



UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

GABRIEL ALVES DOS SANTOS

**TESUTÃ: UM GUIA PARA APOIAR A CONDUÇÃO DE TESTES METAMÓRFICOS
EM CHATBOTS**

DISSERTAÇÃO DE MESTRADO

CORNÉLIO PROCÓPIO
2025

GABRIEL ALVES DOS SANTOS

TESUTÃ: UM GUIA PARA APOIAR A CONDUÇÃO DE TESTES METAMÓRFICOS EM CHATBOTS

**Tesutã: A guide to support conducting metamorphic testing in
chatbots**

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Mestre em Informática.

Orientador: Prof. Dr. Pedro Henrique Dias Valle

Co-orientador: Prof. Dr. Williamson Alison Freitas Silva

CORNÉLIO PROCÓPIO
2025



[4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Esta licença permite remixe, adaptação e criação a partir do trabalho, para fins não comerciais, desde que sejam atribuídos créditos ao(s) autor(es) e que licenciem as novas criações sob termos idênticos. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.



GABRIEL ALVES DOS SANTOS

TESUTÃ: UM GUIA PARA APOIAR A CONDUÇÃO DE TESTES METAMÓRFICOS EM CHATBOTS

Trabalho de pesquisa de mestrado apresentado como requisito para obtenção do título de Mestre Em Informática da Universidade Tecnológica Federal do Paraná (UTFPR).
Área de concentração: Computação Aplicada.

Data de aprovação: 19 de Dezembro de 2024

Dr. Pedro Henrique Dias Valle, Doutorado - Universidade Federal de Juiz de Fora (Ufjf)

Dra. Ana Carolina Oran Rocha, Doutorado - Universidade Federal do Amazonas (Ufam)

Dr. Cleber Gimenez Correa, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Ricardo Ferreira Vilela, Doutorado - Universidade Federal do Cariri

Dr. Williamson Alison Freitas Silva, Doutorado - Fundação Universidade Federal do Pampa - Unipampa (Unipampa)

Documento gerado pelo Sistema Acadêmico da UTFPR a partir dos dados da Ata de Defesa em 19/12/2024.

AGRADECIMENTOS

Agradeço, primeiramente a Deus por sempre me dar forças e sabedoria para alcançar meus objetivos e vencer os obstáculos. Agradeço a minha noiva Millena Fraga Santana pelo apoio incondicional durante os últimos anos. Agradeço aos meus pais Adriana Cristina da Silva Santos e Jair Alves dos Santos pela base sólida e pelas oportunidades, pois sabemos que a oportunidade só aparece para quem a merece. Agradeço à equipe do PPGI que me ajudou evoluir durante o último ano; ao Prof. Dr. Pedro Henrique Dias Valle por me auxiliar neste período e pela confiança neste trabalho. Da mesma forma, agradeço ao Prof. Dr. Williamson Alison Freitas Silva pelo auxílio inestimável neste projeto. Agradeço aos professores e tutores que me ensinaram tudo que sei e me ajudaram a chegar até aqui, em especial à Profa. Dra. Ana Paula Freitas Vilela Boaventura por me auxiliar em toda a minha carreira acadêmica. E por fim e não menos importante a todos que contribuíram de forma direta ou indireta para que este trabalho fosse realizado.

E aos meus filhos.. vai que...

"A determinação é o que nos mantém avançando, mesmo quando todos os outros desistem.- Vegeta, o Príncipe dos Saiyajins.

RESUMO

DOS SANTOS, Gabriel alves. TESUTÃ: UM GUIA PARA APOIAR A CONDUÇÃO DE TESTES METAMÓRFICOS EM CHATBOTS. 137 f. Dissertação – Programa de Pós-Graduação em Informática, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2025.

Contexto: com o aumento no uso de agentes conversacionais (chatbots), os usuários podem interagir com máquinas por meio da linguagem natural. Sendo assim, surge a necessidade de desenvolver um guia que apoie as atividades de teste nesse domínio de aplicação. Além disso, os testes metamórficos têm emergido como uma abordagem eficaz para tratar os desafios únicos desse domínio, oferecendo formas automatizadas de identificar falhas em agentes conversacionais. **Problema:** o aumento progressivo na demanda por chatbots tem gerado dilemas de natureza complexa quanto ao design e implementação desses agentes conversacionais, assim como a condução de testes nesse domínio de aplicação. **Objetivo:** propor um guia que ofereça diretrizes claras e práticas para a realização de testes em chatbots, ajudando os profissionais a escolherem as melhores ferramentas e métodos para suas necessidades específicas. **Método:** para isso, foi conduzido um mapeamento sistemático para identificar o estado da arte em relação ao teste de *chabots*. Em seguida, foram analisadas características específicas de chatbots que os diferenciam do teste para sistemas tradicionais. Nesse sentido, foi definido um conjunto de diretrizes para apoiar o teste em *chabots*. Para validar a efetividade dessas diretrizes, uma avaliação com estudantes foi conduzida para analisar a completude e a corretude dos testes gerados. Durante o mapeamento sistemático, os testes metamórficos foram identificados como uma técnica relevante devido à sua capacidade de validar casos de teste mesmo na ausência de resultados esperados explícitos. Essa técnica foi incorporada ao guia Tesutã como uma diretriz específica para lidar com os desafios de teste em chatbots, especialmente em cenários que envolvem processamento de linguagem natural. **Resultados:** este trabalho desenvolveu o guia Tesutã, que apresenta uma visão das ferramentas e métodos utilizados para testar chatbots. A avaliação do guia, por meio de experimentos com estudantes, revelou que ele facilita a seleção de abordagens adequadas, aumentando a eficácia dos testes. Os resultados mostraram que o Tesutã contribuiu para a revelação de falhas e melhorias para a qualidade dos chatbots, evidenciando seu potencial para ser uma ferramenta útil em um domínio em constante evolução. **Conclusões:** a avaliação do guia Tesutã demonstrou seu potencial na aplicação de testes metamórficos em chatbots. Os resultados mostraram que, embora o Tesutã tenha alcançado uma completude significativa de 75,70% para os requisitos identificados, a variabilidade nos desempenhos dos participantes indicou que o guia necessita de ajustes para melhorar sua acessibilidade.

Palavras-chave: Teste de Software, Chatbot, Teste Metamórfico

ABSTRACT

DOS SANTOS, Gabriel alves. Tesutã: A guide to support conducting metamorphic testing in chatbots. 137 f. Dissertação – Programa de Pós-Graduação em Informática, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2025.

Context: With the increasing use of conversational agents (chatbots), users can interact with machines through natural language. This rise highlights the need for a guide to support testing activities in this domain. Furthermore, metamorphic testing has emerged as an effective approach to address unique challenges in this area by offering automated ways to identify flaws in conversational agents. **Problem:** The growing demand for chatbots has created complex challenges in designing, implementing, and testing these conversational agents. **Objective:** To propose a guide that provides clear and practical guidelines for testing chatbots, helping professionals select the best tools and methods for their specific needs. **Method:** A systematic mapping study was conducted to identify the state of the art in chatbot testing. Subsequently, specific chatbot characteristics that differentiate their testing from traditional systems were analyzed. Based on this analysis, a set of guidelines was defined to support chatbot testing. To validate the effectiveness of these guidelines, an evaluation with students was conducted to analyze the completeness and correctness of the generated tests. Metamorphic testing was identified as a relevant technique during the systematic mapping due to its ability to validate test cases without explicit expected results. This technique was incorporated into the guide, Tesutã, as a specific guideline to address the challenges of chatbot testing, especially in natural language processing scenarios. **Results:** This work developed the Tesutã guide, which provides an overview of tools and methods for testing chatbots. The evaluation of the guide, through experiments with students, revealed that it facilitates the selection of suitable approaches, increasing testing effectiveness. The results showed that Tesutã contributes to fault detection and improves chatbot quality, highlighting its potential as a valuable tool in an ever-evolving domain. **Final Remarks:** The evaluation of the Tesutã guide demonstrated its potential in applying metamorphic testing to chatbots. While Tesutã achieved a significant completeness rate of 75.70% for the identified requirements, the variability in participants' performance indicated the need for adjustments to improve its accessibility.

Keywords: Software Testing, Chatbot, Metamorphic Test

LISTA DE FIGURAS

FIGURA 1	– Método de Pesquisa	18
FIGURA 2	– Cenário típico da atividade de teste. Adaptado de Delamaro et al. (2016)	24
FIGURA 3	– Fluxo do <i>Tesutã</i>	45
FIGURA 4	– Experiência dos participantes em relação ao desenvolvimento de software	63
FIGURA 5	– Qual sua experiência com Chatbots?	64
FIGURA 6	– Avaliação da importância dos Chatbots (Escala de 0 a 10)	65
FIGURA 7	– Avaliação da contribuição das heurísticas no entendimento do guia (Escala de 0 a 10)	65
FIGURA 8	– Resposta do ChatGTP em relação ao CT_{BASE}	104
FIGURA 9	– Resposta do ChatGTP em relação ao CT_1	105
FIGURA 10	– Resposta do ChatGTP em relação ao CT_2	106
FIGURA 11	– Resposta do ChatGTP em relação ao CT_3	107
FIGURA 12	– Resposta esperada no CT_3	107

LISTA DE TABELAS

TABELA 1	– Questões de Pesquisa (QP)	33
TABELA 2	– Critérios de seleção (inclusão e exclusão) da RSL.	34
TABELA 3	– Estudos primários selecionados	35
TABELA 4	– Resumo das Respostas	36
TABELA 5	– Requisitos funcionais identificados no oráculo.	56
TABELA 6	– Relações metamórficas identificadas no oráculo.	57
TABELA 7	– Resultados da Corretude Total.	70
TABELA 8	– Resultados da Completude Total.	71
TABELA 9	– Requisitos e Relações Metamórficas Únicos Identificados.	72
TABELA 10	– Resultados de Completude por Participante.	74
TABELA 11	– Heurísticas <i>Tesutã</i>	99
TABELA 12	– Agrupamento e classificação das heurísticas	132
TABELA 13	– Requisitos do Chatbot Eliza	133
TABELA 14	– Metamorfismos e Relações Metamórficas	133
TABELA 15	– Entradas e respostas do Chatbot Eliza	134
TABELA 16	– Documentação do caso de teste	135
TABELA 17	– Documentação dos resultados obtidos	136

