

Objetos de Aprendizagem para Apoiar a Prática de Testes Caixa Preta

Maria Clara de Medeiros Meira¹, Geísa Moraes Gabriel², Alysso Filgueira Milanez², David Candeia Medeiros Maia³, Roberta Coelho¹, Marcia Lucena¹

¹Universidade Federal do Rio Grande do Norte (UFRN) - Natal - RN - Brasil

²Universidade Federal Rural do Semi-Árido (UFERSA) - Pau dos Ferros - RN - Brasil

³Instituto Federal da Paraíba (IFPB) - Campina Grande - PB - Brasil

clara.meira.017@ufrn.edu.br, geisa.morais@alunos.ufersa.edu.br,
alysso.milanez@ufersa.edu.br, david.candeia@academico.ifpb.edu.br,
{roberta, marciaj}@dimap.ufrn.br

Abstract. *Ensuring the quality and reliability of software is fundamental to guaranteeing its functionality and performance. There is a notable lack of materials in Portuguese regarding software testing, especially black-box testing. This study presents learning objects developed to investigate the effectiveness of black-box techniques and strategies, incorporating didactic resources about their operation, applied to different scenarios. The analysis of the objects demonstrates the effectiveness of black-box testing practices, such that the assertive choice of techniques refines the performance of testing activities.*

Keywords *Software testing, black box testing, learning objects.*

Resumo. *Assegurar a qualidade e confiabilidade de um software é fundamental para garantir o seu funcionamento e desempenho. É notória a carência de materiais no idioma português acerca da realização de testes de software, especialmente os de caixa preta. Este estudo apresenta objetos de aprendizagem, desenvolvidos para investigar a efetividade das técnicas e estratégias caixa preta, incorporando recursos didáticos acerca de seu funcionamento, aplicados a cenários distintos. A análise dos objetos demonstra efetividade da prática caixa preta, de modo que a escolha assertiva das técnicas refina o desempenho das atividades de teste.*

Palavras-Chave *Teste de software, testes caixa preta, objetos de aprendizagem.*

1. Introdução

Considera-se que os erros e falhas de software podem significar consequências gravíssimas durante o desenvolvimento ou entrega de um projeto, dada a qualidade como um aspecto crítico [Delamaro et al. 2016]. Assim, tem-se que o teste de software é o processo de execução de um produto para verificar se o mesmo atingiu os objetivos das especificações e funcionou conforme o ambiente em que foi projetado [Neto 2022].

A atividade de teste é indispensável no processo de desenvolvimento, o que provoca a busca por pessoal capacitado; no entanto, o meio acadêmico não tem conseguido atender satisfatoriamente à demanda existente [Barbosa et al. 2004]; além da evidente quantidade restrita de conteúdos nesta área [Benitti e Albano 2012]. Mesmo

com ênfase à importância das disciplinas de teste de software, a valorização da prática e o aumento de sua procura; existem lacunas e problemas relacionados às atividades de teste durante a formação de estudantes, que se estendem até o mercado de trabalho [Nascimento et al. 2024]. Tal fato associa-se à frequente desmotivação a respeito da realização dos planos e projetos para executar casos de teste mais assertivos, uma vez que, quando abordados, são vistos como muito teóricos e pouco práticos [Nascimento et al. 2024]. Além disso, este pode ser um tópico difícil de aprender ou ensinar sem a utilização dos mecanismos de apoio adequados [Maldonado et al. 2006].

A utilização de metodologias de teste torna-se necessária para garantir sua adequação funcional, visto que seu objetivo principal é revelar possíveis falhas ao longo do desenvolvimento [Delamaro et al. 2016]. Nesse sentido, destaca-se a necessidade da construção de objetos de aprendizagem de acesso aberto que exemplifiquem em meios ilustrativos como as metodologias de teste podem ser utilizadas pelos estudantes ainda durante a sua formação e, posteriormente, para desenvolver competências profissionais.

