

Uma técnica para apoiar profissionais de testes na condução de testes metamórficos em chatbots

Gabriel Santos³, Williamson Silva², Pedro Henrique Dias Valle^{1,3}

¹ Universidade de São Paulo – USP (IME), São Paulo, SP, Brasil

² Universidade Federal do Pampa (UNIPAMPA) – Alegrete, RS, Brasil

³ Universidade Tecnológica Federal do Paraná (UTFPR) – Cornélio Procópio, PR, Brasil

Gabriel.Santos.BCC@gmail.com,

williamson.silva@gmail.com, pedrohenriquevalle@usp.br

Abstract. *Identifying software defects is a constant challenge in Software Engineering, especially for less experienced testers. The lack of practical and accessible tools to guide testing practice can compromise software quality and the efficiency of testing processes. This study proposes using a guide based on metamorphic relationships to help identify defects in chatbots. The guide can simplify testing processes and increase efficiency in defect detection. In addition, the guide is a valuable tool for improving the practice of software testing, especially for beginners.*

Resumo. *Identificar defeitos de software é um desafio constante na Engenharia de Software, principalmente para testadores menos experientes. A falta de ferramentas práticas e acessíveis para orientar a prática de testes pode comprometer a qualidade do software e a eficiência dos processos de teste. Este estudo propõe a utilização de um guia baseado em relações metamórficas para auxiliar na identificação de defeitos em chatbots. O guia pode simplificar os processos de teste e aumentar a eficiência na detecção de defeitos. Além disso, o guia é uma ferramenta valiosa para melhorar a prática de testes de software, principalmente para iniciantes.*

1. Introdução

Nos últimos anos, percebeu-se um aumento significativo no interesse por agentes conversacionais, conhecidos como chatbots, principalmente durante a pandemia de COVID-19, quando o distanciamento social e a necessidade por ferramentas computacionais cresceram exponencialmente [Deepaisarn et al. 2022]. Diante da crescente utilização desses agentes conversacionais em diversas áreas, como atendimento ao cliente, suporte técnico e até mesmo como assistentes virtuais pessoais, observou-se uma necessidade urgente de verificar a qualidade dos chatbots antes deles serem disponibilizados aos usuários finais [Santos et al. 2020, Deepaisarn et al. 2022].

Estudos sobre chatbots têm sido conduzidos desde a década de 1960. O objetivo inicial era simular interações dos chatbots com humanos, fazendo-os passar por humanos reais. No entanto, esses estudos não se limitaram a simular a conversa humana e entreter os usuários [Shawar and Atwell 2007]. Alguns estudos se concentravam

em explorar a capacidade desses sistemas para realizar tarefas específicas, como responder a perguntas ou fornecer informações úteis [Shawar and Atwell 2007]. Por exemplo, alguns chatbots foram projetados para auxiliar na resolução de problemas técnicos, responder a consultas sobre dados específicos ou até mesmo desempenhar funções educacionais, fornecendo informações e suporte em determinados domínios de conhecimento [Shawar and Atwell 2007].

Dentre os principais domínios que utilizam chatbots, destacam-se: educação, recuperação de informações, negócios e comércio eletrônico [Shawar and Atwell 2007]. Além do setor de atendimento ao cliente, eles podem ser encontrados em aplicativos de mensagens, sites e até mesmo em dispositivos de assistência pessoal, como o *Amazon Echo* (Alexa), a Siri (Apple), a Cortana (Microsoft) e o Assistant (Google) [Guerreiro and Barros 2019]. Os chatbots também são conhecidos por *bots* inteligentes, agentes interativos, assistentes digitais e entidades artificiais de conversação [Adamopoulou and Moussiades 2020]. *Bots* são programas automatizados que executam uma variedade de tarefas repetitivas sem interação humana, enquanto chatbots são uma subcategoria de *bots* especificamente projetados para interagir com humanos através de conversas, simulando a comunicação humana por meio de texto ou voz [Griol et al. 2013].

Portanto, conduzir atividades de teste podem contribuir para avaliar a capacidade de um chatbot atender às necessidades dos usuários, analisando se eles conseguem compreender e responder adequadamente às solicitações, bem como verificar possíveis falhas e limitações em seu funcionamento [Santos et al. 2020]. Além disso, por se tratar de uma tecnologia em constante evolução, o teste de software é essencial para aprimorar a qualidade dos chatbots, garantindo sua eficiência e eficácia no apoio dos usuários finais [Okanović et al. 2020].