LISTA DE SIGLAS

ACM	<i>Association for Computing Machinery</i>
AI / IA	Inteligência Artificial
AITest	<i>IEEE International Conference on Artificial Intelligence Testing</i>
API	<i>Application Programming Interface</i>
BERT	<i>Bidirectional Encoder Representations from Transformers</i>
Botest	Ferramenta de teste de chatbots
BotSE	<i>IEEE/ACM International Workshop on Bots in Software Engineering</i>
Botium	Ferramenta de teste de chatbots
BOW	<i>Bag of Words</i>
CBSOft	Congresso Brasileiro de Software
CE	Critério de Exclusão
CI	Critério de Inclusão
CT	Caso de Teste
DDT	<i>Data-driven testing</i>
E2E	<i>End-to-end testing</i>
GQM	<i>Goal Question Metric</i>
GPT	<i>Generative Pre-trained Transformer</i>
ICSCAN	<i>IEEE International Conference on System, Computation, Automation and Ne</i>
ICTSS	<i>International Conference on Testing Software and Systems</i>
IEC	<i>International Electrotechnical Commission</i>
IEEE	<i>Institute of Electrical and Electronic Engineers</i>
IDE	<i>Integrated Development Environment</i>
ISO	<i>International Organization for Standardization</i>
LGPD	Lei Geral de Proteção de Dados
MLP	Modelos de Linguagens Pré-treinados
MRs	<i>Metamorphic Relations</i>
MSL	Mapeamento Sistemático da Literatura
MT	<i>Metamorphic Testing</i>
NLP	<i>Natural Language Processing</i>

LISTA DE SIGLAS

PCV	Problema do Caixeiro Viajante
PDDL	<i>Planning Domain Definition Language</i>
PLN	Processamento de Linguagem Natural
QP	Questão de Pesquisa
RF	Requisito Funcional
RM	Relação Metamórfica
RNF	Requisito Não Funcional
RSL	Revisão Sistemática da Literatura
SBC	Sociedade Brasileira de Computação
T5	<i>Text-To-Text Transfer Transformer</i>
TCLE	Termo de Consentimento Livre e Esclarecido
UFAM	Universidade Federal do Amazonas
UFJF	Universidade Federal de Juiz de Fora
UNIPAMPA	Universidade Federal do Pampa
UTFPR	Universidade Tecnológica Federal do Paraná
UX	<i>User Experience</i>
V&V	Verificação e Validação
VUI	<i>Voice User Interface</i>
WBOTS	<i>Workshop sobre Bots na Engenharia de Software</i>

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Contexto e Motivação	13
1.2	Problema de Pesquisa	15
1.3	Objetivos	17
1.4	Método de Pesquisa	18
1.5	Principais Resultados	19
1.6	Organização do Trabalho	20
2	FUNDAMENTAÇÃO TEÓRICA E TRABALHOS RELACIONADOS	21
2.1	Teste de Software	21
2.1.1	Conceitos Básicos	23
2.1.2	Técnicas e Critérios de Teste	25
2.1.3	Teste Metamórficos	28
2.2	Agentes Conversacionais Baseados em Texto	30
2.2.1	Tipos de agentes conversacionais	31
2.2.1.1	Vantagens e limitações dos agentes conversacionais	31
2.3	Trabalhos Relacionados	33
2.3.1	Execução	34
2.3.2	Resultados	35
2.3.3	Q_1 : Abordagens (técnicas/critérios) utilizadas em teste de <i>chatbots</i>	35
2.3.4	Q_2 : Automatização das atividades de testes em <i>chatbots</i>	36
2.3.5	Q_3 : Nível de teste em <i>chatbots</i>	39
2.3.6	Q_4 : Linguagens de programação utilizadas nos testes	39
2.3.7	Q_5 : Avaliação experimental das abordagens	39
2.3.8	Q_6 : Domínios dos <i>chatbots</i>	41
2.4	Considerações Finais	42
3	TESUTĀ: UM GUIA DE APOIO AO TESTE METAMÓRFICO EM <i>CHATBOTS</i>	43
3.1	Problema de Pesquisa	43
3.2	Materiais e Métodos	44
3.3	Descrição do Tesutā	44
3.4	Heurísticas de apoio à aplicação do <i>Tesutā</i>	47
3.5	Cenário de Uso	49
3.6	Considerações Finais	51
4	AValiação DO TESUTĀ	52
4.1	Planejamento	52
4.1.1	Questões de Pesquisa (QP)	52

4.1.2	Objetivo	53
4.1.3	Seleção dos Participantes	53
4.1.4	Instrumentação	54
4.1.5	Oráculo	55
4.1.5.1	Requisitos Funcionais Identificados	56
4.1.5.2	Relações Metamórficas Identificadas	56
4.1.6	Métricas	57
4.1.6.1	Corretude dos casos de teste identificados pelos participantes	57
4.1.6.2	Compleitude dos requisitos identificados pelos participantes	58
4.1.6.3	Compleitude quanto aos requisitos do oráculo	59
4.1.6.4	Compleitude quanto às relações metamórficas do oráculo	60
4.2	Execução	61
4.3	Análise dos Resultados	62
4.4	Resultados	62
4.4.1	Dados Demográficos	63
4.4.2	RQP_1 : O Tesutã é fácil de entender com relação a sua abrangência, mostrando instruções claras e concisas sob o ponto de vista dos estudantes?	65
4.4.3	QP_2 : O Tesutã apresenta uma boa efetividade para gerar testes metamórficos para chatbots?	66
4.5	Análise da Qualidade dos Casos de Teste	70
4.5.1	Corretude dos casos de teste identificados pelos participantes	70
4.5.1.1	Compleitude dos requisitos identificados pelos estudantes	71
4.5.1.2	Requisitos e Relações Metamórficas Únicos Identificados	72
4.5.1.3	Compleitude quanto aos Requisitos do Oráculo	72
4.5.1.4	Compleitude quanto às Relações Metamórficas do Oráculo	74
4.5.2	Discussão	75
4.6	Ameaças à Validade	77
4.6.1	Validade Interna	77
4.6.2	Validade Externa	77
4.6.3	Validade de Conclusão	78
4.7	Considerações Finais	78
5	CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS	80
5.1	Contribuições	81
5.2	Trabalhos Futuros	82
5.3	Produção Científica	82
5.3.1	Publicações Elaboradas	82
	REFERÊNCIAS	84
	Anexo A – RESUMO DOS ESTUDOS PRIMÁRIOS SELECIONADOS	89
A.1	Resumo dos Estudos Primários Seleccionados	90
A.1.1	OggyBug: A Test Automation Tool in Chatbots	90
A.1.2	Sorry, I don't Understand: Improving Voice User Interface Testing	90
A.1.3	Ontology-based metamorphic testing for chatbots	91

A.1.4 Sustainable Test Path Generation for <i>Chatbots</i> using Customized Response	91
A.1.5 Algorithm Inspection for Chatbot Performance Evaluation	92
A.1.6 BoTest: A framework to test the quality of conversational agents using divergent input examples	93
A.1.7 Bottester: testing conversational systems with simulated users	93
A.1.8 Chatbot and bullyfree chat	94
A.1.9 Chatbot Interaction with Artificial Intelligence: human data augmentation with T5 and language transformer ensemble for text classification	94
A.1.10 Chatbot testing using AI planning	94
A.1.11 Testing challenges for NLP-intensive bots	95
A.1.12 Testing Chatbots Using Metamorphic Relations	95
A.1.13 Testing chatbots with Charm	96
Anexo B – EXEMPLO DE EXECUÇÃO DAS HEURÍSTICAS COM O CHATGPT	97
B.1 Exemplo de execução das heurísticas com o ChatGPT	98
Anexo C – AULA SOBRE TESTES METAMÓRFICOS	108
Anexo D – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO - TCLE	118
Anexo E – QUESTIONÁRIO DE CARACTERIZAÇÃO E FEEDBACK	120
Anexo F – EXECUÇÃO DO TESUTÃ COM O CHATBOT ELISA	129
F.1 Execução do Tesutã com o chatbot Elisa	130

1 INTRODUÇÃO

1.1 CONTEXTO E MOTIVAÇÃO

Agentes conversacionais são programas de computador que utilizam Inteligência Artificial (IA) e Processamento de Linguagem Natural (PLN) para interagir com usuários por meio de conversas simuladas (WALLACE, 1995; GUERREIRO; BARROS, 2019; SOUZA, 2022). Esses agentes são capazes de compreender e responder às perguntas dos usuários dentro de um contexto relevante, simulando uma conversa como um ser humano (VELÁSQUEZ, 2023).

Em particular, os *chatbots* são considerados um tipo de agente conversacional que pode receber *inputs* via texto ou voz (MORAES; SOUZA, 2015; NUNES, 2012). Um exemplo de *chatbot* que ficou conhecido em 2023 é o ChatGPT¹ que é mantido pela OpenAI. O ChatGPT utiliza Modelos de Linguagens Pré-treinados (MLP) otimizados para diálogo e interage com milhões de usuários de diversas áreas, respondendo desde perguntas mais simples até questões complexas, como desafios de matemática (ZHANG et al., 2022).

É importante destacar que estudos sobre *chatbots* têm sido conduzidos desde a década de 1960 (SHAWAR; ATWELL, 2007). O objetivo inicial era simular interações dos *chatbots* com os humanos para enganá-los, fazendo-se passar por humanos reais. No entanto, esses estudos não se limitavam apenas a simular a conversa humana e entreter os usuários (SHAWAR; ATWELL, 2007), mas também se concentravam em explorar a capacidade desses sistemas para realizar tarefas específicas, como responder a perguntas ou fornecer informações úteis. Por exemplo, alguns *chatbots* foram projetados para auxiliar na resolução de problemas técnicos, responder a

¹Disponível em: <https://chat.openai.com/>

consultas sobre dados específicos ou até mesmo desempenhar funções educacionais, fornecendo informações e suporte em determinados domínios de conhecimento (SHAWAR; ATWELL, 2007; ADAMOPOULOU; MOUSSIADES, 2020b).

Dentre os principais domínios de aplicações de *chatbots*, destacam-se (SHAWAR; ATWELL, 2007): educação, saúde, negócios e comércio eletrônico. Além do setor de atendimento ao cliente, eles podem também 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; BARROS, 2019). Vale a pena mencionar que os *chatbots* também são conhecidos por *bots* inteligentes, agentes interativos, assistentes digitais e entidades artificiais de conversação (ADAMOPOULOU; MOUSSIADES, 2020a).