Os testes de caixa preta ou testes funcionais, realizam análises externas para avaliar o comportamento e execução das funcionalidades de um software por meio da criação de casos de teste e aplicação de técnicas [Delamaro et al. 2016]. Analisar cada técnica individualmente possibilitará construir uma perspectiva mais abrangente dos testes funcionais, estabelecendo um panorama referente ao seu desempenho e efetividade. A partir da implementação ilustrativa em diferentes cenários, este trabalho busca promover a compreensão quanto aos contextos nos quais as técnicas se tornam mais apropriadas, permitindo uma escolha mais assertiva e eficaz.

Diante da possível escassez de materiais sobre teste de software, observada durante o levantamento bibliográfico realizado, e a maior incidência de materiais escritos na língua inglesa; há uma insuficiência dessa abordagem em cursos da área de Computação e Tecnologia da Informação [Nascimento et al. 2024], principalmente dos testes funcionais (caixa preta). Este trabalho reúne estratégias e técnicas, e serve como um recurso metodológico para aplicações reais no âmbito acadêmico e profissional. Ademais, propõe-se uma investigação futura acerca da utilização deste material sob a perspectiva de estudantes, para avaliar a sua eficiência no ensino-aprendizagem dos testes caixa preta.

Este trabalho visa contribuir para a comunidade acadêmica e profissional, por meio do desenvolvimento de objetos de aprendizagem dos testes funcionais no idioma português, de modo que promova: (1) a elucidação das suas metodologias e estratégias, (2) a investigação das técnicas e a sua ilustração em circunstâncias distintas de forma assertiva; e, (3) a avaliação da eficácia dessas abordagens para a detecção de problemas.

2. Técnicas de Teste Caixa Preta

Os testes de caixa preta são metodologias e técnicas relacionadas à análise externa do sistema [Delamaro et al. 2016, Chrisantyo et al. 2022], diante disso, existem diferentes técnicas para detectar falhas ou inconsistências em sistemas. O Particionamento por Equivalência divide o domínio de entradas em classes de equivalência que possuem entradas válidas e inválidas. Já a Análise de Valor Limite visa casos de teste com maior probabilidade de encontrar defeitos utilizando valores que podem estar acima ou abaixo do limite pré-definido; quando combinadas, geram um novo método, o Teste Funcional Sistemático [Delamaro et al. 2016].

O *Error Guessing*, é uma análise minuciosa com base na experiência do testador [Delamaro et al. 2016]. O Grafo de Causa e Efeito e a Tabela de Decisão [Delamaro et al. 2016] exploram possíveis inconsistências, mapeando ações e as possíveis consequências. O Teste Combinacional mostra-se útil ao realizar testes com múltiplos cenários, entradas e ações [Kaner et al. 2011].

3. Trabalhos Relacionados

Em relação à criação de um material instrucional como apoio ao ensino de teste de software, [Barbosa et al. 2004] integram práticas de projeto instrucional, modelagem de conteúdos educacionais, e aspectos de desenvolvimento livre e colaborativo, que compõem módulos educacionais aplicados a testes. Análogo a isso, [Maldonado et al. 2006] exibem mecanismos para a criação de módulos educacionais, ilustrados pelo desenvolvimento de um módulo educacional voltado ao teste de mutação.

[Diniz e Dazzi 2011] desenvolveram um jogo educacional orientado ao ensino dos testes caixa preta, para alunos da graduação e profissionais cuja formação não contempla ênfase nessa área, explorando somente o conhecimento e aplicação das técnicas análise de valor limite e classes de equivalência. Similarmente, [Severo e Lelli 2023] aplicaram um jogo educativo para o ensino de testes funcionais e especificações, considerando apenas aspectos introdutórios relacionados à criação de casos de teste, aplicando estrategicamente após as aulas iniciais sobre testes funcionais.

