0,02	0,25	0,65
2019	77	2486	59	1711	324	0,02	0,26	0,70
2021	68	1873	44	940	153	0,03	0,15	0,70
2022	69	2137	52	1321	291	0,02	0,20	0,70
2023	74	2601	59	1704	235	0,02	0,27	0,75

⁶Link para o repositório: https://github.com/jpmedras/assignment_net.

Figura 5. Grafo \mathcal{S} .Figura 6. Grafo \mathcal{G} .

Por outro lado, comparando \mathcal{S} e \mathcal{G} observa-se uma redução importante no número de arestas ($\approx 80\%$). Essa modificação se deve à definição do parâmetro k , que apresentou valores regulares com os dados de diferentes anos. Além disso, observa-se, por meio dos valores de Q , que a filtragem das arestas também resultou em melhores repartições.

Um exemplo do impacto da filtragem com o parâmetro k pode ser visto nas Figuras 5 e 6. Essas figuras foram geradas com os dados de 2023, a partir da modelagem dos grafos \mathcal{S} e \mathcal{G} , respectivamente. Pode-se observar uma redução de 86% no número de arestas e também maior modularidade entre agrupamentos (valores indicados na Tabela 5). Como as arestas não foram filtradas no grafo apresentada na Figura 5, menos comunidades foram formadas, já que os nós possuem muitas conexões. Em contrapartida, na Figura 6, as comunidades estão mais bem definidas, com arestas mais concentradas entre estudantes da mesma comunidade.

4.2. Avaliação das Métricas de Caracterização

Com o objetivo de agregar os dados e avaliar a formação das comunidades com base nas métricas, foi realizado o Teste de *Kruskal-Wallis* [Kruskal and Wallis 1952]. Esse teste compara a variância de cada métrica dentro dos grupos com sua variância entre os grupos, com isso, valores-p baixos indicam que a variância dos dados majoritariamente causada pelas diferenças entre os grupos. Os resultados do teste indicam alta qualidade nas comunidades formadas, já que em todos os anos e para todas as métricas, todos os valores-p foram < 0.01 , indicando estudantes semelhantes dentro grupos, mas significativamente diferentes entre os grupos.

Além disso, foi possível verificar que os alunos dos pseudogrupos (“s. c.”) possuem um alto índice de reprovação na disciplina, seja por falta ou por nota. Esses alunos, em geral, têm poucos exercícios resolvidos, e a maioria deles também foi resolvida pela maioria da turma. Com isso, na filtragem de arestas com k , quase todas as arestas são removidas, resultando em nós isolados no grafo. Este resultado corrobora o apresentado na Figura 6 (grupo na cor azul).

Por fim, com o objetivo verificar a escolha das métricas, foi calculado o coeficiente de correlação de *Pearson* [Benesty et al. 2009] da média do número de exercícios resolvidos em relação às médias das outras métricas utilizadas na Tabela 4, com os resultados apresentados na Figura 7. Na disciplina de origem dos dados, o número de exercícios resolvidos é a forma de pontuação das aulas práticas, por isso, essa variável foi esco-

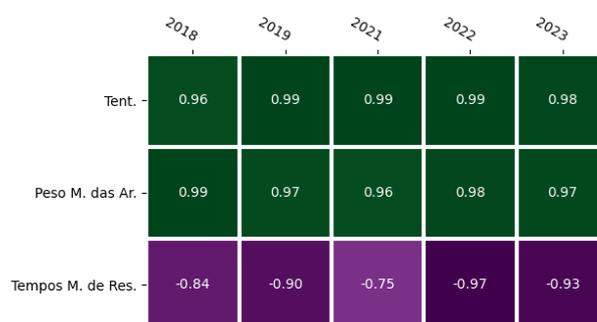


Figura 7. Correlações entre as médias das métricas de caracterização.

lhida como foco de comparação. Pode-se notar que as métricas possuem correlação, em módulo, próxima a 1,0 em todos os anos. Isso indica que o valor das métricas pode ser relativamente bem relacionado com a nota nas atividades. Dessa forma, entende-se o valor significativo das métricas, devido à relação com esse critério pedagógico.

5. Conclusão

Nesse trabalho, os dados de interação de estudantes foram explorados a partir de um JO utilizado em uma disciplina de programação durante cinco anos para extrair informações relevantes sobre os estudantes. Os dados, que representam o histórico de tentativas dos alunos para a resolução de exercícios de programação, foram explorados para a modelagem de uma rede de estudantes. Em seguida, eles foram agrupados e caracterizados a partir de diferentes métricas.

De forma geral, as descobertas desse estudo evidenciam o potencial da proposta como mecanismo diagnóstico de alunos, primeiro pela semelhanças entre dentro das comunidades e diferenças entre elas, mas também pela relevância das métricas avaliada em sua correlação com a nota dos estudantes nas aulas práticas. Com os valores de correlação obtidos, considera-se que as comunidades encontradas são relevantes e que as métricas indicam informações significativas sobre o desempenho dos estudantes, permitindo uma análise mais detalhada por parte do docente. Ademais, a abordagem fornece insumos valiosos para uma possível intervenção do professor a um grupo, e não somente a um estudante específico. Isso pode ser vantajoso, pois diminui o número de diferentes casos com os quais o professor deve focar.