A atividade de teste de software aplicada no contexto de *chatbots* refere-se ao processo de avaliar e validar o desempenho, a eficácia e a qualidade de um chatbot [Abdul-Kader and Woods 2015]. Portanto, essa atividade está associada a um conjunto de técnicas, critérios e ferramentas para avaliar o comportamento do chatbot em diferentes situações, garantindo que ele funcione corretamente, atenda às necessidades dos usuários e forneça respostas precisas e relevantes [Abdul-Kader and Woods 2015, Bozic et al. 2019]. O teste em chatbots pode envolver a análise de diferentes aspectos, como a compreensão da linguagem natural, a capacidade de resposta adequada às consultas dos usuários, a identificação de intenções corretas, a correção de erros e a consistência na interação [Abdul-Kader and Woods 2015].

Nesse sentido, o presente estudo visa propor um guia que ofereça diretrizes claras e práticas para a realização dos testes em chatbots, ajudando profissionais da indústria na condução dessa atividade de forma eficiente. O restante do trabalho está organizado da seguinte forma: na Seção 2 são apresentados os principais conceitos para o entendimento do trabalho. Na Seção 3 o guia proposto é descrito, bem como um conjunto de diretrizes para apoiar sua aplicação. Na Seção 4 são apresentadas as considerações finais.

2. Referencial Teórico

Essa seção apresenta os principais conceitos necessário para o entendimento desse trabalho. Em particular, as definições de teste de software e chatbots são apresentadas.

2.1. Teste de Software

O teste de software pode ser dividido em diferentes técnicas, como o teste funcional e o teste estrutural. O teste funcional, conforme descrito por Delamaro et al (2016), concentra-se na verificação do comportamento funcional de um sistema ou componente, garantindo que ele funcione conforme o esperado e atenda aos requisitos do sistema. Por outro lado, o teste estrutural, também conhecido como teste de caixa-branca, analisa a estrutura interna do programa, examinando seu código-fonte, fluxo de controle e caminhos de execução para identificar possíveis erros lógicos e vulnerabilidades.

O teste metamórfico é uma abordagem relevante para identificar defeitos, conforme relatado na literatura. Segundo Silva (2022), essa técnica transcende as limitações dos métodos convencionais ao explorar as transformações observadas nas saídas quando as entradas sofrem alterações específicas. Essa abordagem é particularmente útil para validar a corretude e a robustez dos *chatbots* em cenários onde a determinação de respostas corretas pode ser ambígua ou inatingível.

Diversas ferramentas de teste de software têm sido utilizadas no mercado para apoiar as atividades de teste em *chatbots*. Ferramentas como Selenium, TestComplete e Cypress são amplamente empregadas para automatizar testes em diferentes plataformas, desde aplicações web até dispositivos móveis [Valle et al. 2015]. Essas ferramentas oferecem funcionalidades que permitem realizar testes de regressão, testes end-to-end (E2E) e testes baseados em dados, facilitando a detecção de defeitos e a melhoria contínua do software. Ferramentas como Ranorex Studio e Telerik Test Studio também são mencionadas por oferecerem funcionalidades adicionais, como teste de *performance* e teste exploratório, que são essenciais para garantir a qualidade de software em diferentes cenários [UDS 2022].

2.2. Chatbots

Espindola and Cordenonzi (2002) ressaltam que a qualidade dos chatbots é um fator crucial para o sucesso desses sistemas, especialmente em termos de satisfação do usuário e retenção de clientes. Estudos recentes, como os de Padmanabhan (2019), têm investigado métodos para aprimorar a precisão e a eficiência dos chatbots, propondo diretrizes e boas práticas para a realização de testes abrangentes que assegurem a confiabilidade e a eficácia desses agentes conversacionais.

O teste de software é uma atividade crítica que tem como objetivo identificar defeitos no sistema em teste, eliminando falhas por meio da depuração de código. Estudos como o proposto por Santos et al. (2020), destacam a importância de realizar testes abrangentes em *chatbots* para garantir que eles compreendam e respondam adequadamente às solicitações dos usuários, evitando respostas incorretas ou inadequadas que possam prejudicar a experiência do usuário e comprometer a confiabilidade do sistema.