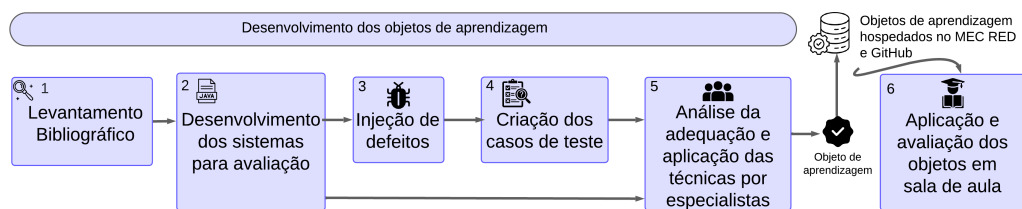
Em seu estudo, [Almeida et al. 2025] propõem apoio aos estudantes durante a interpretação das especificações de problemas, aproximando as práticas formais de teste de software ao ensino de programação; mediante uma abordagem interativa, baseando-se apenas nos dados de entrada e saída. Para a construção de objetos de aprendizagem, observa-se o estudo de [Garrido et al. 2018], em que utilizam artefatos como ferramentas e recursos epistêmicos para nortear a compreensão de conteúdos por meio do *design* instrucional, para cumprir objetivos educacionais articulando entre a forma e a função.

Observando os trabalhos existentes, nota-se a limitação quanto à disponibilidade de objetos de aprendizagem no domínio dos testes caixa preta. A presente pesquisa busca incentivar a aprendizagem de teste de software, por meio do desenvolvimento de objetos de aprendizagem de acesso aberto, visando contribuir para sanar lacunas na formação dos estudantes quanto às atividades de teste, especificamente das abordagens funcionais.

4. Design Instrucional

Este trabalho promove o desenvolvimento de objetos de aprendizagem para apoio ao ensino dos testes caixa preta. A Figura 1 apresenta uma visão geral do *design* instrucional realizado. Os objetos criados foram analisados por três especialistas de áreas relacionadas à Computação e Teste de Software.

Inicialmente, foi realizado um levantamento bibliográfico, analisando trabalhos e pesquisas que abordassem conceitos educacionais de teste de software e, especialmente, dos testes funcionais. Posteriormente, foram desenvolvidos sistemas com defeitos propositalmente injetados, para ilustrar a utilização das técnicas de teste caixa preta; utilizando a linguagem de programação Java, considerando situações comumente executadas por indivíduos, quando estão em contato com a tecnologia.

Figura 1. Desenvolvimento dos Objetos de Aprendizagem.

Posto isto, foram gerados repositórios com os sistemas (Seção 7), que têm como propósito a disponibilidade e demonstração dos testes para estudos futuros e pessoas que desejam visualizar os pontos apresentados neste artigo. Nesta etapa, houve a criação dos casos de teste, baseados nos requisitos e especificações dos sistemas selecionados como amostras. Este processo promoveu a exploração quanto à adequação das técnicas, investigando como podem ser implementadas de forma individual ou combinada, então, os sistemas e os casos de testes criados foram analisados por especialistas, que realizaram inspeções nos códigos de teste. Por fim, estes objetos serão aplicados em sala de aula da seguinte forma: os alunos receberão os sistemas com defeitos injetados e precisarão criar casos de testes seguindo os critérios e especificações definidos para cada sistema. Avaliaremos a adoção dos objetos de forma quantitativa (i.e., percentual de defeitos encontrados pelos testes dos alunos), e qualitativamente com formulários sobre a utilidade/usabilidade aplicados à alunos e professor da disciplina de teste de software.

5. Objetos de Aprendizagem

A ilustração da aplicação das técnicas de teste caixa preta em diferentes cenários, descritos nesta seção, acontece sob a investigação de diferentes sistemas, que possuem características e complexidades distintas. Além disso, é importante destacar que defeitos foram propositalmente injetados de modo a permitir a avaliação da aplicação das técnicas.

5.1. Particionamento por Equivalência

Para aplicar a técnica de particionamento por equivalência, é preciso analisar as especificações do sistema para subdividir os casos de teste entre classes válidas, que contemplam os requisitos necessários para utilização, e as inválidas, que violam os parâmetros determinados. Sendo úteis para a investigação de sistemas que possuem regras bem definidas para a execução das suas funcionalidades.

5.1.1. Sistema de Validação de Senhas

Este sistema simula a autenticação de uma senha, que segue as restrições: ter uma quantidade mínima de oito caracteres; incluir letras maiúsculas, minúsculas, dígitos e caracteres especiais. O comportamento esperado consiste na inserção de senhas dentro do limite determinado de caracteres, evitando campos vazios ou entradas excessivamente longas, além de cumprir as exigências de composição de caracteres e dígitos permitidos.

