

Avaliação da Qualidade de Testes Elaborados por Estudantes na Resolução de Problemas de Programação

André Almeida¹, Dalton D. S. Guerrero¹, Wilkerson L. Andrade¹

¹Universidade Federal de Campina Grande, Campina Grande - Paraíba, Brasil

andrealmeida@copin.ufcg.edu.br, dalton@computacao.ufcg.edu.br,

wilkerson@computacao.ufcg.edu.br

Abstract. *Solving programming problems requires understanding problem specifications and validating solutions through testing. However, many students face difficulties in designing test cases that accurately reflect the expected behavior of a program. This study investigates the quality of the tests produced during problem-solving activities, analyzing their ability to identify incorrect solutions. The analysis considers both performance metrics and structural aspects, such as equivalence classes and redundancy rate. The results indicate a good capacity for fault detection; however, a significant proportion of the tests are redundant or contribute little to the identification of incorrect behaviors.*

Resumo. *Resolver problemas de programação exige compreender enunciados e validar soluções por meio de testes. No entanto, muitos estudantes apresentam dificuldades em elaborar testes que reflitam corretamente o comportamento esperado. Este estudo investiga a qualidade dos testes produzidos durante a resolução de problemas, analisando sua capacidade de identificar soluções incorretas. A análise considera métricas de desempenho e aspectos estruturais, como classes de equivalência e taxa de redundância. Os resultados indicam boa capacidade de detecção de falhas; no entanto, uma parcela significativa dos testes é redundante ou apresenta baixa contribuição para a identificação de comportamentos incorretos.*

1. Introdução

O ensino de programação tem sido amplamente reconhecido como um dos maiores desafios na formação em Computação, especialmente nos estágios iniciais. Diversos estudos apontam que estudantes frequentemente apresentam dificuldades na compreensão de conceitos fundamentais, como abstração, controle de fluxo e decomposição de problemas, o que impacta diretamente sua capacidade de desenvolver soluções corretas e eficientes [Robins et al. 2003]. Além disso, a natureza abstrata da programação e a necessidade de raciocínio lógico estruturado tornam o processo de aprendizagem particularmente complexo, contribuindo para altos índices de evasão e reprovação em disciplinas introdutórias [Watson and Li 2014].

Nesse contexto, o uso de técnicas de teste de software tem sido apontado como uma estratégia relevante para apoiar o aprendizado de programação, ao incentivar os estudantes a refletirem sobre o comportamento esperado dos programas e a validarem

suas soluções de forma sistemática [Edwards 2004]. Além disso, estudos mais recentes indicam que a prática de escrever testes pode melhorar significativamente a qualidade do código produzido por estudantes, bem como sua capacidade de detectar falhas e compreender melhor os problemas propostos [Wrenn and Krishnamurthi 2019, Denny et al. 2025].

Diante desse contexto, este trabalho investiga a qualidade dos testes elaborados por estudantes, considerando-os como artefatos centrais de análise. O objetivo é avaliar em que medida esses testes são capazes de identificar soluções incorretas e de classificar corretamente o comportamento dos programas, bem como analisar aspectos estruturais relacionados à diversidade e redundância dos conjuntos de teste. Para orientar a investigação, foram definidas as seguintes questões de pesquisa:

- QP1: Quão eficazes são os testes elaborados pelos estudantes na detecção de soluções incorretas?
- QP2: Em que medida os testes dos estudantes classificam corretamente soluções corretas e incorretas?
- QP3: Como as métricas de precisão, revocação e acurácia se relacionam na avaliação dos testes produzidos pelos estudantes?
- QP4: Como os testes dos estudantes se distribuem em classes de equivalência e qual o impacto da redundância na efetividade desses testes?

A partir dessas questões, o trabalho analisa tanto métricas tradicionais de desempenho (como precisão, revocação e acurácia), quanto propriedades estruturais dos testes, incluindo classes de equivalência e taxa de redundância.

Como principais contribuições, este trabalho aplica uma abordagem para a avaliação da qualidade de testes elaborados por estudantes e fornece evidências empíricas sobre seu comportamento na elaboração desses testes, destacando a relação entre diversidade, redundância e efetividade na identificação de soluções incorretas.

2. Trabalhos Relacionados

Pesquisas na área de Educação em Computação têm investigado diferentes aspectos relacionados à qualidade dos testes elaborados por estudantes, ao ensino de testes de software e ao uso de métricas para avaliação da efetividade de suítes de teste. Em geral, os estudos apontam que estudantes apresentam dificuldades significativas na elaboração de casos de teste capazes de identificar comportamentos incorretos de forma consistente, especialmente em contextos introdutórios de programação.

Trabalhos que analisam diretamente suítes de teste produzidas por estudantes mostram que, mesmo em cursos avançados, os testes tendem a se concentrar em requisitos explícitos e cenários considerados “felizes”, negligenciando entradas inválidas, casos de borda e requisitos implícitos. Como consequência, observa-se frequentemente alta redundância entre os testes produzidos, com diferentes casos exercitando essencialmente os mesmos comportamentos do sistema, além de lacunas importantes de cobertura comportamental [Showler et al. 2025, Andleeb et al. 2025]. Esses resultados reforçam que a quantidade de testes produzidos não garante, necessariamente, maior efetividade na identificação de falhas.