Assim como nos sistemas de software tradicionais, a qualidade dos *chatbots* é algo essencial para seu sucesso (ESPINDOLA; CORDENONZI, 2002). Em particular, a indústria de software tem exigido produtos com alta qualidade, os quais devem atender às necessidades requeridas pelos usuários (CAMATTI et al., 2023; SILVA, 2016). Os produtos de software que possuem alta qualidade podem aumentar a satisfação dos clientes, o que melhora a retenção e aumenta a confiabilidade no produto (TONTINI; ZANCHETT, 2010). De acordo com Delamaro et al. (2016), o Teste de software é uma das atividades essenciais de Verificação e Validação (V&V), as quais podem ser utilizadas para avaliar a qualidade desse tipo de software.

Nessa perspectiva, a Verificação envolve a revisão e análise do software para garantir que ele atenda aos requisitos estabelecidos e aos padrões de qualidade definidos. Isso pode ser feito por meio de técnicas como revisões de código, inspeções e análise estática de código (DELAMARO et al., 2016). A Validação, por outro lado, tem como objetivo avaliar se o software atende às necessidades e expectativas dos usuários finais. Ela envolve a execução de testes funcionais, de desempenho, de usabilidade, entre outros, visando verificar se o software está funcionando corretamente e cumprindo seu propósito (DELAMARO et al., 2016). Ambas as atividades de V&V são complementares e têm como objetivo garantir a qualidade do software, identificando defeitos, falhas e possíveis melhorias. Elas são fundamentais para aumentar a confiabilidade e a eficácia do software em seu ambiente

de produção (DELAMARO et al., 2016).

Em resumo, o teste de software que tem por objetivo identificar defeitos no sistema de software em teste, e conseqüentemente, eliminar os defeitos que originam as falhas, por meio da depuração de código (DELAMARO et al., 2016). Um exemplo de um defeito no código-fonte não percebido pode ser aplicado em produção e gerar uma exceção em tela para o usuário (falha do sistema) (KO et al., 2011).

Testes metamórficos são uma técnica de teste de software que visa verificar a validade de um sistema mesmo quando não há um conjunto explícito de resultados esperados para comparar diretamente as saídas (CHEN et al., 2020). Essa abordagem baseia-se na definição de relações metamórficas, que descrevem como as saídas de um programa devem mudar em resposta a modificações específicas nas entradas. Por meio dessas relações, é possível detectar falhas ao observar inconsistências no comportamento do sistema, mesmo quando os resultados esperados para cada caso de teste individual não são conhecidos (CHEN et al., 2020).

1.2 PROBLEMA DE PESQUISA

É importante ressaltar que percebeu-se um aumento no interesse por chatbots, principalmente, durante a pandemia de COVID-19 em que 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, observa-se uma necessidade de verificar a qualidade dos *chatbots* antes de serem disponibilizados aos usuários finais (SANTOS et al., 2020; DEEPAISARN et al., 2022).

Nesse sentido, a falta de teste em *chatbots* pode resultar em respostas falsamente positivas, ou seja, situações em que o *chatbot* fornece informações incorretas ou inadequadas (SANTOS et al., 2020). Essas respostas podem prejudicar a experiência do usuário e comprometer a confiabilidade da plataforma. Portanto, é crucial implementar testes abrangentes e regulares para garantir a precisão e eficácia das interações do *chatbot* (SANTOS et al., 2020).

Conduzir atividades de teste de software pode contribuir para avaliar a capacidade de um *chatbot* atender às necessidades dos usuários, analisando se eles conseguem compreender e responder adequadamente às solicitações dos usuários, 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 o desempenho do *chatbot*, garantindo sua eficiência e eficácia (OKANOVIĆ et al., 2020).

O teste de *chatbots* envolve múltiplos aspectos críticos para assegurar interações eficazes e confiáveis. Entre esses aspectos estão a compreensão da linguagem natural, a capacidade de resposta adequada às consultas dos usuários, a identificação correta de intenções, a correção de erros e a consistência na interação. Além disso, os testes podem ser realizados em diferentes etapas do desenvolvimento do *chatbot*, desde testes unitários até testes de aceitação. Diversas características influenciam a confiabilidade dos *chatbots*, incluindo compreensão, ambiguidade e ética.

Para sustentar essas afirmações, destacam-se trabalhos acadêmicos recentes que abordam essas questões. Silva (2022) exploram a aplicação de Testes Metamórficos (TM) na verificação de *chatbots*, enfatizando a importância de técnicas de teste que garantam a compreensão correta das solicitações dos usuários e a precisão nas respostas fornecidas. Outro estudo relevante é apresentado por Soares e Silva (2024), que investiga a literatura acadêmica sobre *chatbots* no contexto brasileiro, realizando um mapeamento inicial dos principais objetivos, período de publicação e área de conhecimento.

Além disso, Oliveira (2023) destaca o papel do Processamento de Linguagem Natural (PLN) e do design de Experiência do Usuário (UX) na eficácia dos *chatbots*, abordando questões de compreensão e ambiguidade nas interações. Complementando essas abordagens, Filho (2024) apresenta o framework DUBI como um meio de avaliar automaticamente *chatbots*, cobrindo seus componentes estáticos e interativos, e fornecendo recomendações para aprimorar sua estrutura e desempenho.

Esses estudos reforçam a importância de uma abordagem abrangente no teste de *chatbots*, considerando tanto aspectos técnicos quanto éticos para garantir

interações de alta qualidade e confiabilidade.

Portanto, essas premissas são essenciais e devem ser consideradas durante o teste em *chatbots* para garantir sua funcionalidade e qualidade. Melhorar a qualidade e a confiabilidade de *chatbots* por meio de estratégias que permitam a realização de testes mais eficazes, assegurando interações precisas e relevantes para os usuários. Auxiliar profissionais na realização de testes em *chatbots* para aprimorar sua qualidade e confiabilidade, promovendo interações mais precisas e satisfatórias para os usuários.

Os Testes Metamórficos têm se destacado como uma técnica eficaz para avaliar a qualidade de sistemas de software que lidam com entrada e saída complexas, como *chatbots* e sistemas de inteligência artificial. Essa abordagem permite identificar falhas que não seriam detectadas por testes tradicionais, ao verificar se certas relações esperadas entre entradas e saídas do sistema se mantêm (CHEN et al., 2018). No contexto de *chatbots*, os Testes Metamórficos são particularmente úteis para validar a coerência das respostas e garantir que pequenas variações nas perguntas dos usuários não resultem em respostas inconsistentes ou inadequadas (BOZIC; WOTAWA, 2019). Estudos recentes indicam que essa técnica pode melhorar significativamente a robustez e a confiabilidade de sistemas baseados em aprendizado de máquina (SEGURA et al., 2016).

1.3 OBJETIVOS

Nesse contexto, o objetivo geral deste projeto de mestrado foi propor um guia para apoiar a execução de testes metamórficos para aprimorar a qualidade e a confiabilidade de *chatbots*, resultando em interações mais precisas e relevantes para os usuários. Para alcançar esse objetivo, os seguintes objetivos específicos foram necessários:

- Desenvolver heurísticas para dar apoio a execução de testes metamórficos;
- Estabelecer um guia para apoiar a condução de teste de *chatbots*;
- Avaliar a eficácia das heurísticas e do Tesutã proposto em cenários reais;

1.4 MÉTODO DE PESQUISA

Para desenvolver um conjunto de diretrizes de apoio ao teste de *chatbots*, um conjunto de passos foi seguido, conforme representado na Figura 1. A figura ilustra de forma sequencial as etapas que foram executadas, começando com a identificação do estado da arte, passando pela análise das abordagens existentes e pela definição das características únicas dos *chatbots*. Desta maneira, desenvolveu-se o Tesutã, bem como as heurísticas de apoio, culminando na criação de uma wiki para disponibilização do conteúdo. As etapas finais incluem o planejamento e a execução de um treinamento sobre testes metamórficos, além da avaliação da efetividade do guia proposto.

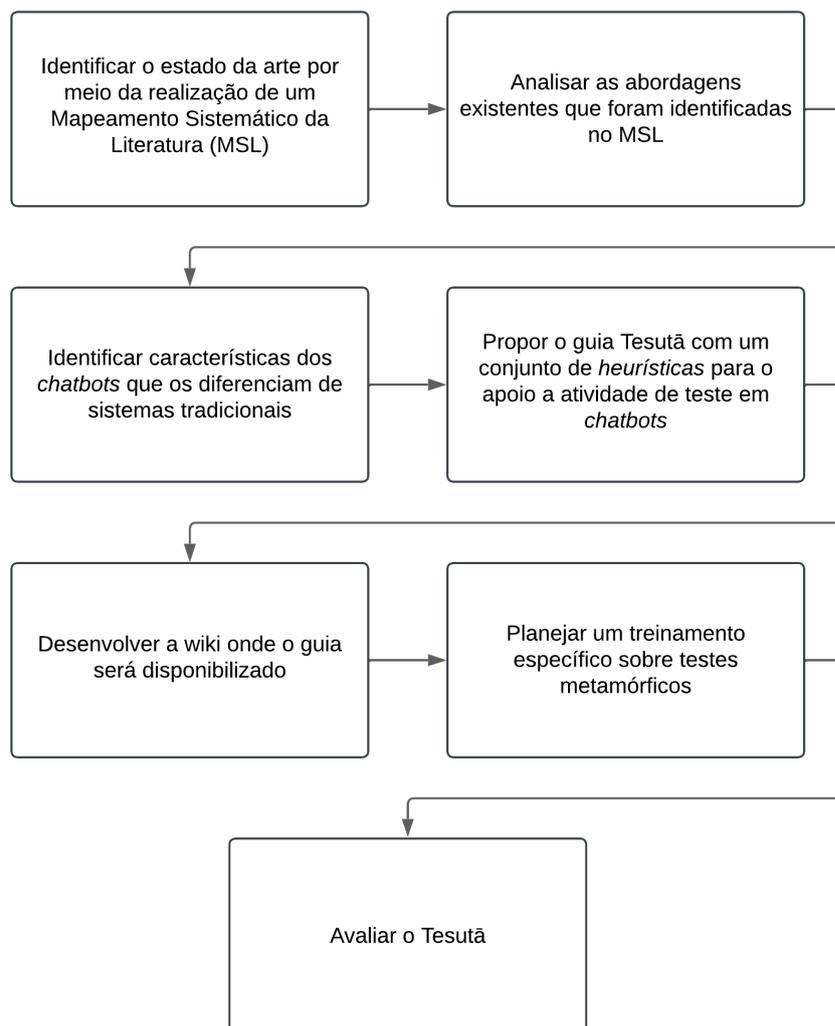


Figura 1: Método de Pesquisa

1.5 PRINCIPAIS RESULTADOS

Os resultados obtidos a partir da avaliação do *Tesutã* demonstraram sua capacidade de auxiliar na aplicação de testes metamórficos em *chatbots*, contribuindo para a melhoria da qualidade e confiabilidade desses sistemas. A validação foi conduzida por meio de um experimento com estudantes, que utilizaram o guia proposto para identificar requisitos e relações metamórficas em cenários de teste.