Os seguintes defeitos foram injetados na aplicação: ausência de limites mínimos e máximos para a inserção de caracteres; aceitação indevida de espaços; solicitação da inserção de caracteres especiais, mas limitando seu uso durante a verificação; inconsistências durante a verificação *case-sensitive*; falta de validação dos dados de entrada; não atendimento a aspectos de usabilidade (Tabela 1).

Tabela 1. Validação de Senhas.

Entradas	Classes Válidas	Resultado esperado	Classes Inválidas	Resultado obtido
#Quantidade de caracteres	≥ 8	Aceito	< 8	Negado
#Letras maiúsculas	≥ 1	Aceito	< 1	Negado
#Letras minúsculas	≥ 1	Aceito	< 1	Negado
#Caracteres especiais	≥ 1	Aceito	< 1	Negado
#Dígitos	≥ 1	Aceito	< 1	Negado

Após a aplicação de oito casos de teste, foram detectados os seguintes problemas: não há verificação do limite máximo ou mínimo de caracteres; podem ser inseridos espaços dentro da senha; há uma limitação quanto aos caracteres especiais que podem ser utilizados, atraindo comportamento inesperado na verificação; a confirmação para verificar, ou não, a nova senha, apresenta inconsistências de *case-sensitive*; são aceitas entradas vazias; e não são enviadas mensagens de erro sobre qual regra está divergente.

5.2. Grafo de Causa e Efeito e Tabela de Decisão

Para executar as técnicas de grafo de causa e efeito e tabela de decisão, é necessário que os sistemas possuam funcionalidades codependentemente implementadas, de modo que para cada ação tomada, uma consequência seja obtida. A partir disso, é feito um mapeamento e organização dos casos de teste, estabelecendo interações e conexões lógicas entre as diferentes opções. Assim, pode-se identificar a causa raiz de um problema e as implicações de cada uma das funcionalidades.

5.2.1. Sistema de Reserva de Voos

Neste objeto de aprendizagem, simula-se o ambiente real para a reserva de voos, respeitando a formatação de datas, cálculo de dias de viagem e as localizações de origem e destino. Para mapear as ações e reações que o sistema executa com base nas entradas fornecidas pelos usuários (Tabela 2 e Figura 2), para a identificação de possíveis falhas em interações mais complexas. Neste caso, foram injetados defeitos relacionados a: ausência de validação quanto aos endereços de origem e destino; ausência do tratamento correto para validação das datas; problemas com o cálculo dos dias de viagem; ausência de verificação da inserção de caracteres indevidos; ausência de limite mínimo ou máximo de inserção de caracteres; não atendimento a aspectos de usabilidade.

Essas técnicas permitem estabelecer rotas de teste a serem seguidas, determinando condições e ações que ilustram o fluxo seguido pelo sistema com base nas escolhas de combinações de entradas. Após a execução de 11 casos de teste, a investigação detectou os defeitos: falta de validação dos endereços de origem e destino; inconsistências no tratamento das datas; viagens com ida e volta para o mesmo dia geram valor negativo, havendo problemas no cálculo de dias de viagem; podem ser inseridos quaisquer caracteres sem verificação de conformidade; não há verificação dos limites mínimos e máximos de inserção; e, apresenta problemas de usabilidade, de modo que as mensagens de erro não especificam o problema.

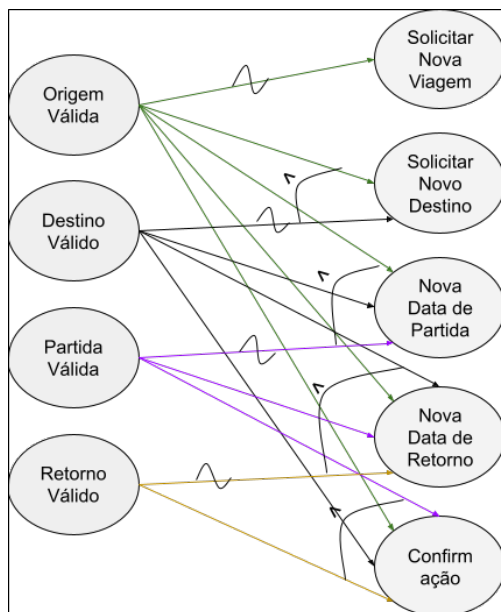


Figura 2. Grafo de Causa e Efeito para Reserva de Voos.