Nesse contexto, diferentes estudos têm utilizado técnicas de teste de software, especialmente análise de mutação, para avaliar a qualidade dos testes elaborados pelos

estudantes. Resultados anteriores mostram que métricas tradicionais, como cobertura de código, frequentemente superestimam a qualidade das suítes de teste, enquanto a mutação apresenta maior capacidade de avaliar se os testes realmente distinguem comportamentos corretos e incorretos [Perretta et al. 2022, Neto et al. 2018, Shams and Edwards 2013]. Em ambientes educacionais, abordagens baseadas em mutação também têm demonstrado potencial para incentivar a construção de testes mais precisos e assertivos, embora ainda existam desafios relacionados ao custo computacional, à dependência de soluções de referência de alta qualidade e à aplicação em problemas mais abertos [Vasconcelos 2025, Potter et al. 2025].

Além da análise de efetividade, estudos na área de Engenharia de Software têm investigado estratégias para medir diversidade e redundância em suítes de teste. Em contextos industriais, técnicas baseadas em particionamento em classes de equivalência, agrupamento de comportamentos similares e métricas de diversidade são utilizadas para identificar testes redundantes, reduzir custos de execução e melhorar a manutenção das suítes de teste sem comprometer a capacidade de detecção de falhas [Mohammadian and Javed 2021, Marijan and Sen 2017]. Essas abordagens também permitem identificar lacunas de cobertura entre diferentes conjuntos de testes e apoiar decisões de otimização em larga escala.

Entretanto, a maior parte desses estudos concentra-se em contextos industriais, havendo poucas investigações voltadas ao uso de classes de equivalência, diversidade e redundância na análise de testes elaborados por estudantes de programação introdutória. Nesse contexto, este trabalho investiga empiricamente a qualidade dos testes produzidos pelos estudantes, relacionando classes de equivalência e redundância à capacidade de detectar soluções incorretas por meio de métricas como precisão, revocação e acurácia. Dessa forma, o estudo contribui para ampliar a compreensão sobre a efetividade dos testes elaborados em contextos educacionais.

3. Metodologia

Esta seção descreve a metodologia adotada neste estudo, detalhando os procedimentos utilizados para coleta e processamento dos dados, bem como as estratégias de avaliação aplicadas.

3.1. Estrutura do Experimento

O presente estudo investiga a relação entre a formulação de testes por estudantes e a capacidade desses testes em identificar falhas em programas. Para isso, foi conduzido um experimento estruturado em etapas, conforme ilustrado na Figura 1.

Inicialmente, realizou-se a definição dos problemas e a coleta das submissões dos estudantes, considerando diferentes implementações para uma mesma especificação, incluindo soluções corretas e incorretas. Em seguida, procedeu-se à definição dos testes, abrangendo tanto um conjunto de referência quanto os testes elaborados pelos próprios estudantes, compostos por pares de entrada e saída esperada. Nesse contexto, as saídas associadas às entradas fornecidas pelos estudantes são geradas automaticamente por uma ferramenta que possui as soluções de referência, apoiando o processo de validação. Importante salientar que tanto os testes quanto as soluções de referência utilizadas no processo de avaliação foram elaborados por três professores com mais de 20 anos de experiência em ensino de programação.

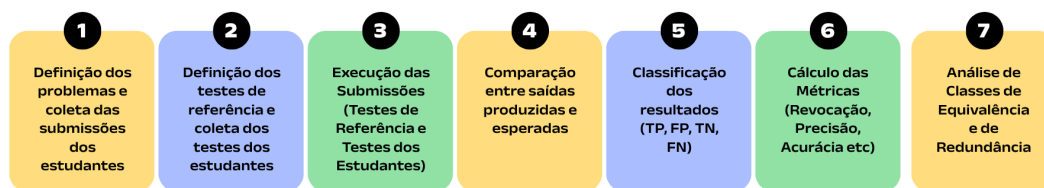


Figura 1. Etapas da metodologia adotada no estudo.

O *ground truth* adotado nesse estudo foi definido a partir dos testes de referência elaborados pelos especialistas. Dessa forma, uma solução foi considerada correta apenas quando produziu as saídas esperadas para todos os casos de teste de referência, enquanto soluções que apresentaram divergência em pelo menos um desses testes foram classificadas como incorretas.

A avaliação dos programas foi realizada comparando-se as saídas produzidas por cada submissão com as saídas esperadas definidas nos testes de referência. Uma submissão foi considerada incorreta quando apresentou divergência em pelo menos um caso de teste. Com base nessa avaliação, foi possível classificar os resultados de cada teste dos estudantes em termos de:

- Verdadeiro Positivo (TP): quando um teste identifica corretamente uma submissão incorreta;
- Falso Positivo (FP): quando um teste classifica como incorreta uma submissão que é, na realidade, correta;
- Verdadeiro Negativo (TN): quando um teste identifica corretamente uma submissão correta;
- Falso Negativo (FN): quando um teste falha em identificar uma submissão incorreta, classificando-a como correta.

A partir dessas classificações, foram calculadas métricas de desempenho para avaliar a qualidade dos testes:

- Taxa de Detecção de Falhas (TF): definido como a razão entre o número de submissões incorretas identificadas pelos testes e o total de submissões incorretas existentes, isto é, $TF = \text{nº de submissões incorretas detectadas} / \text{total de submissões incorretas}$.
- Revocação (*Recall*): equivalente à TF neste contexto, representando a proporção de submissões incorretas corretamente identificadas pelos testes. Mede a capacidade dos testes em detectar falhas.
- Precisão (*Precision*): calculada como $TP / (TP + FP)$, indicando o quão confiáveis são as falhas reportadas pelos testes, ou seja, a proporção de identificações corretas entre todas as detecções realizadas.
- Acurácia (*Accuracy*): definida como $(TP + TN) / (TP + TN + FP + FN)$, representando o desempenho geral dos testes ao considerar tanto a identificação de submissões corretas quanto incorretas.