Durante a análise dos resultados, foi avaliada a completude dos requisitos identificados, ou seja, a proporção de requisitos de teste que os participantes conseguiram mapear corretamente em comparação ao oráculo definido previamente. O *Tesutã* permitiu que os participantes alcançassem uma taxa de completude de 75,70%, indicando que a maioria dos requisitos essenciais foi identificada corretamente.

Além disso, foram analisadas as relações metamórficas identificadas, que correspondem àquelas que descrevem como os resultados esperados devem se comportar mediante variações nas entradas. Os participantes identificaram 48 requisitos únicos e 23 relações metamórficas adicionais que não estavam previamente contempladas pelo oráculo, demonstrando a capacidade do guia de estimular a reflexão crítica e a identificação de novas abordagens para a validação de *chatbots*.

Entretanto, a completude das relações metamórficas do oráculo obteve uma média de apenas 21,42%, indicando que os participantes tiveram dificuldades em mapear todas as relações metamórficas previamente estabelecidas. Esse resultado sugere a necessidade de materiais complementares, exemplos práticos e possíveis ajustes na estrutura do guia para facilitar a compreensão e aplicação dessa técnica.

Por fim, a percepção geral dos participantes sobre o *Tesutã* foi positiva, com 76,9% dos estudantes avaliando a clareza e a organização das instruções de forma favorável. Esse feedback reforça o potencial do guia como ferramenta de apoio ao teste de *chatbots*, ao mesmo tempo que destaca oportunidades de melhorias para torná-lo mais acessível e eficiente para diferentes perfis de usuários. Esses achados apontam para a necessidade de aprimoramentos futuros, como a inclusão de exemplos mais detalhados e treinamentos direcionados, visando ampliar sua

aplicabilidade e impacto na prática de teste de *chatbots*.

1.6 ORGANIZAÇÃO DO TRABALHO

Este capítulo introduziu o contexto no qual este projeto de mestrado está inserido, destacando as motivações que impulsionaram sua realização, bem como os objetivos definidos. No Capítulo 2 são apresentados os principais conceitos sobre teste de software e chatbots, estabelecendo a base teórica necessária para o entendimento do projeto de mestrado. O Capítulo 3 descreve detalhadamente o Tesutã, incluindo sua conceituação, desenvolvimento e aplicação prática. No Capítulo 4 é realizada a avaliação do Tesutã, explorando sua eficácia e limitações com base nos experimentos conduzidos. Por fim, o Capítulo 5 apresenta as considerações finais, discutindo as contribuições do trabalho e sugerindo possíveis direções para pesquisas futuras.

2 FUNDAMENTAÇÃO TEÓRICA E TRABALHOS RELACIONADOS

Esse capítulo apresenta uma visão geral sobre os principais conceitos sobre teste de software e *chatbots*. Esses conceitos são importantes para entender a proposta desse projeto de mestrado.

2.1 TESTE DE SOFTWARE

Antes de aprofundar o entendimento sobre o conceito de Teste de Software, é fundamental compreender o conceito de qualidade em sistemas de software. A qualidade de software se refere à medida em que um software atende aos requisitos definidos, assim como às expectativas dos usuários (ISO, 2014). Além disso, envolve a avaliação de diversos atributos de qualidade como (ISO, 2014): funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade, portabilidade e segurança. Portanto, a qualidade de software busca garantir que um produto seja livre de defeitos, desempenhe adequadamente as funções para as quais foi projetado e proporcione uma boa experiência de uso (ISO, 2014).

A Verificação é o processo de avaliar se um sistema ou componente de software atende às especificações e aos requisitos estabelecidos no início do projeto (DELAMARO et al., 2016), envolvendo atividades como revisões de documentos, inspeções de código-fonte e análise estática para identificar possíveis erros, inconsistências ou violações das normas definidas (DELAMARO et al., 2016).

Por outro lado, a Validação tem como objetivo avaliar se o sistema de software em desenvolvimento atende às necessidades e expectativas dos usuários finais (DELAMARO et al., 2016), envolve a realização de testes funcionais, de desempenho, de usabilidade e outros tipos de testes para verificar se o software está correto,

completo e adequado para seu uso pretendido (DELAMARO et al., 2016). Ammann e Offutt (2017) reforçam a importância dessas atividades ao destacar que técnicas formais de verificação podem aumentar a confiabilidade do software ao detectar falhas antes da fase de teste dinâmico (AMMANN; OFFUTT, 2017).

Por outro lado, a Validação tem como objetivo avaliar se o sistema de software em desenvolvimento atende às necessidades e expectativas dos usuários finais (DELAMARO et al., 2016). Esse processo envolve a realização de testes funcionais, de desempenho, de usabilidade e outros tipos de testes para verificar se o software está correto, completo e adequado para seu uso pretendido (DELAMARO et al., 2016). Ammann e Offutt (2017) enfatiza que a validação deve incluir estratégias baseadas em critérios de teste bem definidos para garantir que o sistema satisfaça os requisitos do usuário (AMMANN; OFFUTT, 2017). Em resumo, a Verificação visa garantir que o software seja desenvolvido corretamente de acordo com as especificações, enquanto a Validação busca garantir que o software seja apropriado e atenda às necessidades dos usuários (DELAMARO et al., 2016). Ambas as atividades de Verificação e Validação são fundamentais para assegurar a qualidade e confiabilidade do software (DELAMARO et al., 2016).

O teste de software tem como objetivo executar programas ou modelos com entradas específicas, analisando se o comportamento observado corresponde ao esperado, conduzindo uma comparação detalhada para identificar e corrigir os defeitos que causaram as falhas (DELAMARO et al., 2016). Além disso, o teste de software pode fornecer informações objetivas e medidas sobre a qualidade do produto, ajudando a mitigar riscos e fornecer confiança aos usuários finais. O teste de software visa revelar falhas no software antes de sua liberação, permitindo que sejam corrigidas o mais rápido possível, possibilitando assim que o software atenda às necessidades e expectativas dos usuários (DELAMARO et al., 2016).

Portanto, o teste de software também ajuda a melhorar a qualidade do processo de desenvolvimento, fornecendo *feedback* contínuo sobre o progresso do projeto e a adequação dos requisitos e especificações (DELAMARO et al., 2016). Nesse sentido, o teste de software contribui para o desenvolvimento de sistemas mais confiáveis, eficientes e seguros (DELAMARO et al., 2016). Outro aspecto importante

é a Verificação e Validação (V&V), que são atividades essenciais no processo para a garantia da qualidade dos produtos de software (DELAMARO et al., 2016). Aniche (2022) destaca a importância da automação de testes como forma de garantir que as verificações sejam reexecutadas continuamente ao longo do ciclo de desenvolvimento, aumentando a eficácia da V&V (ANICHE, 2022).

2.1.1 CONCEITOS BÁSICOS

Em relação à atividade de teste de software é comum encontrar diferentes termos como: defeito, engano, falha, erro, dentre outros. Apesar desses termos serem utilizados no dia a dia e em diferentes domínios, a IEEE¹ (IEEE, 1990) propôs uma definição para cada um deles, conforme apresentado a seguir:

- **Defeito:** passo, processo ou definição de dados incorretos;
- **Engano:** ação humana que ocasiona em um resultado incorreto;
- **Falha:** produção de uma saída incorreta com relação à especificação, ou seja, ocorre quando um sistema ou componente não se comporta conforme esperado;
- **Erro:** diferença entre valor obtido e valor esperado, ou seja, é o resultado de um engano, podendo ser ocasionado por um defeito introduzido no código, projeto ou no sistema.

Além destes termos, Delamaro et al. (2016) também apresentam outros termos que são comuns e pertinentes à atividade de teste em software, e são relevantes para entender o contexto de teste de software. São eles:

- **Domínio de entrada:** é o conjunto de valores que podem ser utilizados como entrada durante a execução do teste de um determinado programa. O domínio de entrada envolve todos os valores possíveis que podem ser fornecidos ao programa para testar seu comportamento e funcionalidade;
- **Dado de teste:** são os dados de entrada e as condições associadas a um determinado caso de teste selecionados a partir de um domínio de entrada;

¹ *Institute of Electrical and Electronic Engineers*

- **Caso de teste:** são elementos que definem os dados de entrada do teste e as saídas esperadas a partir da execução de um teste;
- **Conjunto de casos de teste:** representa o grupo de todos os casos de teste projetados durante uma atividade de teste;
- **Oráculo:** é o instrumento que verifica se a saída obtida a partir da execução do programa ou software em teste coincide com a saída esperada pelo caso de teste. O oráculo pode ser uma pessoa ou ferramenta que automatiza essa atividade.

A Figura 2 apresenta o cenário típico da atividade de teste, onde T representa o conjunto de casos de teste, P refere-se ao software que está sendo testado, operando com elementos dos domínios de entrada D e domínios de saída S. No contexto do teste de software, o domínio de entrada $D(P)$ é composto pelos conjuntos de valores que podem ser utilizados para realizar a execução do programa. Por sua vez, o domínio de saída $S(P)$ é o conjunto correspondente de valores resultantes da execução do programa para os dados de entrada fornecidos (DELAMARO et al., 2016).