Apesar das iniciativas para obtenção da qualidade de *chatbots*, ainda existem desafios significativos a serem superados. A natureza dinâmica e variável das interações em linguagem natural, a necessidade de tratar ambiguidades e nuances e a preocupação com a ética e a reprodução de preconceitos nos dados de treinamento são alguns dos obstáculos que os pesquisadores e profissionais da área enfrentam [Ramesh et al. 2017]. Estudos futuros precisam continuar explorando novas abordagens e refinando as práticas existentes

para garantir que os *chatbots* possam oferecer interações mais precisas, eficientes e eticamente responsáveis.

Os estudos destacam a importância de adotar abordagens sistemáticas e eficazes para testar *chatbots*, garantindo que esses produtos de software possam atender às expectativas dos usuários e oferecer uma experiência de alta qualidade [AbuShawar and Atwell 2015]. A pesquisa proposta neste trabalho visa contribuir para esse campo, oferecendo um guia que ajude profissionais de teste a realizarem testes metamórficos a fim de garantir seu funcionamento e qualidade.

3. Um guia ao teste metamórfico de chatbots

Essa seção apresenta o *Tesutā* que é um guia de apoio a condução de testes metamórficos em *chatbots*. Em particular, as próximas subseções apresentam a metodologia seguida para a construção do *Tesutā*, a descrição do guia e as heurísticas de apoio ao uso do *Tesutā*

3.1. Metodologia

Foi realizado um Mapeamento Sistemático da Literatura (MSL) para compreender o estado da arte em que esse estudo está inserido [Santos et al. 2024]. Desta forma, foi possível observar que os testes metamórficos podem ser aplicados na atividade de teste em *chatbots*, uma vez que este tipo de aplicação pode ser difícil de testar devido a complexidade da natureza de interação deste tipo de software.

No intuito de alcançar o objetivo do estudo, após a análise do MSL para identificar as principais técnicas, critérios e ferramentas para realizar a atividade de teste em *chatbots*. Realizou-se um estudo sobre testes metamórficos. A partir das evidências coletadas no passo anterior, observou-se que o teste metamórfico é uma boa solução para o contexto de *chatbots*. Portanto, uma investigação aprofundada nesse tópico foi realizada.

A partir das evidências coletadas, desenvolveu-se o *Tesutā*¹ que é um guia para apoiar os profissionais de teste a executar testes metamórficos em *chatbots*. Esse guia é um passo a passo de apoio aos profissionais de teste, contemplando desde a compreensão do contexto dos *chatbots* até a avaliação dos testes realizados.

3.2. Descrição do *Tesutā*

O guia denominado *Tesutā* foi desenvolvido pensando no fluxo geral da condução de testes em *chatbots*. Sendo assim, o guia foi dividido em etapas e, em cada etapa são disponibilizadas heurísticas que servirão de base para os testadores.

O fluxo de execução desse guia pode ser visualizado na Figura 1. A figura possui três tipos de atividades e elas se repetem podendo ser opcionais. São exemplos as atividades do processo (atividades que envolvem análise), atividades de execução (atividades que envolvem ações como selecionar e executar) e atividades de inspeção (atividades que envolvem análise e tomada de decisão, exemplo caso os metamorfismos selecionados não contemple o caso de teste almejado, é necessário retroceder à fase de compreensão do *chatbots*).