Adicionalmente, foram realizadas análises de correlação entre as métricas de precisão, revocação e acurácia, com o objetivo de investigar possíveis relações entre diferentes aspectos da qualidade das suítes de testes produzidas pelos estudantes. Ressalta-se

que todas as métricas foram calculadas considerando como unidade de análise a **suíte de testes submetida por cada estudante**, e não testes individuais.

Por fim, foram conduzidas análises complementares envolvendo a identificação de classes de equivalência e o cálculo da taxa de redundância dos testes. Essas análises permitiram investigar a diversidade dos cenários de entrada explorados pelos estudantes e analisar como a distribuição desses cenários se relaciona com a efetividade dos testes na distinção entre soluções corretas e incorretas.

Nesse contexto, foram definidas **classes de equivalência** a partir de um conjunto de testes de referência, seguindo o princípio do particionamento em classes de equivalência descrito na literatura de teste de software [Ammann and Offutt 2017]. Cada classe agrupou testes com características de entrada semelhantes e que exercitavam o mesmo comportamento esperado da solução.

Em seguida, cada teste elaborado pelos estudantes foi comparado manualmente às classes previamente definidas. Quando um teste apresentava o mesmo comportamento de entrada e verificava o mesmo cenário funcional de uma classe existente, ele era associado a essa classe de equivalência. Caso contrário, o teste era considerado representativo de uma nova classe de equivalência identificada a partir das submissões dos estudantes. A classificação foi realizada por um pesquisador, seguindo critérios previamente definidos para caracterização dos cenários de entrada.

Para avaliar o grau de repetição entre os testes produzidos, foi calculada a **taxa de redundância**, definida como a proporção de testes que não acrescentam novas classes de equivalência, dada por $TR = \frac{T-C}{T}$, em que T representa o número total de testes produzidos, C o número de testes que representam a primeira ocorrência de uma classe de equivalência distinta e TR a taxa de redundância. Dessa forma, valores próximos de 1 indicam maior concentração de testes em classes já exploradas, enquanto valores próximos de 0 indicam maior diversidade de classes de equivalência. Por exemplo, uma suíte com 10 testes, dos quais apenas 6 pertencem a classes de equivalência distintas, apresenta $TR = \frac{10-6}{10} = 0,4$, indicando que 40% dos testes repetem classes de equivalência já exploradas.

3.2. Caracterização dos Exercícios de Programação

O estudo foi conduzido utilizando dois exercícios de programação introdutória selecionados por contemplarem diferentes aspectos de manipulação de dados, estruturas de repetição e controle de fluxo, além de permitirem avaliação automática das soluções. O Exercício 1 envolve a verificação da ordenação lexicográfica de palavras, exigindo comparação entre strings e identificação da primeira violação da ordem. Já o Exercício 2 aborda a análise de sequências de caracteres representando frequência de alunos, demandando processamento de múltiplas entradas, contagem de ocorrências e verificação de condições relacionadas à reprovação por faltas.

Os exercícios foram escolhidos por apresentarem múltiplos caminhos de execução e possibilidade de geração de erros sutis nas implementações, características relevantes para o estudo da construção de testes. Além disso, ambos permitem a definição clara de saídas esperadas, viabilizando a utilização de um oráculo de referência para geração automática das saídas e aplicação das métricas de análise utilizadas neste estudo. O funcionamento e a arquitetura desse oráculo são descritos em detalhes em nosso trabalho ante-

rior [Almeida et al. 2025], sendo utilizado nesta pesquisa como mecanismo para obtenção das saídas esperadas e avaliação das submissões.

3.3. Aspectos Éticos

Esta pesquisa foi conduzida em conformidade com as diretrizes éticas para estudos envolvendo seres humanos. O protocolo de pesquisa foi submetido e aprovado pelo Comitê de Ética em Pesquisa da instituição responsável antes do início da coleta de dados, sob registro CAAE nº 89199725.6.0000.5182. Todos os dados utilizados foram tratados de forma anonimizada, preservando a identidade e a privacidade dos participantes envolvidos no estudo.

4. Resultados e Discussões

Esta seção apresenta e discute os resultados obtidos a partir da análise dos testes elaborados pelos estudantes nos diferentes exercícios propostos. No total, foram consideradas 296 soluções, distribuídas entre dois exercícios, além de 899 testes produzidos pelos participantes. No Exercício 1, foram analisadas 150 soluções, sendo 44 incorretas e 106 corretas, juntamente com 517 testes elaborados pelos estudantes. Já no Exercício 2, foram avaliadas 146 soluções, das quais 57 foram classificadas como incorretas e 89 como corretas, além de 382 testes produzidos pelos participantes. Esses dados servem de base para a análise das métricas de desempenho e das propriedades estruturais dos testes apresentadas nas subseções a seguir.

4.1. QP1 - Taxa de Detecção de Falhas

A análise da taxa de detecção de falhas (Recall) revelou desempenho consistente dos testes elaborados pelos estudantes nos dois exercícios considerados, permitindo responder à QP1 (*Quão eficazes são os testes elaborados pelos estudantes na detecção de soluções incorretas?*). De maneira geral, os resultados indicam que os testes produzidos apresentam boa capacidade de identificar submissões incorretas, embora exista variabilidade significativa entre os estudantes.

Para o Exercício 1, foi observada uma média de 0,73, mediana de 0,88 e desvio padrão de 0,26. Já no Exercício 2, os valores foram ligeiramente superiores, com média de 0,77, mediana também de 0,88 e desvio padrão de 0,26.