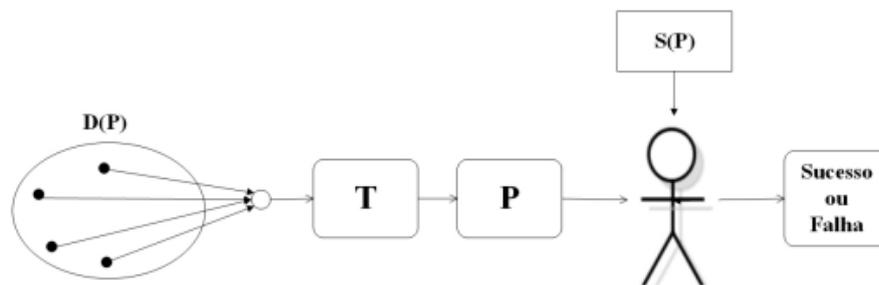


Figura 2: Cenário típico da atividade de teste. Adaptado de Delamaro et al. (2016)

Dessa forma, o testador atua como um oráculo, tomando decisões com base nos dados de saída $S(P)$ para determinar se algum erro foi revelado durante a execução dos testes. Em outras palavras, o testador compara os resultados obtidos $S(P)$ com o esperado, desempenhando o papel crítico de identificar discrepâncias e potenciais falhas no software testado (DELAMARO et al., 2016).

2.1.2 TÉCNICAS E CRITÉRIOS DE TESTE

Nesta subseção apresentam-se as principais técnicas e critérios de teste. O **teste funcional** é uma técnica que se concentra na verificação do comportamento funcional de um sistema ou componente (DELAMARO et al., 2016). O teste funcional é baseado na especificação, ou seja, essa técnica não considera aspectos do código e, portanto, verifica se o software em teste cumpre os requisitos funcionais especificados, com base na documentação do software. Ele se concentra em testar as funcionalidades e recursos do software, garantindo que eles funcionem conforme o esperado e atendam aos requisitos do sistema.

Além disso, é necessário entender que um critério de teste é um conjunto de regras ou diretrizes usadas para determinar se um sistema de software ou componente atende aos requisitos de teste estabelecidos e funciona conforme o esperado (RIOS, 2010). Esses critérios são usados para projetar casos de teste, executar testes e avaliar os resultados obtidos. Eles ajudam a garantir que todos os aspectos importantes do software sejam testados de maneira abrangente e sistemática (RIOS, 2010).

Portanto, estabeleceram-se critérios específicos para o teste funcional, sendo eles: particionamento em classes de equivalência e a análise do valor limite, conforme descritos a seguir:

- **Particionamento em classes de equivalência:** visa reduzir a quantidade de dados de entrada do domínio de entrada, permitindo que um subconjunto representativo seja selecionado. Esse critério tem como objetivo revelar a presença de falhas que seriam descobertos apenas por meio de um teste completo e abrangente de todas as combinações possíveis de dados de entrada (DELAMARO et al., 2016). Ao utilizar o particionamento em classes de equivalência, busca-se reduzir a quantidade de dados de entrada, agrupando os dados de entrada em conjuntos que compartilham características semelhantes. Assim, em vez de testar cada valor individualmente, é possível focar dados mais representativos de cada classe, maximizando a cobertura e economizando tempo e recursos (DELAMARO et al., 2016). Dessa forma, o particionamento

em classes de equivalência oferece uma abordagem eficiente para explorar a criação de casos de teste, sem a necessidade de testar exaustivamente todas as combinações de entrada (DELAMARO et al., 2016).

- **Análise do valor limite:** é um critério que complementa o particionamento em classes de equivalência, pois se concentra em testar os limites de cada classe. Esse critério seleciona valores que estão exatamente no limite superior, inferior ou imediatamente ao redor dos limites das classes de equivalência. Em vez de escolher valores aleatórios dentro das classes, como mencionado anteriormente, a análise do valor limite extrai especificamente valores dos limites das classes. Esse critério proporciona uma cobertura adicional e mais aprofundada (DELAMARO et al., 2016).

O **teste estrutural** se concentra na análise da estrutura interna do programa ou software (ASSUNÇÃO, 2015). Nesse teste, o testador considera os dados de entrada, a semântica e a sintaxe do programa, bem como o programa em execução e os dados de saída, a fim de elaborar os casos de teste (ASSUNÇÃO, 2015; CRESPO, 1997). Enquanto o teste funcional é comparado a uma “caixa-preta”, o teste estrutural pode ser relacionado a uma “caixa-branca”. Isso ocorre porque o testador tem acesso a estrutura interna do software, podendo examinar seu código-fonte, fluxo de controle, caminhos de execução e estruturas de dados (BARBOSA et al., 2007).

Durante o teste estrutural, o foco está na cobertura dos diferentes caminhos e condições do programa, a fim de identificar possíveis erros lógicos, fluxos incorretos ou vulnerabilidades (BARBOSA et al., 2007). Essa abordagem é especialmente útil para testar componentes críticos, como algoritmos complexos ou módulos cruciais, visando garantir que o software funcione corretamente em um nível mais detalhado (BARBOSA et al., 2007). A seguir são apresentados os principais critérios relacionados a essa técnica:

- **Critérios baseados em fluxo de controle:** são utilizados para garantir que todos os caminhos possíveis de execução do programa sejam exercitados (ZHOU et al., 2015; BARBOSA et al., 2007). Esses critérios levam em consideração a estrutura do fluxo de controle do programa, incluindo os

diferentes ramos condicionais, e sequências de instruções. O objetivo é identificar e testar todas as condições e ramificações lógicas presentes no código, garantindo uma cobertura abrangente do fluxo de controle (ZHOU et al., 2015; BARBOSA et al., 2007). Existem vários critérios baseados em fluxo de controle, como: critério de cobertura de decisão, cobertura de condição, cobertura de caminho e cobertura básica de fluxo (ZHOU et al., 2015). Cada uma dessas técnicas visa atingir diferentes níveis de cobertura do fluxo de controle, com o objetivo de encontrar erros relacionados a decisões lógicas, *loops* infinitos, falta de execução de determinadas partes do código e outras falhas de fluxo (CRESPO, 1997; BARBOSA et al., 2007).

- **Crítérios baseados em fluxo de dados:** critérios utilizados para garantir que todas as interações entre as variáveis do programa sejam exercitadas (BARBOSA et al., 2007). Esses critérios são baseados na análise das dependências de dados entre as variáveis do programa, buscando identificar todas as possíveis combinações e modificações dos dados durante a execução do programa. O objetivo é encontrar erros relacionados a problemas de definição, inicialização, uso ou atualização das variáveis, bem como problemas de fluxo incorreto dos dados (CRESPO, 1997; BARBOSA et al., 2007).

Existem diferentes critérios baseadas em fluxo de dados, como *Data Flow Coverage*², *Variable Usage Coverage*³, *Def-Use Coverage*⁴, *Control Flow Path Coverage*⁵ e *Data Dependency Coverage*⁶ (BARBOSA et al., 2007). Cada uma dessas técnicas tem como objetivo garantir uma cobertura abrangente das interações entre as variáveis, de modo a encontrar possíveis erros relacionados a inconsistências de dados e problemas de fluxo de dados (CRESPO, 1997; BARBOSA et al., 2007).

O teste de mutação ou análise de mutantes é o critério mais conhecido da técnica de **teste baseado em defeitos** (VALLE, 2016). Esse critério envolve

²Data Flow Coverage: Cobertura de Fluxo de Dados

³Variable Usage Coverage: Cobertura de Uso de Variáveis

⁴Def-Use Coverage: Cobertura de Definição-Utilização

⁵Control Flow Path Coverage: Cobertura de Caminhos de Controle

⁶Data Dependency Coverage: Cobertura de Dependência de Dados

a introdução de mutações ou alterações deliberadas no código-fonte do programa original, a fim de avaliar a eficácia dos casos de teste existentes (DELAMARO et al., 2016). O teste de mutação consiste em criar versões mutantes do programa, introduzindo defeitos artificiais ou mutações no código-fonte original. Cada mutação representa uma possível falha no programa. Em seguida, são aplicados os casos de teste existentes no programa original às versões mutantes, verificando se esses casos de teste conseguem detectar as mutações introduzidas (DELAMARO et al., 2016).

O objetivo do teste de mutação é avaliar a capacidade dos casos de teste em identificar e eliminar as mutações. Se um caso de teste consegue detectar uma mutação, significa que ele é eficaz na detecção de erros no programa original. Por outro lado, se uma mutação não é detectada por casos de teste, isso pode indicar uma falha na cobertura dos casos de teste existentes (DELAMARO et al., 2016). O teste de mutação é uma técnica exigente e computacionalmente intensiva, pois envolve a geração e execução de várias versões mutantes do programa. No entanto, essa abordagem é considerada eficaz para avaliar a qualidade dos casos de teste e melhorar a robustez do software, uma vez que identifica áreas com baixa cobertura de teste (DELAMARO et al., 2016).

2.1.3 TESTE METAMÓRFICOS

Os testes metamórficos transcendem as limitações dos métodos convencionais ao explorar não apenas os resultados finais de um sistema, mas, crucialmente, as transformações observadas nas saídas quando as entradas sofrem alterações específicas (SILVA, 2022). Esta abordagem paradigmática busca não apenas validar a corretude absoluta, mas compreender como o sistema responde a variações nas condições iniciais, proporcionando uma compreensão mais abrangente do seu comportamento (SILVA, 2022).

A principal vantagem dos testes metamórficos reside na redução da dependência de oráculos, superando as limitações associadas à definição rígida de respostas “corretas” (SILVA, 2022). Essa flexibilidade, combinada com a capacidade de modelar transformações nas entradas, torna-os particularmente aptos a enfrentar desafios complexos, onde a determinação de corretude absoluta pode ser ambígua

ou inatingível (SILVA, 2022).