¹Tesutā: Testador

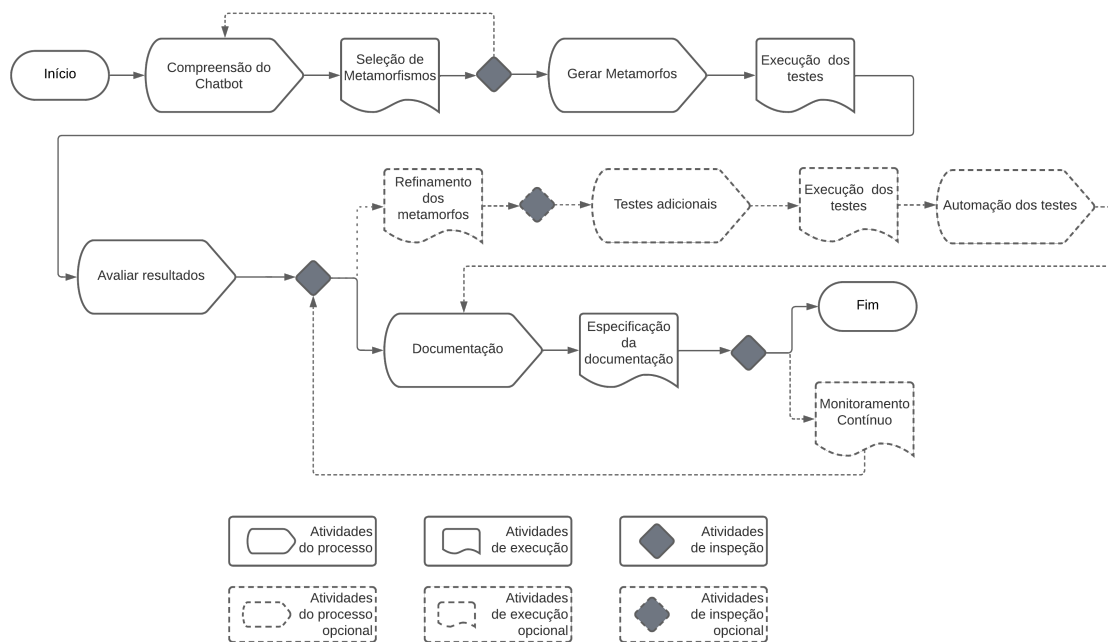


Figure 1. Fluxo do Tesutã

No primeiro passo, o testador realizará a atividade do processo **compreensão do chatbot**, isto é, analisar as funcionalidades, identificar comportamentos esperados e listar respostas típicas para diferentes tipos de perguntas. Em seguida, na atividade de execução, será necessário realizar a **Seleção de Metamorfismos**, ou seja, identificar quais são os metamorfos relevantes para o contexto do *chatbot*. Nesta etapa, é necessário realizar a atividade de inspeção e entender se os metamorfos relevantes foram selecionados ou se é necessário retroceder à atividade de compreensão do *chatbot*. Após a escolha o testador deverá gerar os conjuntos na atividade de processo **Gerar Metamorfos**, modificando sistematicamente as entradas e incluir variações simulando diferentes interações.

Depois de seguir esses passos iniciais, o testador deverá realizar a atividade de **execução dos testes**, introduzindo os metamorfos no *chatbot*. Durante essa fase, é fundamental observar atentamente as respostas geradas para cada conjunto de metamorfos e registrar qualquer desvio ou comportamento inesperado.

A atividade de processo **avaliar resultados** é o próximo passo crítico. O testador deve comparar as respostas obtidas com as expectativas definidas anteriormente durante a análise dos comportamentos esperados. Caso haja variações que o *Chatbot* não lida adequadamente, o testador deve ajustar os metamorfos conforme necessário, refinando o conjunto de testes. Para garantir a abrangência dos testes, é recomendável repetir o processo com conjuntos de metamorfos diferentes. Isso contribuirá para validar a consistência do *Chatbot* diante de diversas entradas modificadas.

Considerando a eficiência e a necessidade de testes frequentes, o testador pode explorar a automação dos testes, utilizando ferramentas disponíveis no mercado. Essa abordagem automatizada facilitará a execução de testes repetitivos, identificando rapidamente possíveis problemas e garantindo a estabilidade do *Chatbot* ao longo do tempo. Além disso, é crucial implementar um sistema de monitoramento contínuo para o *Chatbot*, per-

mitindo a detecção rápida de problemas após atualizações ou alterações no sistema.

A documentação dos metamorfos utilizados, dos resultados obtidos e dos problemas identificados é fundamental para a comunicação efetiva com a equipe de desenvolvimento. Essa documentação servirá como um artefato valioso para futuros testes e iterações, promovendo uma abordagem de melhoria contínua.