A comparação entre média e mediana em ambos os exercícios indica uma distribuição assimétrica, na qual a mediana superior à média sugere que uma parcela significativa dos estudantes alcançou altos níveis de detecção de falhas, enquanto um subconjunto apresentou desempenho inferior, reduzindo o valor médio. Esse comportamento é corroborado pelos boxplots apresentados na Figura 2, nos quais é possível observar a concentração de valores elevados próximos ao topo da distribuição, bem como a presença de valores mais baixos que ampliam a dispersão.

O desvio padrão idêntico (0,26) nos dois exercícios indica um nível semelhante de variabilidade no desempenho dos estudantes, sugerindo que, independentemente do exercício, há uma heterogeneidade consistente na capacidade de formular testes eficazes. Esse resultado aponta para a existência de perfis distintos de estudantes: alguns conseguem elaborar testes com alta capacidade de detecção de falhas, enquanto outros apresentam dificuldades em cobrir adequadamente os comportamentos incorretos das submissões.

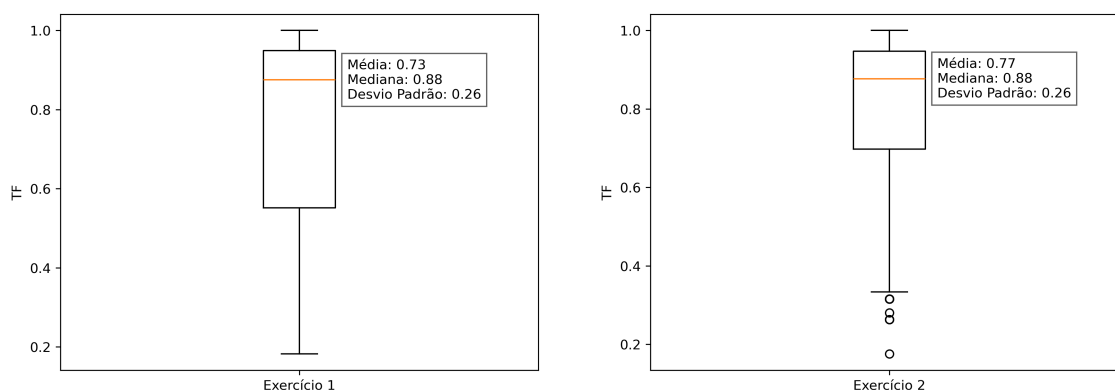


Figura 2. Distribuições das taxas de detecção de falhas para os dois exercícios utilizados.

No caso do Exercício 2, observa-se ainda a presença de outliers abaixo do primeiro quartil, com valores de taxa de detecção de falhas inferiores a 0,3. Esses casos indicam a existência de um pequeno grupo de estudantes cujos testes apresentam baixa efetividade, possivelmente por cobrirem apenas cenários triviais ou não explorarem adequadamente condições críticas do problema. A presença desses outliers contribui para a redução da média e reforça a assimetria da distribuição, ainda que não afete significativamente a mediana.

Apesar dessa variabilidade, os valores de mediana elevados (0,88) em ambos os exercícios indicam que, para pelo menos metade dos estudantes, os testes construídos foram altamente eficazes na identificação de falhas. Esse resultado sugere que a atividade proposta favorece, em geral, a criação de testes com bom poder de detecção, ainda que exista espaço para melhoria na consistência entre os participantes.

Por fim, a leve superioridade da média no Exercício 2 (0,77 em comparação a 0,73 no Exercício 1) pode indicar que esse exercício apresenta características que facilitam a identificação de falhas por meio de testes, como padrões mais evidentes de erro ou menor complexidade na definição de casos críticos. No entanto, como a mediana e o desvio padrão permanecem iguais, essa diferença parece estar associada principalmente à redução de casos de baixo desempenho, e não a uma mudança significativa no comportamento geral da distribuição.

4.2. QP2 - Precisão e Acurácia

A análise das métricas de precisão e acurácia permitiu responder à QP2 (*Em que medida os testes dos estudantes classificam corretamente soluções corretas e incorretas?*). De maneira geral, os resultados indicam que os testes elaborados pelos estudantes apresentam boa capacidade de classificação, especialmente no Exercício 2, embora ainda exista variabilidade entre os participantes.

No Exercício 1, a precisão apresentou média de 0,77, mediana de 0,81 e desvio padrão de 0,21, indicando confiabilidade moderada na identificação de falhas, mas com elevada variação entre os estudantes. A acurácia apresentou média de 0,83, mediana de 0,81 e desvio padrão de 0,10, sugerindo desempenho geral relativamente consistente.

Já no Exercício 2, observou-se desempenho superior. A precisão atingiu média

de 0,97, mediana de 0,98 e desvio padrão de 0,05, indicando testes altamente confiáveis, com baixa incidência de falsos positivos e comportamento mais homogêneo entre os estudantes. A acurácia também apresentou melhora, com média de 0,90, mediana de 0,94 e desvio padrão de 0,10. Conforme ilustrado nos boxplots da Figura 3, o Exercício 2 apresentou distribuições mais concentradas em valores elevados, embora ainda existam outliers com desempenho inferior.