Segura et al. (2016) apresentaram em seu trabalho um processo básico para aplicação de testes metamórficos. Esse processo pode ser dividido em três etapas, sendo elas: Identificação das relações metamórficas, Criação de um conjunto de dados de teste e Detecção de falhas. Andrade (2022) representou essas etapas visando um panorama geral para aplicação destes testes, conforme descrito a baixo:

- **Identificação das relações metamórficas:** Segundo Andrade (2022), este é um - “passo fundamental para o sucesso na aplicação da abordagem, o primeiro passo é a identificação de um conjunto de relações metamórficas. As relações metamórficas são derivadas utilizando um conhecimento a respeito do domínio de aplicações de software em teste”. Além disso, Andrade (2022) identificou que as abordagens descritas na literatura para definição destas relações envolvem: desenvolver as relações com base na especificação do software ou com base nos requisitos esperados pelo usuário. Desta maneira, a seleção da abordagem implica diretamente em como a metamorfose será testada, ou seja, as especificações foram atendidas após a metamorfose ou se os requisitos do usuário foram atendidos. Nem sempre a segunda opção reflete em propriedades necessárias para que o sistema funcione corretamente, ou seja, algumas funcionalidades podem ficar de fora neste cenário (ANDRADE, 2022; SEGURA et al., 2016).
- **Criação de um conjunto de dados de teste:** Para realizar esta etapa, é preciso criar casos de teste iniciais e casos de teste de acompanhamento. Segundo Andrade (2022), para criar os casos de teste iniciais pode ser utilizado qualquer técnica de teste convencional, por exemplo, técnica estrutural, funcional, baseada em defeitos ou teste aleatório⁷. O próximo passo é criar o caso de teste de acompanhamento que são realizados alternando os casos iniciais de acordo com as modificações que foram definidas pelas relações metamórficas (ANDRADE, 2022).
- **Detecção de falhas:** Segundo Andrade (2022), cada um dos pares de teste

⁷Teste aleatório: são testes gerados de forma aleatória, sem seguir uma estratégia específica de seleção.

iniciais e de acompanhamento são executados para validar se a alteração corresponde à saída e se está prevista pela relação metamórfica definida. Quando ocorre uma violação desta relação em tempo de execução, isto significa uma prerrogativa para uma possível falha no programa que está sendo testado. Sendo assim, os testes metamórficos verificam a relação dos dados de entrada e de saída, isso possibilita verificar a conformidade do software mesmo quando o resultado (saída) é desconhecido (ANDRADE, 2022).

Particularmente relevante no âmbito dos *chatbots*, os testes metamórficos oferecem um arcabouço sistemático para avaliar a eficácia e a utilidade desses agentes virtuais. Ao introduzir variações nas consultas dos usuários, a abordagem metamórfica permite uma avaliação contextualizada das respostas, refletindo diretamente na adaptabilidade e desempenho do chatbot em cenários diversificados (SILVA, 2022).

2.2 AGENTES CONVERSACIONAIS BASEADOS EM TEXTO

Agentes conversacionais baseados em texto, também são conhecidos por *chatbots* ou assistentes virtuais, são programas de computador projetados para simular uma conversa entre dois humanos (PASCHOAL et al., 2020). Essas ferramentas podem utilizar técnicas de processamento de linguagem natural (PLN) para compreender e gerar respostas em linguagem natural, permitindo assim a comunicação com humanos (SOUZA, 2022; PASCHOAL et al., 2020).

Este tipo de programa pode ter várias características, como interação em linguagem natural - compreensão e geração de respostas em linguagem natural; personalização - utiliza aprendizado de máquina para aprender com o usuário e se adaptar; eficiência - como qualquer outro software, pode ser altamente escalável e receber diversas requisições simultâneas (SOUZA, 2022; PASCHOAL et al., 2020).

Conforme discutido anteriormente, os *chatbots* podem estar inseridos em diversos domínios. No âmbito da web, estas aplicações são utilizadas diariamente por milhares de usuários para diálogos genéricos, intermediar acesso a outros sistemas, interpretar papel de “acompanhantes digitais” em sistemas imersivos

(TEDE; BARROS, 2016). Sendo assim, estes programas podem ser encontrados em aplicações isoladas na web para conversação, pode estar embutido em sistemas interativos variados (TEDE; BARROS, 2016).

2.2.1 TIPOS DE AGENTES CONVERSACIONAIS

Estes programas podem ser desenvolvidos de diversas formas (MORAIS; SANTIAGO, 2023). Portanto podem ser divididos em duas categorias, conforme apresentado abaixo.

- **Agentes baseados em regras:** sistemas que utilizam um conjunto de regras pré-definidas, isto é, caso o usuário pergunte algo fora do escopo para qual o agente foi desenvolvido, não será possível responder adequadamente ou será apresentado uma resposta padrão. Cada regra consiste em uma condição e uma ação correspondente. O agente avalia as condições no ambiente, aplica as regras atividades e executa as ações. No entanto, esse tipo de agente é limitado, pois não possuem a capacidade de aprendizado uma vez que as regras são fixas e pré-definidas (MORAIS; SANTIAGO, 2023; RUSSELL, 2010).
- **Agentes baseados em aprendizado de máquina:** sistemas que utilizam algoritmos de aprendizagem para adquirir conhecimento para tomada de decisão (MORAIS; SANTIAGO, 2023; BISHOP; NASRABADI, 2006). Este tipo de agente aprende e adapta seu comportamento com base nos dados fornecidos durante seu treinamento. Eles são compostos por um modelo de representação matemática para realizar a tomada de decisão. Além disso, por terem a capacidade de aprendizado, tornam-se flexíveis, ou seja, se tornam capazes de lidar com mais problemas e aprender com eles (MORAIS; SANTIAGO, 2023; BISHOP; NASRABADI, 2006).

2.2.1.1 VANTAGENS E LIMITAÇÕES DOS AGENTES CONVERSACIONAIS

A partir da vasta diversidade de aplicações, cada qual com suas vantagens e limitações, torna-se impraticável listar todos os tipos. Sendo assim, serão abordados

apenas as vantagens e limitações dos tipos de agentes conversacionais citados na subseção anterior.

- **Agentes baseados em regras**

- Vantagens: Implementação simples e de fácil compreensão; eficaz em cenários com um conjunto limitado de regras bem definidas (RUSSELL, 2010).
- Limitações: Problemas com identificação de ambiguidade, nuances e variações na linguagem. É necessário definir todas as regras antecipadamente o que torna essa classe de agentes inflexíveis em situações complexas (RUSSELL, 2010).

- **Agentes baseados em aprendizado de máquina**

- Vantagens: Consegue trabalhar em ambientes complexos e trata ambiguidades. É capaz de aprender com os dados de treinamento e adaptar-se com base em experiências anteriores (BISHOP; NASRABADI, 2006).
- Limitações: Requer um conjunto de dados de treinamento extremamente definido. Pode exigir recursos computacionais significativos (BISHOP; NASRABADI, 2006).

- **Agentes baseados em redes neurais**

- Vantagens: Redes neurais são capazes de aprender padrões complexos e representar relações lineares; podem lidar com problemas de processamento de linguagem natural, reconhecimento de padrões e visão computacional; pode adaptar-se para melhorar seu desempenho ao longo do tempo e a novos desafios (BISHOP; NASRABADI, 2006).
- Limitações: O treinamento dessa classe de agentes pode demandar um conjunto de dados de treinamento muito grande e irá demandar de alto poder computacional. Além disso, o processo de treinamento pode ser demorado e requerer ajustes. Durante o treinamento, pode ocorrer

*overfitting*⁸, onde a rede se ajusta excessivamente aos dados e passa a ter dificuldade em generalizar para novos dados (BISHOP; NASRABADI, 2006).

2.3 TRABALHOS RELACIONADOS

Nesta seção, serão apresentados os trabalhos relacionados obtidos a partir das questões de pesquisa da Revisão Sistemática da Literatura (RSL) publicada no *Workshop sobre Aspectos Sociais, Humanos e Econômicos de Software (WASHES)* (SANTOS et al., 2024).

Desta maneira, a fim de obter o estado da arte da área de teste em chatbots, conduziu-se uma RSL seguindo as diretrizes propostas por Kitchenham et al. (2010). O estudo foi realizado em três etapas: planejamento, execução e análise dos resultados.

Sendo assim, foram definidas as seguintes Questões de Pesquisa (QP) para serem respondidas a partir dos estudos selecionados (Tabela 1).

Tabela 1: Questões de Pesquisa (QP)

QP	Descrição
QP ₁	Quais abordagens (técnicas/critérios) têm sido utilizadas para apoiar a atividade de teste em <i>chatbots</i> ?
QP ₂	Como os autores automatizaram as atividades para apoiar os testes?
QP ₃	Qual o nível de teste tem sido aplicado nos teste em <i>chatbots</i> ?
QP ₄	Quais linguagens de programação estão sendo utilizadas para apoiar os testes?
QP ₅	Foi realizada uma avaliação experimental? Se sim, como foi feito?
QP ₆	Quais foram os domínios analisados nos trabalhos?
QP ₇	Qual a forma de interação do <i>chatbot</i> avaliado?

Para identificar os estudos analisados, definiu-se a *string de busca*: ((“testing” OR “validation” OR “verification”) AND (“software” OR “metamorphic”)) and (“chatbot” OR “chatbot testing”). A *string* de busca foi utilizada para identificar estudos primários nas seguintes bases de dados: ACM Digital Library, Science Direct, Scopus e IEEE Xplore. A busca foi conduzida considerando o período temporal de oito anos (2015-2023) com o objetivo de assegurar que os estudos sejam pertinentes e representativos, alinhando-se assim ao panorama atual do estado da arte desta

⁸*Overfitting*: significa que a rede neural se ajustou demais aos dados de treinamento, perdendo sua capacidade de generalização e levando a um desempenho pior em dados não vistos.

investigação. Considerou-se os últimos oito anos pois, este período representa o início de pesquisas relevantes sobre o tema considerado.

Foram também definidos os Critérios de Inclusão (CI) e Critérios de Exclusão (CE) para selecionar o os estudos que foram analisados, conforme descritos na Tabela 2.

Tabela 2: Critérios de seleção (inclusão e exclusão) da RSL.

ID	Descrição
CI ₁	Estudos disponíveis para a leitura.
CI ₂	Estudos escritos em língua inglesa e portuguesa.
CI ₃	Estudos que discutam teste de software em agentes conversacionais baseados em texto (<i>chatbot</i>).
CI ₄	Estudos que discutam métodos, técnicas e abordagens para testar <i>chatbots</i> .
CE ₁	Estudos que são livros, teses, dissertações, patentes, livros.
CE ₂	Estudos curtos (2 páginas) ou incompletos, tutoriais, propostas de <i>workshops</i> ou pôsteres.

2.3.1 EXECUÇÃO

Inicialmente, na fase de busca, a *string* de busca foi aplicada nas bases descritas anteriormente, resultando em 80 estudos. Em seguida, na fase de pré-seleção, procedeu-se à na leitura dos títulos e resumos dos estudos (Fase 1), resultando em 22 trabalhos selecionados. Posteriormente, na fase de seleção (Fase 2), foram lidos os textos de introdução e conclusão dos estudos pré-selecionados, resultando na escolha de 11 estudos para análise mais detalhada. Por fim, realizou-se uma leitura completa e minuciosa dos cinco estudos finais (Fase 3), os quais foram selecionados após uma análise criteriosa. Vale ressaltar que foram adotados os critérios da Tabela 2 para seleção dos estudos em cada uma das fases. Devido à limitação na quantidade de estudos primários, decidiu-se complementar a RSL com a técnica de *snowballing backward* e o *snowballing forward* (KITCHENHAM et al., 2010).