Tabela 2. Tabela de Decisão para Reservas de Voos.

Condições	C1	C2	C3	C4	C5
Origem válida	F	V	V	V	V
Destino válido	F	F	V	V	V
Data de partida válida	-	-	F	V	V
Data de retorno válida	-	-	-	F	V
Ações					
Solicitar nova origem	X				
Solicitar novo destino		X			
Solicitar nova data de partida			X		
Solicitar nova data de retorno				X	
Confirmar reserva					X

5.3. Teste Funcional Sistemático e *Error Guessing*

O teste funcional sistemático é composto pela união das técnicas de particionamento por equivalência e análise de valor limite, e pode ser aplicado a sistemas que possuam em suas especificações diretrizes bem definidas acerca da utilização das suas funcionalidades, isto é, determinando parâmetros para as inserções e que devem seguir os requisitos. Então, divide-se as funções do sistema em casos de teste de classes de equivalência válidas e inválidas, determinando limites mínimos e máximos para as suas extremidades. Adicionalmente, o *error guessing*, carece da *expertise* de quem está testando para identificar ponto sensíveis e mapear defeitos iminentes.

5.3.1. Sistema de Calculadora de Determinantes

Em outra aplicação, este sistema simula o ambiente de uma calculadora de matrizes 3x3, para auxiliar atividades matemáticas (Tabela 3). Os elementos para a criação dos casos de teste são: os valores válidos (positivos, negativos, decimais e inteiros) e inválidos (matrizes de outros tamanhos e entradas não numéricas), aliado à análise das extremidades e pontos críticos, que determina e testa limites de entradas possíveis.

Para esta análise têm-se os defeitos: a fórmula do cálculo de determinantes está incorretamente implementada; a utilização de variáveis do tipo *float* causa inconsistências nos cálculos; permite a inserção de caracteres alfanuméricos, prejudicando o processamento; não há limite mínimo ou máximo de números inseridos.

Este sistema deveria calcular corretamente o determinante da matriz. Mas, durante a aplicação de oito casos de teste, detectaram-se os defeitos: a fórmula de cálculo de matrizes não está implementada corretamente, implicando em erros no resultado final, assim como o uso de variáveis do tipo *float* pode causar imprecisão; podem ser inseridos

Tabela 3. Cálculos de Matrizes.

Classes de Equivalência	Entradas	Análise de Valor Limite	Saídas
C01: Números naturais maiores que zero	{{1, 2, 3}, {4, 5, 6}, {7, 8, 9}}	Limite em modelo ideal	Válida
C02: Números naturais e inserções vazias	{{1, 2, 3}, {4, 5}}	Valor da matriz incompleto	Inválida
C03: Números naturais e caractere alfanumérico	{{1, 2, "a"}, {4, 5, 6}, {7, 8, 9}}	Entradas não numéricas	Inválida
C04: Números naturais iguais a zero	{{0, 0, 0}, {0, 0, 0}, {0, 0, 0}}	Limite com determinante igual a zero	Válida
C05: Números naturais maiores que zero, excedendo os campos	{{1.5, 2, 3}, {4, 5.5, 5}, {7, 8, 5.5}}	Valores decimais positivos	Inválida
C06: Números inteiros e decimais	{{-1, -2.5, -3}, {-4.5, -5, -6}, {-7, -8, -9.5}}	Valores decimais negativos	Válida
C07: Números naturais e inteiros	{{1, 0, -1}, {0, 1, -1}, {-1, 0, 1}}	Valores positivos, negativos e zero	Válida

caracteres alfanuméricos; não há limite na quantidade de números inseridos, variando entre números negativos e infinitos, impactando no cálculo e processamento do sistema.