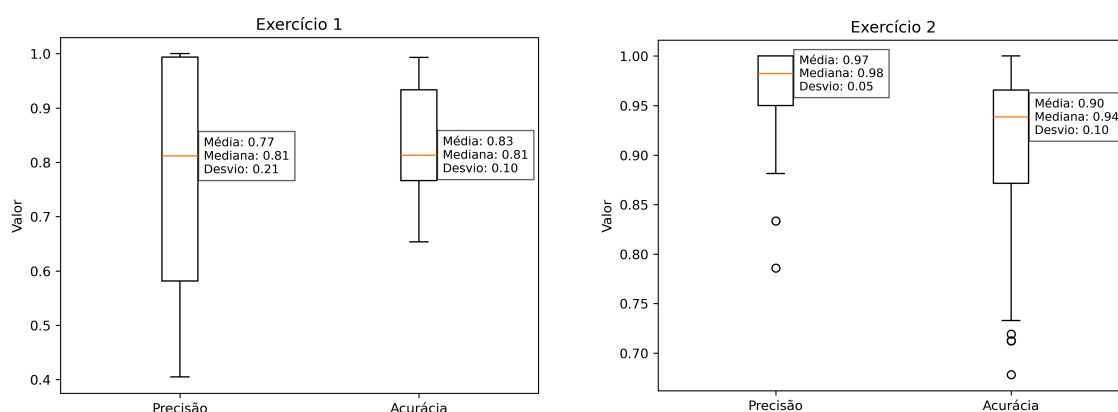


Figura 3. Distribuições da qualidade da classificação dos testes para os exercícios utilizados.

Esses casos de menor precisão e acurácia não estão associados a erros na definição das saídas esperadas, uma vez que os estudantes não as especificam manualmente. Como as saídas foram obtidas automaticamente por meio do oráculo de referência descrito anteriormente, buscou-se eliminar esse tipo de erro na construção dos testes. Assim, as limitações observadas estão relacionadas principalmente à seleção de entradas pouco discriminativas, incapazes de distinguir adequadamente comportamentos corretos e incorretos das submissões avaliadas.

A comparação entre os exercícios revela que a principal melhoria no Exercício 2 está relacionada à escolha de entradas mais adequadas, capazes de reduzir falsos positivos e aumentar a confiabilidade dos testes. Em conjunto com os resultados de taxa de detecção de falhas, esses dados indicam que o Exercício 2 favoreceu não apenas a identificação de falhas, mas também a construção de testes mais informativos e consistentes.

4.3. QP3 - Correlação entre as métricas

A análise das correlações entre as métricas de precisão, revocação e acurácia permitiu responder à QP3 (*Como as métricas de precisão, revocação e acurácia se relacionam na avaliação dos testes produzidos pelos estudantes?*). Para isso, foi utilizado o coeficiente de correlação de Pearson, que mede o grau de relação linear entre variáveis. Os resultados evidenciam padrões distintos entre os dois exercícios, indicando diferenças na forma como as métricas se relacionam durante a construção dos casos de teste pelos estudantes.

No Exercício 1 (Figura 4), observa-se uma correlação negativa fraca entre precisão e revocação ($r = -0,13$), sugerindo um leve *trade-off* entre essas métricas. Isso indica que, em alguns casos, testes capazes de identificar um número maior de soluções incorretas tendem, ainda que de forma limitada, a apresentar maior incidência de classificações

incorretas. Entretanto, como a magnitude da correlação é baixa, esse comportamento não representa uma tendência predominante entre os estudantes.

Por outro lado, verifica-se uma correlação positiva forte entre precisão e acurácia ($r = 0,73$), indicando que a redução de falsos positivos contribui significativamente para o desempenho geral dos testes. A correlação moderada entre revocação e acurácia ($r = 0,55$) mostra que a capacidade de identificar soluções incorretas também influencia a qualidade global dos testes, embora com menor intensidade do que a precisão. Em conjunto, esses resultados indicam que, no Exercício 1, tanto a precisão quanto a revocação contribuem para a acurácia, sendo a precisão o fator de maior influência.

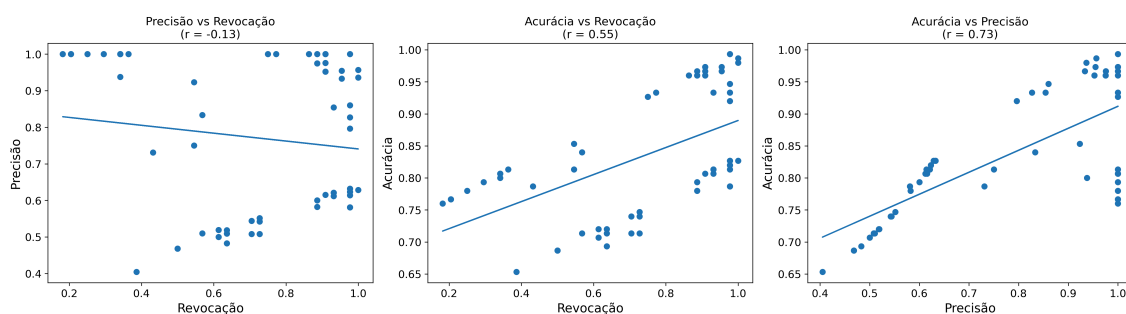


Figura 4. Correlações entre as métricas analisadas para o Exercício 1.

No Exercício 2 (Figura 5), o padrão observado é diferente. A correlação entre precisão e revocação permanece negativa e fraca ($r = -0,17$), indicando novamente um leve *trade-off*. Entretanto, destaca-se a correlação extremamente forte entre revocação e acurácia ($r = 0,99$), evidenciando que o desempenho geral dos testes passa a depender quase exclusivamente da capacidade de identificar soluções incorretas. Em contrapartida, a correlação entre precisão e acurácia é praticamente nula ($r = -0,01$), indicando que a precisão exerce pouca influência sobre a acurácia nesse exercício.