3.3. Heurísticas para aplicação do *Tesutã*

Para utilizar o *Tesutã* de forma eficiente é fundamental, recomenda-se utilizar um conjunto de heurísticas desenvolvidas para oferecer suporte aos passos planejados. As heurísticas foram baseadas em conhecimentos especializados e também experiência prática. Elas servirão como diretrizes úteis para os testadores durante a execução das diferentes etapas do processo de teste. As heurísticas ajudarão a orientar a análise do *chatbot*, a seleção e aplicação dos metamorfos, a execução dos testes, a avaliação dos resultados e a tomada de decisões para refinamento e melhoria contínua. Ao empregar heurísticas específicas, os utilizadores poderão abordar de forma mais eficaz os desafios e complexidades inerentes ao teste de *Chatbot*, garantindo uma abordagem mais abrangente e precisa na avaliação de sua funcionalidade e desempenho. Portanto, na próxima seção são apresentadas as heurísticas de apoio a utilização do *Tesutã*.

- **Compreensão do *chatbot***

1. H_1 - Verificar a documentação de requisitos do *chatbot* e analisar quais são os requisitos funcionais e não funcionais a serem validados. Se necessário, levantar mais informações com o cliente/público alvo;
2. H_2 - Identificar em qual domínio o *Chatbot* enquadra-se;
3. H_3 - Identificar quais são os comportamentos esperados do *Chatbot*;

- **Seleção de metamorfismos**

1. H_4 - Identificar, a partir dos requisitos levantados, quais são elegíveis a sofrerem metamorfismo, isto é, quais são passíveis de receber mudança no *input*;
2. H_5 - Garantir que as funcionalidades principais do *Chatbot* sejam selecionadas;

- **Geração dos Metamorfos**

1. H_6 - A partir das funcionalidades selecionadas, crie casos de teste que refletem os metamorfismos selecionados. Por exemplo: testar diferentes cumprimentos de entrada; variações de ortografia; perguntas formuladas de maneiras diferentes; inserir caracteres que não correspondem necessariamente a ação esperada; adição de variações linguísticas, etc;

- **Execução dos testes**

1. H_7 - Assegurar que os metamorfos identificados foram incluídos no caso de teste;
2. H_8 - Registrar as respostas geradas para cada conjunto de metamorfos;
3. H_9 - Registrar desvio de comportamento, caso haja;

- **Avaliar resultados**

1. H_{10} - Comparar os resultados obtidos com as expectativas definidas;

- **Refinamento dos metamorfos**

1. H_{11} - A partir dos resultados obtidos, ajuste os metamorfos caso encontre variações que o *Chatbot* não lide corretamente, ou seja, faça mais testes no problema encontrado para garantir que não haja mais situações semelhantes;
- **Execução de Testes adicionais**
 1. H_{12} - Considere um conjunto de metamorfos diferentes dos iniciais nesta etapa;
 - **Execução dos testes**
 1. H_{13} - Após o refinamento dos metamorfos e a seleção de um novo conjunto, execute novamente os testes obedecendo as heurísticas H_7 , H_8 e H_9 ;
 - **Automação dos testes**
 1. H_{14} - Considere utilizar a automação de testes para aumentar a eficiência do processo, analise se o *Chatbot* em questão pode ser automatizado com as ferramentas disponíveis no mercado;
 - **Documentação**
 1. H_{15} - Documentar os metamorfos utilizados em cada caso de teste;
 2. H_{16} - Documentar os resultados obtidos e quais problemas foram identificados²;
 - **Especificação da documentação**
 1. H_{17} - Faça uma análise dos resultados obtidos e problemas encontrados para aprimorar o *Chatbot*;
 - **Monitoramento contínuo**
 1. H_{18} - Implemente um sistema/processo de monitoramento contínuo, ou seja, para cada *feature*³/*bug fix*⁴ seja realizado uma nova verificação do caso de teste;

A partir destas heurísticas, o *Tesutã* oferece uma estrutura para orientar os testadores durante o processo de teste em *chatbots* usando relações metamórficas. Além disso, o *Tesutã* destaca a importância de compreender completamente o *Chatbot*, identificar comportamentos esperados e listar as respostas típicas como uma base essencial para testes bem sucedidos.

4. Considerações Finais e Trabalhos Futuros

A pesquisa forneceu *insights* valiosos sobre como utilizar o *Tesutã* para identificar defeitos por meio das relações metamórficas. O guia pode ser considerado uma ferramenta eficaz para simplificar processos e aumentar a eficiência na identificação de defeitos, especialmente entre indivíduos com menor experiência em testes de software.

Em resumo, o *Tesutã* pode ser considerado um artefato valioso às práticas de teste de software, com potencial para beneficiar uma ampla gama de profissionais na área. Futuras pesquisas podem focar em refinamentos do *Tesutã* e em avaliar seu impacto a longo prazo em diferentes contextos de desenvolvimento de software.