5.4. Teste Combinacional e *Error Guessing*

O teste combinacional pode ser aplicado a sistemas que carecem de atenção durante a associação das suas funcionalidades; desse modo, utilizando as especificações definidas como parâmetros para a execução de ações seguidas. Contribuindo para a obtenção da cobertura das combinações possíveis no sistema, representando as múltiplas ações mediante tabelas, que dependem de decisões para seguir diferentes percursos. Somada a *error guessing* para a suposição de vulnerabilidades durante a execução dessas ações.

5.5. Sistema de Empréstimo de Livros

Para este objeto de aprendizagem, simula-se o ambiente de empréstimo de livros por meio de ações exercidas para manusear os livros disponíveis que considera os caminhos e ações possíveis (Tabela 4). Esperando comportamentos como: o preenchimento correto dos dados de um livro, respeitando a quantidade de caracteres inseridos conforme pré-determinado, para identificar erros antes de seguir o fluxo das combinações. Este objeto possui os seguintes defeitos: não há o tratamento de exceções no menu; não há validação quanto aos caracteres inseridos, aceitando até entradas vazias; empréstimos podem ser realizados mesmo com o preenchimento incompleto; não há limite mínimo ou máximo de inserção de caracteres; a busca é *case-sensitive*.

Utilizando essa abordagem, com a aplicação de 9 casos de teste, foram detectados problemas relacionados à falta de tratamento de exceções durante a escolha do menu, não havendo validação de caracteres; são permitidas entradas vazias para alguns campos; podem ser realizados empréstimos com informações incompletas sobre um livro; a busca faz distinção entre letras maiúsculas e minúsculas, assim como não limita a quantidade

Tabela 4. Empréstimo de Livros.

Caso de Teste	Ação 1	Ação 2	Parâmetros
CT01	Registrar livro	Listar livros registrados	Título, autor, número de páginas, ano e localização válidos
CT02	Registrar livro	Emprestar livro	Título vazio, autor e ano válidos
CT03	Emprestar livro	Conferir livros alugados	Título existente e disponível
CT04	Emprestar livro	Emprestar livro	Título existente já emprestado
CT05	Devolver livro	Emprestar livro	Título existente, livro já emprestado
CT06	Devolver livro	Conferir livros alugados	Título não emprestado
CT07	Conferir livros alugados	Listar livros registrados	Nenhum parâmetro
CT08	Listar livros registrados	Sair do sistema	Nenhum parâmetro

máxima e mínima de caracteres inseridos; e, as informações não estão bem armazenadas, havendo perda de dados. Assim, a análise das técnicas de teste caixa preta ilustradas neste trabalho, por meio dos cenários empregados oferece suporte direto às aplicações reais, promovendo o desenvolvimento de novas metodologias de teste que possuem impacto na qualidade e confiabilidade dos sistemas.

6. Considerações Finais

No contexto em que as aplicações estão crescendo cada vez mais, em complexidade e tamanho, a atividade de Teste de Software se torna ainda mais necessária e complexa. Apesar do aumento da importância da atividade de Teste na indústria como forma de encontrar falhas e mensurar a qualidade do software produzido; observamos que as práticas e técnicas de Teste de Software ainda são muitas vezes abordadas de forma incipiente no *currículo* de cursos de graduação em TI [Astigarraga et al. 2010] (i.e., disciplinas eletivas ou tópicos em disciplinas de programação). Este cenário tem levado à carência de profissionais capacitados na área de Teste [Astigarraga et al. 2010].

Neste trabalho, apresentamos objetos de aprendizagem que focam na aplicação das metodologias e estratégias dos testes de caixa preta utilizando sistemas distintos criados para uma aplicação mais didática de cada técnica. Neste trabalho, também ilustramos as execuções de cada técnica visando um melhor entendimento dos testes funcionais como um meio para avaliar a qualidade e confiabilidade de um software.