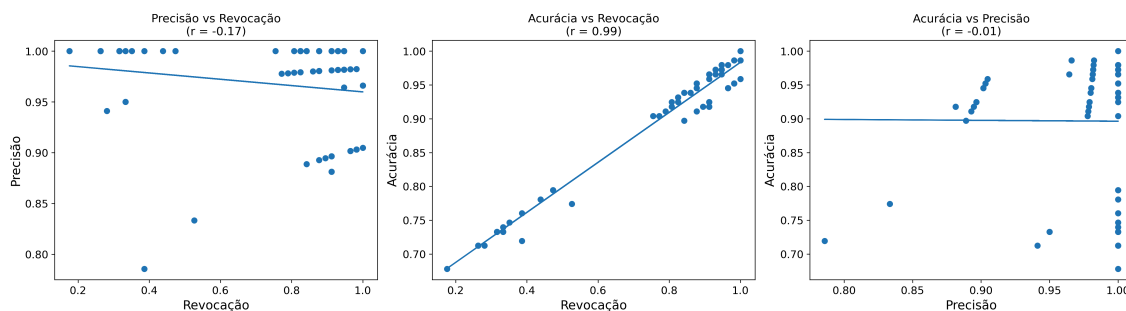


Figura 5. Correlações entre as métricas analisadas para o Exercício 2.

Esse comportamento pode ser explicado pelos resultados apresentados na seção anterior. No Exercício 2, a precisão apresentou valores elevados e pouca variabilidade entre os estudantes, reduzindo seu poder discriminatório. Assim, como a maioria dos participantes produziu testes com alta confiabilidade, as diferenças na acurácia passaram a ser explicadas principalmente pela revocação, isto é, pela capacidade de identificar soluções incorretas.

A comparação entre os exercícios evidencia uma mudança no fator predominante para o desempenho dos testes. Enquanto no Exercício 1 tanto a precisão quanto

a revocação influenciam a acurácia, com maior contribuição da precisão, no Exercício 2 a revocação torna-se o principal determinante do desempenho. Esse resultado sugere que, à medida que os estudantes passam a produzir testes mais confiáveis, o principal desafio deixa de ser evitar classificações incorretas e passa a ser elaborar entradas capazes de revelar um maior número de falhas.

De forma geral, os resultados mostram a existência de um leve *trade-off* entre precisão e revocação em ambos os exercícios, embora sua baixa magnitude indique que esse comportamento não compromete significativamente a qualidade dos testes produzidos. Além disso, a mudança observada entre os exercícios reforça que a análise conjunta das métricas fornece uma visão mais abrangente sobre a evolução da qualidade dos testes elaborados pelos estudantes.

4.4. QP4 - Classes de Equivalência e Análise da Taxa de Redundância

A análise das classes de equivalência e da taxa de redundância permitiu responder à QP4 (*Como os testes dos estudantes se distribuem em classes de equivalência e qual o impacto da redundância na efetividade desses testes?*). De maneira geral, os resultados indicam que os estudantes concentraram grande parte dos testes em classes de equivalência já previstas, embora o Exercício 2 tenha apresentado maior diversidade estrutural e menor redundância entre os casos produzidos.

No Exercício 1, 473 testes (**91,49%**) foram associados às classes de equivalência de referência, enquanto apenas 12 testes (**2,32%**) representaram novas classes, relacionadas principalmente à exploração de entradas vazias. Esse resultado sugere que os estudantes tendem a reproduzir cenários já identificados no conjunto de referência, apresentando maior dificuldade em propor entradas alternativas capazes de explorar comportamentos adicionais do problema. Além disso, 32 testes (**6,19%**) foram considerados inválidos por violarem restrições do enunciado, utilizando caracteres, formatos ou valores não permitidos. Esse comportamento evidencia que parte dos estudantes ainda apresenta dificuldades na interpretação das especificações antes mesmo da elaboração dos casos de teste.

No Exercício 2, 220 testes (**57,59%**) permaneceram associados às classes de equivalência de referência, enquanto 78 testes (**20,42%**) foram classificados como novas classes de equivalência, indicando maior capacidade de explorar cenários adicionais em comparação ao Exercício 1. Entretanto, a análise desses novos casos revelou que uma parcela significativa estava associada a situações simples, como entradas envolvendo apenas um aluno, sem faltas ou com quantidade de faltas próxima ao limite de aprovação. Embora esses testes ampliem a diversidade estrutural, eles nem sempre representam cenários mais informativos ou capazes de revelar falhas em implementações incorretas. Esse resultado reforça que a diversidade de entradas, isoladamente, não é suficiente para avaliar a qualidade de uma suíte de testes. Além disso, 84 testes (**21,99%**) foram considerados inválidos por utilizarem entradas incompatíveis com as restrições do problema.

A Figura 6 apresenta a distribuição da diversidade de classes de equivalência exploradas pelos estudantes. No Exercício 1, observa-se maior concentração entre 1 e 5 classes distintas por aluno, com predominância de estudantes que exploraram 3 classes (**18 alunos**). Já no Exercício 2, embora tenha ocorrido aumento no número de novas classes identificadas, a diversidade permaneceu concentrada em poucos cenários, principalmente 2 (**28 alunos**) e 3 (**19 alunos**) classes distintas. Esse comportamento sugere

que os estudantes passaram a explorar mais possibilidades de entrada, mas ainda apresentavam dificuldade em distribuir seus testes entre diferentes categorias de comportamento do problema.