Utilizando o *snowballing backward*, foram considerados um total de 145 estudos relacionados ao tópico de pesquisa. Na Fase 1, foram selecionados 31 estudos; na Fase 2, 16 estudos foram selecionados; e na Fase 3, sete estudos foram selecionados. Por outro lado, ao considerar o *snowballing forward* foram recuperados 16 estudos. Na Fase 1, foram selecionados cinco estudos; na Fase 2, três estudos foram selecionados; e na Fase 3, apenas um estudo foi escolhido.

2.3.2 RESULTADOS

Como resultado, foram selecionados 13 estudos (ver Tabela 3). Ao observar a distribuição temporal, nota-se que o ano de 2019 representou o período em que a temática recebeu maior atenção. Nos anos seguintes, houve uma queda, não implica na perda de importância, mas sim em um ajuste de atenção à medida que o campo amadurece.

Tabela 3: Estudos primários selecionados

ID	Ano	Título do trabalho	Snowballing	Referência
E_1	2020	Algorithm Inspection for Chatbot Performance Evaluation	Não	(VIJAYARAGHAVAN et al., 2020)
E_2	2018	BoTest: A framework to test the quality of conversational agents using divergent input examples	Sim	(RUANE et al., 2018)
E_3	2017	Bottester: testing conversational systems with simulated users	Sim	(VASCONCELOS et al., 2017)
E_4	2019	Chatbot and bullyfree chat	Sim	(SELVI et al., 2019)
E_5	2023	Chatbot Interaction with Artificial Intelligence: human data augmentation with T5 and language transformer ensemble for text classification	Sim	(BIRD et al., 2023)
E_6	2019	Chatbot testing using AI planning	Sim	(BOZIC et al., 2019)
E_7	2020	OggyBug: A Test Automation Tool in Chatbots	Não	(SANTOS et al., 2020)
E_8	2022	Ontology-based metamorphic testing for chatbots	Não	(BOŽIĆ, 2022)
E_9	2022	Sorry, i don't Understand: Improving Voice User Interface Testing	Não	(GUGLIELMI et al., 2022)
E_{10}	2019	Sustainable Test Path Generation for Chatbots using Customized Response	Não	(PADMANABHAN, 2019)
E_{11}	2021	Testing challenges for NLP-intensive bots	Sim	(CABOT et al., 2021)
E_{12}	2019	Testing Chatbots Using Metamorphic Relations	Sim	(BOZIC; WOTAWA, 2019)
E_{13}	2020	Testing chatbots with Charm	Sim	(BRAVO-SANTOS et al., 2020)

A Tabela 4 sumariza as respostas das QPs, destacando as abordagens utilizadas (QP_1), o nível de teste (QP_3), a linguagem de programação de apoio a condução dos testes (QP_4), a validação da abordagem (QP_5), o domínio ao qual cada estudo está associado (QP_6) e a forma de interação do *chatbot* (QP_7).

2.3.3 Q_1 : ABORDAGENS (TÉCNICAS/CRITÉRIOS) UTILIZADAS EM TESTE DE CHATBOTS

Diversas abordagens têm sido exploradas para apoiar a atividade de teste em *chatbots*, incluindo técnicas e critérios específicos. Os principais achados indicam

Tabela 4: Resumo das Respostas

ID	Ferramenta	Nível/Técnica	Linguagem	Validação	Domínio
E_1	-	-	-	Não	Não definido
E_2	BoTest	-	-	Sim	Entretenimento
E_3	Bottester	-	-	Sim	Finanças
E_4	Anaconda	-	Python	Não	Bem-Estar Social
E_5	Modelo de parafraseamento T5	-	Java	Sim	-
E_6	Algoritmos de planejamento em conjunto com a Planning Domain Definition Language	-	-	Sim	Hotelaria
E_7	Testes automatizados por meio de APIs, através de uma interface web.	Integração	JavaScript	Sim	Advocacia
E_8	-	Unidade	-	Sim	Hotelaria
E_9	-	-	Python	Não	Não definido
E_{10}	-	Testes funcionais	-	Sim	Educação
E_{11}	Botium, Zypnos, Chatbottest, QBox, DashBot e Botest	Unidade	-	Não	Não definido
E_{12}	-	Metamórficos	-	Sim	Hotelaria
E_{13}	Charm	-	-	Sim	Hotelaria, Alimentação

a utilização de abordagens como testes funcionais ($E_1, E_2, E_3, E_4, E_6, E_9, E_{11}$ e E_{13}), unitários (E_4), metamórficos (E_8, E_{12} e E_{13}) e de integração (E_1, E_2, E_3, E_7, E_9 e E_{11}). Por exemplo, no estudo E_2 foram aplicados testes funcionais e de integração para avaliar o *chatbot* ChitChatBot. Os testes funcionais verificaram se o *chatbot* executava as funções esperadas e se respondia corretamente às entradas do usuário. Os autores relatam que o ChitChatBot foi implementado utilizando o *Language Understanding Intelligent Service* (LUIS), exigindo testes de integração para verificar a a correta integração com o LUIS e a adequada interpretação e respostas às entradas dos usuários.

2.3.4 Q_2 : AUTOMATIZAÇÃO DAS ATIVIDADES DE TESTES EM CHATBOTS

No estudo E_2 , foi empregada a ferramenta **BoTest**, um sistema modular capaz de integrar diversas técnicas, como detecção de erros e estilo linguístico. A detecção de erros abrange divergências sintáticas, morfológicas e semânticas, implementada por meio da introdução controlada de variações na entrada de teste. Exemplos específicos incluem erros de preposição não nativa e expressões coloquiais nativas, desafiando a capacidade do agente em lidar com diferentes formas de entrada. Quanto ao estilo linguístico, a avaliação envolve a introdução de elementos que refletem nuances de estilo na linguagem, como formalidade ou informalidade. Dentro do contexto do **BoTest**, isso foi realizado por meio da introdução de divergências no estilo coloquial nativo, incorporando expressões informais ou regionais. Essas

estratégias visam testar a capacidade do agente em compreender e reagir de maneira adequada diante de variações no estilo linguístico.

No estudo E_3 foi utilizada a ferramenta Bottester para simular interações de usuários com o *chatbot* e coletar métricas relacionadas às conversas. As métricas coletadas foram: Tamanho médio da resposta - avaliando a concisão das respostas, medindo o tamanho em caracteres e palavras de cada resposta do *chatbot*; Frequência de respostas - observando a frequência com que determinadas respostas são apresentadas, indicando possíveis limitações na base de conhecimento do *chatbot*; Frequência de palavras - analisando a frequência de palavras nas respostas, destacando padrões de linguagem e possíveis limitações no vocabulário do *chatbot*; Número de respostas corretas/incorretas - avaliando a capacidade do *chatbot* em fornecer respostas corretas, comparando-as com as respostas esperadas definidas nos casos de teste; e Tempo médio de resposta - mensurando o intervalo de tempo entre a submissão da pergunta e a chegada da resposta, sendo uma métrica-chave relacionada à percepção de qualidade do serviço pelo usuário.

No estudo E_4 , foi utilizada a distribuição Anaconda de Python para gerenciar o ambiente virtual e as dependências do projeto. O ambiente virtual fornecido pela Anaconda foi essencial durante os testes, garantindo um isolamento eficaz das dependências específicas. Além disso, o uso do Jupyter Notebook, integrado à distribuição Anaconda, foi fundamental para a execução interativa de trechos de código, facilitando a análise e verificação incremental do desempenho do chatbot e do algoritmo de detecção de *cyberbullying*. A combinação dessas ferramentas proporcionou um ambiente coeso e eficiente para os testes, contribuindo para a estabilidade, reprodutibilidade e gerenciamento simplificado das dependências do projeto.

No estudo E_5 , foi utilizado o modelo T5 (Text-To-Text Transfer Transformer) para expandir o conjunto de treinamento com dados adicionais. O T5 é um modelo de linguagem que possui a capacidade única de abordar várias tarefas de Processamento de Linguagem Natural (PLN) como problemas de conversão de texto para texto. Essa capacidade possibilita a geração de paráfrases, essenciais para diversificar e enriquecer o conjunto de dados de treinamento. Além do T5, foram empregadas várias

outras ferramentas para avaliar e aprimorar o desempenho do modelo, tais como BERT, DistilBERT, RoBERTa e XLM-RoBERTa, além de estratégias como *Logistic Regression* e *Random Forests*. Essas ferramentas foram cruciais para analisar a eficácia do modelo proposto em interações de chatbot, classificação de texto e reconhecimento de sentimentos.

No estudo E_6 , foram utilizados algoritmos de planejamento com a *Planning Domain Definition Language* (PDDL), uma linguagem usada em Inteligência Artificial para descrever domínios e problemas de planejamento. A PDDL permitiu modelar ações, parâmetros e intenções do usuário para os testes funcionais em chatbots. A implementação da abordagem de teste utilizou a linguagem Java com um framework de execução de testes. A ferramenta jsoup foi utilizada para analisar as respostas do chatbot durante os testes. Os algoritmos de planejamento, especialmente o *Fast Downward Planning System*, foram essenciais para criar sequências de ações para os cenários de teste. A PDDL desempenhou um papel crucial na definição das condições iniciais, ações possíveis e condições de objetivo. Essas ferramentas possibilitaram a execução automatizada de casos de teste abstratos, avaliando o comportamento do chatbot em várias situações e garantindo uma abordagem eficiente nos testes funcionais.

No estudo E_7 , foram realizados testes automatizados utilizando APIs⁹, por meio de uma interface *Web*. Os testes incluíram: Testes de Reconhecimento de Padrões, que visam evoluir a base de conhecimento dos *chatbots* e identificar conflitos entre os módulos; Testes de Variáveis de Contexto, para verificar o contexto de diálogos realizados pelos *chatbots*; e Testes de Integração, para lidar com a troca de dados com outros sistemas. Esses testes são particularmente relevantes quando os *chatbots* precisam interagir com serviços externos.

No estudo E_{11} , foram mencionadas várias ferramentas desenvolvidas pelos autores, incluindo Botium, Zynpos, Chatbottest, QBox, DashBot, Botest, entre outras. Já no estudo E_{13} , a ferramenta CHARM foi utilizada, com Botium como backend para a execução dos testes automatizados. Nos demais estudos, não foram encontradas informações sobre ferramentas específicas utilizadas para suportar a atividade de

⁹API: *Application Programming Interface*

teste em *chatbots*.