Entre as limitações, destaca-se que os dados usados para avaliar a metodologia foram coletados de apenas uma disciplina em uma universidade específica. Contudo, uma vez que os dados usados não são específicos do JO, os experimentos podem ser expandidos para outras turmas ou disciplinas que registrem as tentativas de resolução dos exercícios em outros sistemas. Além disso, este trabalho disponibiliza acesso aos códigos utilizados nos experimentos, permitindo replicações.

Por fim, como trabalhos futuros, pretende-se disponibilizar a abordagem em uma plataforma interativa para suporte ao docente neste contexto. Nessa ferramenta, usuários poderão acompanhar as comunidades ao longo da disciplina a partir da integração com a base de dados do JO em um painel de visualizações interativas com os grafos e métricas gerados de forma semelhante ao proposto neste trabalho. A partir da ferramenta, a metodologia pode ser avaliada por professores em um contexto prático ao longo da disciplina.

Agradecimentos. FAPEMIG, PIBEN/Funarbe/PRE-UFV e CCE-UFV.

Referências

- Aguiar, G. O., Silva, M. A. P., and Silva, J. G. R. (2023). Análise do desempenho dos estudantes na disciplina de programação de computadores utilizando princípios de comunidades em redes complexas. In *Encontro Nacional de Computação dos Institutos Federais (ENCompIF)*, pages 1–4.
- Araujo, M., Ferreira, C., Reis, J., Silva, A., and Almeida, J. (2023). Identificação e caracterização de campanhas de propagandas eleitorais antecipadas brasileiras no twitter. In *Anais do Brazilian Workshop on Social Network Analysis and Mining (BrasNAM)*, pages 67–78.
- Barbosa, A. d. A., de Barros Costa, E., and Brito, P. H. (2023). Juízes online são suficientes ou precisamos de um var? In *Anais do Simpósio Brasileiro de Educação em Computação (SBIE)*, pages 386–394.
- Benesty, J., Chen, J., Huang, Y., and Cohen, I. (2009). *Pearson Correlation Coefficient*, pages 1–4. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*.
- Borba, E. M. (2013). Medidas de centralidade em grafos e aplicações em redes de dados. Master's thesis, Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Matemática Aplicada.
- Capovilla, D., Berges, M., Mühlhng, A., and Hubwieser, P. (2015). Handling heterogeneity in programming courses for freshmen. In *Proceedings of the International Conference on Learning and Teaching in Computing and Engineering*, pages 197–203.
- de Oliveira, M., Nogueira, M., and Oliveira, E. (2015). Sistema de apoio à prática assistida de programação por execução em massa e análise de programas. In *Anais do Workshop sobre Educação em Computação*, pages 90–99.
- Easley, D. and Kleinberg, J. (2010). *Networks, Crowds, and Markets: Reasoning about a Highly Connected World*. Cambridge University Press, 1 edition.
- Fernandes, J., Carvalho, L., Oliveira, D., Oliveira, E., Pereira, F., and Lauschner, T. (2023). Correlação entre complexidade e dificuldade de questões de programação em juízes online. In *Anais do Simpósio Brasileiro de Educação em Computação (SBIE)*, pages 97–107.
- Garin, R. S., Lichtnow, D., Palazzo, L. A. M., Loh, S., Kampff, A. J. C., Primo, T. T., Oliveira, J. P. M. d., and Lima, J. V. d. (2006). O uso de técnicas de recomendação em um sistema para apoio à aprendizagem colaborativa. *Revista Brasileira de Informática na Educação*, pages 49–59.
- Kruskal, W. H. and Wallis, W. A. (1952). Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*, 47(260):583–621.
- Lopes, C. C. (2003). Um sistema de apoio à tomada de decisão no acompanhamento do aprendizado em educação a distância. Master's thesis, Universidade Federal de Campina Grande. Programa de Pós-Graduação em Informática.

- Mohamad, S. K. and Tasir, Z. (2013). Educational data mining: A review. *Procedia - Social and Behavioral Sciences*, pages 320–324.
- Silva, C., Solano, J., Santos, A., and Reis, J. (2023). Previsão de reprovações em disciplinas introdutórias de programação: Um estudo em um ambiente de correção automática de códigos. In *Anais do Simpósio Brasileiro de Informática na Educação (SBIE)*, pages 1524–1535.
- Wasik, S., Antczak, M., Badura, J., Laskowski, A., and Sternal, T. (2018). A survey on online judge systems and their applications. *ACM Comput. Surv.*
- Weber, H., Schwenzer, M., and Hillmert, S. (2020). Homophily in the formation and development of learning networks among university students. *Network Science*, pages 469–491.
- Ícaro Alvim, Bittencourt, R. A., and Duran, R. (2024). Evasão nos cursos de graduação em computação no brasil. In *Anais do Simpósio Brasileiro de Educação em Computação (SBIE)*, pages 1–11.