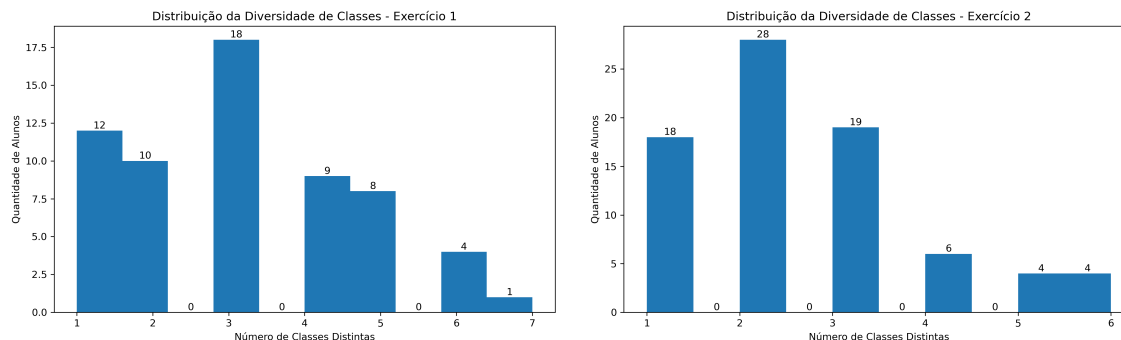


Figura 6. Distribuição da diversidade de classes de equivalência exploradas pelos estudantes nos exercícios.

Em relação à redundância, no Exercício 1, aproximadamente **19,35%** dos estudantes produziram conjuntos de testes sem redundância significativa, enquanto uma parcela significativa apresentou repetição de cenários pertencentes às mesmas classes de equivalência. No Exercício 2, esse resultado foi mais positivo, com cerca de **51,90%** dos estudantes apresentando taxas de redundância próximas de zero, indicando maior complementaridade entre os testes produzidos. Entretanto, a existência de estudantes com alta redundância demonstra que alguns participantes ainda possuem dificuldade em reconhecer quando novos testes acrescentam informações relevantes ao processo de avaliação.

Esses resultados indicam que, embora os estudantes tenham ampliado a diversidade de cenários explorados no Exercício 2 e produzido conjuntos de testes com menor redundância, ainda existem limitações na seleção de entradas capazes de representar comportamentos relevantes do problema. A criação de novos casos de teste, por si só, não garante maior efetividade, uma vez que testes associados a cenários semelhantes ou pouco discriminativos podem não contribuir para a identificação de soluções incorretas. Dessa forma, os resultados reforçam a importância de considerar conjuntamente aspectos como diversidade, validade e capacidade de detecção de falhas na avaliação da qualidade dos testes produzidos em ambientes de programação introdutória suportados por testes.

5. Conclusões e Trabalhos Futuros

A análise da taxa de detecção de falhas mostrou que, de maneira geral, os testes elaborados pelos estudantes conseguiram identificar uma parcela relevante das submissões incorretas. Em ambos os exercícios, observou-se mediana de **0,88**, indicando boa capacidade de distinção entre soluções corretas e incorretas. Entretanto, a existência de estudantes com baixos valores de detecção evidencia dificuldades na elaboração de entradas realmente discriminativas, sugerindo que parte dos estudantes ainda associa a criação de testes à simples verificação de exemplos de funcionamento esperado, em vez de explorá-los como uma estratégia para revelar comportamentos incorretos.

A análise das classes de equivalência e da taxa de redundância complementa esses resultados. No Exercício 1, apenas **2,32%** dos testes corresponderam a novas classes

de equivalência válidas, enquanto **91,49%** permaneceram associados às classes já previstas. No Exercício 2, houve aumento da diversidade estrutural, com **20,42%** de novas classes e redução significativa da redundância. Contudo, muitas dessas novas classes correspondiam a cenários simples, mostrando que maior diversidade não implica, necessariamente, maior capacidade de detectar falhas. Além disso, parte dos testes foi considerada inválida por violar restrições dos enunciados, evidenciando dificuldades na interpretação das especificações e na definição do domínio de entrada dos problemas.

Do ponto de vista pedagógico, esses resultados indicam que a elaboração de testes pode ser explorada como uma habilidade complementar à resolução de problemas de programação, auxiliando os estudantes a refletirem sobre diferentes comportamentos de uma solução. Estratégias como discussão de classes de equivalência, identificação de casos extremos e *feedback* sobre validade, redundância e detecção de falhas podem contribuir para que os estudantes compreendam que a qualidade de um conjunto de testes depende mais da capacidade de representar cenários relevantes e identificar erros do que da quantidade de entradas produzidas. Essa constatação complementa trabalhos anteriores sobre geração e avaliação automática de suítes de testes, como [Perretta et al. 2022, Shams and Edwards 2013], ao indicar que métricas estruturais, isoladamente, não são suficientes para caracterizar a qualidade dos testes produzidos em contextos educacionais.

Como trabalhos futuros, pretende-se investigar mecanismos automáticos de apoio à elaboração de testes que integrem métricas de detecção de falhas, diversidade, redundância e validade das entradas, fornecendo *feedback* contínuo aos estudantes durante a resolução de exercícios de programação.

6. Ameaças à Validade

Este estudo apresenta algumas ameaças à validade que devem ser consideradas. Em relação à **validade interna**, os resultados dependem do conjunto de submissões utilizado na avaliação, uma vez que a detecção de falhas, as classes de equivalência e a redundância são influenciadas pelos padrões de erro presentes nas soluções analisadas. Além disso, a definição das classes a partir de testes de referência pode introduzir risco de circularidade. Para minimizar esse efeito, testes que não correspondiam às classes iniciais foram considerados como representantes de novas classes de equivalência.