²A forma de documentação pode variar de empresa para empresa

³Feature: Funcionalidade ou recursos novos adicionados ao *software*

⁴Bug fix: Ajuste de um problema/defeito em um *software*

Agradecimentos

Os autores agradecem a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código Financeiro 001 e a Universidade Tecnológica Federal do Paraná (UTFPR - Cornélio Procópio) pelo apoio. Os autores também agradecem pelo apoio financeiro da FAPERGS (Projeto ARD/ARC – processo 22/2551-0000606-0) e a FAPEMIG (Processo APQ-00743-22).

References

- Abdul-Kader, S. A. and Woods, J. C. (2015). Survey on chatbot design techniques in speech conversation systems. *International Journal of Advanced Computer Science and Applications*, 6(7).
- AbuShawar, B. and Atwell, E. (2015). Alice chatbot: Trials and outputs. *Computación y sistemas*, 19(4):625–632.
- Adamopoulou, E. and Moussiades, L. (2020). Chatbots: History, technology, and applications. *Machine Learning with Applications*, 2:100006.
- Bozic, J., Tazl, O. A., and Wotawa, F. (2019). Chatbot testing using ai planning. In *2019 IEEE International Conference On Artificial Intelligence Testing (AITest)*, pages 37–44. IEEE.
- Deepaisarn, S., uma Imkome, E., Wongpatikaseree, K., Yuenyong, S., Lakanavidid, P., Soonthornchaiva, R., Yomaboot, P., Angkoonawaengsuk, A., and Munpansa, N. (2022). Validation of a thai artificial chatmate designed for cheering up the elderly during the covid-19 pandemic. *F1000Research*, 11:1411.
- Delamaro, M., Maldonado, J., and Jino, M. (2016). *Introdução ao teste de software (2 edição ed.)*. Elsevier, Rio de Janeiro.
- Espindola, M. G. and Cordenonzi, W. (2002). Qualidade em pacotes de software. *Disciplinarum Scientia Naturais e Tecnológicas*, 3(1):29–43.
- Griol, D., Carbó, J., and Molina, J. M. (2013). An automatic dialog simulation technique to develop and evaluate interactive conversational agents. *Applied Artificial Intelligence*, 27(9):759–780.
- Guerreiro, A. and Barros, D. M. V. (2019). Novos desafios da educação a distância: programação e uso de chatbots.
- Okanović, D., Beck, S., Merz, L., Zorn, C., Merino, L., Hoorn, A. V., and Beck, F. (2020). Can a chatbot support software engineers with load testing? approach and experiences. pages 120–129. Association for Computing Machinery, Inc.
- Padmanabhan, M. (2019). Sustainable test path generation for chatbots using customized response. *International Journal of Engineering and Advanced Technology*, 8:149–155.
- Ramesh, K., Ravishankaran, S., Joshi, A., and Chandrasekaran, K. (2017). A survey of design techniques for conversational agents. In *International conference on information, communication and computing technology*, pages 336–350. Springer.
- Santos, G., Silva, W., and Valle, P. (2024). O que sabemos sobre testes em chatbots? uma revisão sistemática da literatura. In *Anais do IX Workshop sobre Aspectos Sociais, Humanos e Econômicos de Software*, pages 106–117, Porto Alegre, RS, Brasil. SBC.

- Santos, M. B. D., Furtado, A. P. C., Nogueira, S. C., and Moreira, D. D. (2020). Ogybug: A test automation tool in chatbots. pages 79–87. Association for Computing Machinery.
- Shawar, B. A. and Atwell, E. (2007). Chatbots: are they really useful? *Journal for Language Technology and Computational Linguistics*, 22(1):29–49.
- Silva, W. d. L. (2022). Uso de testes metamórficos para verificação de aplicação chatbot. Master's thesis, Universidade Federal do Rio Grande do Norte.
- UDS (2022). Ferramentas para teste automatizado de software. <https://uds.com.br/blog/ferramentas-para-teste-automatizado-de-software>. Acessado em 19 de Junho de 2023.
- Valle, P., Barbosa, E. F., and Maldonado, J. (2015). Um mapeamento sistemático sobre ensino de teste de software. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 26, page 71.