Este trabalho evidencia a importância da construção de objetos de aprendizagem, que possibilitem o aperfeiçoamento de profissionais e o aumento do desempenho destes nas atividades de teste. Como ações futuras, pretende-se ampliar o *design* instrucional, adicionando os resultados da aplicação destes objetos a estudantes de disciplinas de teste de software, visando validar a sua eficácia no processo de ensino-aprendizagem, sob a perspectiva de usabilidade e contribuições pedagógicas. Objetiva-se investigar o impacto da utilização dos objetos de aprendizagem como apoio à prática dos testes caixa preta, bem como, analisar a sua eficiência para elucidar a implementação das técnicas. O acesso aos objetos de aprendizagem deste trabalho é, portanto, o primeiro passo neste sentido.

7. Disponibilização de Artefatos

Os materiais instrucionais utilizados como apoio para a realização desta pesquisa, estão disponíveis na Plataforma de Recursos Educacionais Digitais (MEC RED): <https://mecred.mec.gov.br/recurso/370234>, e, no repositório *GitHub* de acesso aberto: <https://github.com/clarameira/BBTEST>. Em decorrência da limitação de espaço, além dos objetos de aprendizagem descritos nesta pesquisa, o repositório contém outros três objetos que não puderam ser contemplados nesta versão.

8. Declaração sobre uso de Inteligência Artificial

Em atendimento ao código de conduta da Sociedade Brasileira de Computação (SBC), os autores declaram que não houve o uso de Inteligência Artificial (IA) generativa na escrita e/ou revisão do conteúdo deste artigo.

Referências

- Almeida, A., Guerrero, D., e 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.
- Astigarraga, T., Dow, E. M., Lara, C., Prewitt, R., e Ward, M. R. (2010). The emerging role of software testing in curricula. In *2010 IEEE Transforming Engineering Education: Creating Interdisciplinary Skills for Complex Global Environments*, pages 1–26. IEEE.
- Barbosa, E. F. et al. (2004). *Uma contribuição ao processo de desenvolvimento e modelagem de módulos educacionais*. PhD thesis, PhD thesis, ICMC-USP, Sao Carlos, SP.
- Benitti, F. e Albano, E. (2012). Teste de software: o que e como é ensinado? In *Anais do XX Workshop sobre Educação em Computação*, pages 180–189, Porto Alegre, RS, Brasil. SBC.
- Chrisantyo, L., Wibowo, A., Anggiarini, M. N., e Chrismanto, A. R. (2022). Blackbox testing on the revamp results of the dutatani agricultural information system. In *Proceedings of 11th International Congress on Advanced Applied Informatics*, pages 407–417.
- Delamaro, M., Jino, M., e Maldonado, J. (2016). *Introdução ao teste de software*. Elsevier Brasil, Rio de Janeiro, 2 edition.
- Diniz, L. e Dazzi, R. (2011). Jogo digital para o apoio ao ensino do teste de caixa-preta. In *Anais do X Simpósio Brasileiro de Qualidade de Software*, pages 231–245, Porto Alegre, RS, Brasil. SBC.
- Garrido, F., Rêgo, B., e Matos, E. (2018). Design instrucional orientado a artefatos: uma abordagem participativa e distribuída. In *Anais do XXIX Simpósio Brasileiro de Informática na Educação*, pages 258–267, Porto Alegre, RS, Brasil. SBC.
- Kaner, C., Falk, J., e Nguyen, H. Q. (2011). *Testing Computer Software*. Wiley.

-

Maldonado, J., Barbosa, E., Maldonado, J., e Barbosa, E. (2006). Establishing a mutation testing educational module based on ima-cid. In *Second Workshop on Mutation Analysis (Mutation 2006 - ISSRE Workshops 2006)*, pages 14–14.

Nascimento, E., Coelho, R., Madeira, C., Medeiros, K., Silva, L., e Neto, C. O. (2024). Definição e avaliação de uma abordagem gamificada para o ensino de teste de software. In *Anais do XXXVIII Simpósio Brasileiro de Engenharia de Software*, pages 455–465, Porto Alegre, RS, Brasil. SBC.

Neto, A. C. D. (2022). Introdução a teste de software.

Severo, J. e Lelli, V. (2023). Testing maze: an educational game for teaching functional testing. In *Proceedings of the XXXVII Brazilian Symposium on Software Engineering*, SBES '23, page 407–415, New York, NY, USA. Association for Computing Machinery.