Quanto à **validade externa**, a principal limitação está relacionada ao uso de apenas dois exercícios introdutórios de programação, ambos com domínio de entrada restrito. Embora esses problemas permitam diferentes comportamentos e padrões de erro, os resultados podem não ser generalizáveis para outros níveis de complexidade, linguagens ou contextos educacionais. Essa ameaça foi parcialmente reduzida pelo uso de um conjunto expressivo de dados, composto por 296 soluções e 899 testes produzidos pelos estudantes.

Por fim, em relação à **validade de construto**, as métricas utilizadas (detecção de falhas, precisão, revocação, acurácia, diversidade e redundância) representam diferentes aspectos da qualidade dos testes, mas não capturam completamente o processo de elaboração dos estudantes. Optou-se por utilizar submissões reais em vez de análise de mutação, pois o objetivo era avaliar a identificação de erros produzidos em um contexto educacional real. A comparação com abordagens baseadas em mutação constitui uma oportunidade para trabalhos futuros.

7. Declaração sobre uso de Inteligência Artificial

Ferramentas de Inteligência Artificial generativa foram utilizadas como apoio durante a elaboração deste trabalho, principalmente para revisão gramatical, refinamento da escrita acadêmica e reorganização textual de trechos previamente produzidos pelos autores. Todas as análises, interpretações dos resultados, definições metodológicas e conclusões apresentadas são de responsabilidade dos autores.

8. Disponibilidade dos Artefatos

Com o objetivo de favorecer a reprodutibilidade deste estudo, os scripts desenvolvidos para o processamento dos dados, cálculo das métricas (precisão, revocação, acurácia, classes de equivalência e taxa de redundância) e geração das tabelas e figuras estão disponibilizados publicamente no Google Drive¹.

Os dados utilizados neste estudo não serão disponibilizados, pois contêm informações que podem permitir a identificação dos estudantes participantes, em conformidade com os princípios éticos e de proteção de dados adotados na pesquisa.

Referências

- Almeida, A., Guerrero, D., and Andrade, W. (2025). Avaliando o efeito da criação de testes na aprendizagem e no desempenho de estudantes de programação. In *Anais do XXXVI Simpósio Brasileiro de Informática na Educação*, pages 140–152, Porto Alegre, RS, Brasil. SBC.
- Ammann, P. and Offutt, J. (2017). *Introduction to software testing*. Cambridge University Press.
- Andleeb, S., Mendoza, T., Cordova, L. P., Walia, G., and Carver, J. (2025). Enhancing software testing education: Understanding where students struggle. In *Proceedings of the 26th ACM Annual Conference on Cybersecurity & Information Technology Education*, pages 154–160.
- Denny, P., Kumar, V., MacNeil, S., Prather, J., and Leinonen, J. (2025). Probing the unknown: Exploring student interactions with probeable problems at scale in introductory programming. In *Proceedings of the 30th ACM Conference on Innovation and Technology in Computer Science Education V. 1*, pages 618–624.
- Edwards, S. H. (2004). Using software testing to move students from trial-and-error to reflection-in-action. In *Proceedings of the 35th SIGCSE technical symposium on Computer science education*, pages 26–30.
- Marijan, D. and Sen, S. (2017). Detecting and reducing redundancy in software testing for highly configurable systems. In *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*, pages 96–99. IEEE.
- Mohammadian, M. and Javed, Z. (2021). Intelligent evaluation of test suites for developing efficient and reliable software. *International Journal of Parallel, Emergent and Distributed Systems*, 36(2):159–188.

¹<https://drive.google.com/drive/folders/14sppCWkK0I85t4m1HqrQp0naerICx5oO?usp=sharing>

- Neto, F. G. D. O., Feldt, R., Erlenhov, L., and Nunes, J. B. D. S. (2018). Visualizing test diversity to support test optimisation. In *2018 25th Asia-Pacific Software Engineering Conference (APSEC)*, pages 149–158. IEEE.
- Perretta, J., DeOrio, A., Guha, A., and Bell, J. (2022). On the use of mutation analysis for evaluating student test suite quality. In *Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis*, pages 263–275.
- Potter, H., Paiva, A. C., Amalfitano, D., Fasolino, A. R., Tramontana, P., and Just, R. (2025). Evaluating the impact of scaffolding and visualizations for mutation testing exercises in software engineering education. In *Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering*, pages 823–834.
- Robins, A., Rountree, J., and Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer science education*, 13(2):137–172.
- Shams, Z. and Edwards, S. H. (2013). Toward practical mutation analysis for evaluating the quality of student-written software tests. In *Proceedings of the ninth annual international ACM conference on International computing education research*, pages 53–58.
- Showler, A., Miljanovic, M. A., and Bradbury, J. S. (2025). How effective and efficient are student-written software tests? In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1*, pages 1057–1063.
- Vasconcelos, P. (2025). Exploring mutation testing for teaching introductory programming. In *6th International Computer Programming Education Conference (ICPEC 2025)*, pages 1–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.
- Watson, C. and Li, F. W. (2014). Failure rates in introductory programming revisited. In *Proceedings of the 2014 conference on Innovation & technology in computer science education*, pages 39–44.
- Wrenn, J. and Krishnamurthi, S. (2019). Executable examples for programming problem comprehension. In *Proceedings of the 2019 ACM conference on international computing education research*, pages 131–139.