2.3.5 Q_3 : NÍVEL DE TESTE EM *CHATBOTS*

No estudo E_1 foram conduzidos testes de integração, enquanto nos estudos E_3 e E_{11} foram realizados testes de unidade. O estudo E_4 abordou vários níveis de teste nos *chatbots*, incluindo testes unitários para avaliar partes específicas do código, como funções de processamento de texto, algoritmos de aprendizado de máquina e detecção de *cyberbullying*. Os testes de integração verificaram a interação entre os diferentes componentes do *chatbot*, como o processamento de linguagem natural e a lógica de resposta. Os testes de aceitação foram realizados para garantir que o *chatbot* atendesse às expectativas do usuário e cumprisse os requisitos funcionais e não funcionais estabelecidos, incluindo a usabilidade, eficiência e capacidade de resposta. Por sua vez, no estudo E_{12} , foram conduzidos teste de sistema. Em relação aos demais estudos, não foram identificadas evidências sobre os níveis de teste considerados para o contexto de *chatbot*.

2.3.6 Q_4 : LINGUAGENS DE PROGRAMAÇÃO UTILIZADAS NOS TESTES

No estudo E_7 , a linguagem de programação adotada foi JavaScript. Nos estudos E_9 e E_4 , a linguagem escolhida foi Python. Já no estudo E_5 , a linguagem de programação utilizada foi Java. Nos demais estudos, não foi adotada uma linguagem específica, uma vez que se basearam em ferramentas já existentes.

2.3.7 Q_5 : AVALIAÇÃO EXPERIMENTAL DAS ABORDAGENS

No estudo E_7 , a abordagem proposta foi validada por meio da realização de testes com duas empresas distintas, com o objetivo de verificar sua aplicabilidade em domínios e em plataformas de criação/gerenciamento de *bots* diferentes. Já o estudo E_8 , a abordagem foi avaliada em um Sistema de Teste (SUT) que processa entradas em linguagem natural, utilizando um caso de teste que representava o cenário de teste.

No estudo E_{10} , a validação da abordagem foi realizada por meio da execução

de testes funcionais. Por sua vez, no estudo E_3 , embora não tenha sido explicitamente mencionada a validação da abordagem usando o Bottester, foi conduzido um "teste de sanidade" para o CognIA. Esse teste teve como objetivo inicial verificar a integridade e funcionalidade básica do sistema, com foco na correta resposta a todas as perguntas implementadas. Durante essa avaliação, foram contabilizadas as respostas corretas e incorretas, além da análise do tempo de resposta para identificar os tipos de perguntas com maior demanda de tempo. Essa etapa inicial de testes foi realizada para garantir o funcionamento adequado do sistema antes de prosseguir com testes mais abrangentes.

No estudo E_2 , a validação do framework envolveu o ChitChatBot, um agente de conversação desenvolvido pelos autores para discussões informais. O objetivo do *framework* era testar a qualidade das conversas dos agentes usando diferentes tipos de entradas. Foram criados exemplos divergentes para cada uma das 48 sentenças corretamente classificadas, e o desempenho do ChitChatBot nesses exemplos foi avaliado em comparação com suas respostas às entradas originais. No estudo E_5 , a validação foi realizada por meio de experimentos com sete modelos de classificação de texto baseados em transformadores: BERT, DistilBERT, DistilRoBERTa, RoBERTa, XLM-RoBERTa, Logistic Regression e Random Forest. Esses modelos foram treinados e avaliados em um conjunto de dados de validação.

No estudo E_6 , a abordagem foi validada por meio de experimentos com um *chatbot* de reserva de hotel, usando diferentes conjuntos de teste e especificações PDDL. Os testes envolveram a execução dos casos gerados pela abordagem, abrangendo vários cenários e condições. Durante a execução, a interação foi monitorada para avaliar o comportamento do sistema. Os conjuntos de teste foram criados com três abordagens: 1) uma única informação por solicitação, com ações repetitivas e valores constantes; 2) várias informações por solicitação, incluindo solicitações subsequentes com valores válidos e não intencionais; 3) mensagens de cancelamento adicionadas às especificações anteriores. A validação incluiu a comparação das respostas do *chatbot* com os resultados esperados. Os testes foram repetidos com diferentes conjuntos e especificações PDDL para garantir a robustez da abordagem.

No estudo E_{12} , a avaliação ocorreu por meio de um *chatbot* de turismo por meio de casos de teste gerados pela técnica de teste metamórfico (MT) e *metamorphic relations* (MRs). Utilizando o algoritmo **BotMorph**, foram executados casos de teste de origem (Is) e seus resultados (Os) foram registrados. Em seguida, MRs foram aplicadas para gerar casos de teste de acompanhamento (If) e testados no *chatbot*. Comparando as saídas resultantes (Of) com os casos de origem, a validação avaliou a consistência do sistema em diversas situações. Os resultados destacaram a eficácia da abordagem em detectar comportamentos inesperados, revelando tanto o desempenho satisfatório do *chatbot* em certas condições quanto suas falhas em outras.

No estudo E_{13} , a abordagem foi validada por meio de um experimento para responder questões de pesquisa sobre detecção de problemas nos *chatbots* e melhoria da qualidade. Os testes incluíram esturdiez, coesão e precisão, com diversos operadores de mutação. Cada *chatbot*, incluindo Baseline, Nutrition e RoomService, foi avaliado. Os testes de esturdiez aplicaram mutações em caracteres e números, revelando falhas em todos os *chatbots*. Uma abordagem de correspondência difusa foi empregada para melhorar os resultados. Na coesão, sem mutações, todos os *chatbots* passaram. Os testes de precisão envolveram mutações em palavras e idiomas, com falhas em alguns testes para todos os *chatbots*. Após treinamento adicional com casos de falha, os *chatbots* foram retestados, resultando em melhorias significativas na precisão.

2.3.8 Q_6 : DOMÍNIOS DOS *CHATBOTS*

Os domínios dos *chatbots* investigados nos estudos foram variados, abrangendo uma variedade de setores. No âmbito da pesquisa, foram analisados os seguintes domínios:

- **Hoteleria** (E_8 , E_6 , E_{12} e E_{13}), indicando um interesse em integrar *chatbots* nesse setor, possivelmente para melhorar serviços e interações com clientes.
- **Educação** (E_{10}), trouxe uma reflexão sobre como os *chatbots* podem ser implementados ou otimizados para melhorar processos educacionais.

2.4 CONSIDERAÇÕES FINAIS

Neste capítulo, foram apresentados os principais conceitos teóricos relacionados ao teste de software e agentes conversacionais, destacando suas aplicações, desafios e abordagens existentes na literatura. A revisão realizada permitiu compreender como as técnicas de teste vêm sendo aplicadas ao contexto dos chatbots, identificando lacunas e oportunidades para aprimoramento.

Os estudos analisados evidenciam que os chatbots, especialmente aqueles baseados em Processamento de Linguagem Natural (PLN), apresentam desafios únicos em termos de verificação e validação, devido à sua natureza não determinística e dependente do contexto da interação. Métodos tradicionais de teste podem não ser suficientes para garantir a robustez e confiabilidade desses sistemas, tornando abordagens alternativas, como os testes metamórficos, uma estratégia promissora.

Dessa forma, os conceitos discutidos ao longo deste capítulo servem como base para o desenvolvimento do Tesutã, um guia de apoio ao teste metamórfico em chatbots, detalhado no próximo capítulo. A partir da revisão da literatura e da identificação das técnicas e heurísticas disponíveis, a pesquisa avança para a proposição de uma abordagem sistemática que auxilie testadores na identificação de falhas e na melhoria da qualidade dos chatbots.

3 TESUTÃ: UM GUIA DE APOIO AO TESTE METAMÓRFICO EM *CHATBOTS*

Este capítulo apresenta o guia de apoio ao teste metamórfico em *chatbots* proposto no contexto dessa dissertação de Mestrado. Em particular, o problema de pesquisa é abordado na Seção 3.1. Posteriormente, será apresentada a metodologia de pesquisa que foi utilizada, conforme exposto na Seção 3.2. O guia será detalhado na Seção 3.3, delineando as estratégias e abordagens adotadas. Em seguida, um cenário de uso é mostrado na Seção 3.5, demonstrando na prática como um profissional de teste pode tomar decisões baseados no guia. Por fim, as considerações finais são apresentadas na Seção 3.6.

3.1 PROBLEMA DE PESQUISA

Os resultados da Revisão Sistemática da Literatura (RSL) (SANTOS et al., 2024) revelaram que ainda não existe uma definição concreta sobre como testar *chatbots*, mostrando quais as técnicas e ferramentas têm sido utilizadas na detecção de defeitos nesse cenário.

É importante destacar que os *chatbots* baseados em PLN são produtos de software particularmente difíceis de testar, pois uma entrada pode gerar N saídas e isso depende de como esse *chatbot* foi programado para interpretar a entrada do usuário (AQUINO; ADANIYA, 2018). Uma alternativa para resolver esse problema são as relações metamórficas, pois transcendem as limitações dos métodos convencionais de teste, uma vez que é possível realizar variações na entrada e testar a corretude da saída de acordo com as especificações desse software (BOŽLÍČ, 2022). Portanto, essa abordagem permite uma avaliação contextualizada das respostas, refletindo a adaptabilidade e o desempenho do software em diversos cenários.

3.2 MATERIAIS E MÉTODOS

Como descrito anteriormente, realizou-se uma RSL para compreender o estado da arte em que essa dissertação de Mestrado está inserida (SANTOS et al., 2024). Desta forma, observou-se que os testes metamórficos podem ser aplicados em *Chatbots* para revelar falhas, uma vez que *chatbots* podem ser difíceis de testar devido à complexidade da natureza de interação deste tipo de software.

Com o objetivo de desenvolver um modelo conceitual, heurísticas inovadoras e um guia prático para testes metamórficos em chatbots, além de avaliar sua eficácia e adaptabilidade, definiram-se os seguintes passos a serem executados:

- **Passo 1:** Análise do Estado da Arte: Foi conduzido uma RSL para identificar as principais técnicas, critérios e ferramentas para realizar a atividade de teste em *chatbots*. Além de mapear as limitações das abordagens de teste convencionais.
- **Passo 2:** 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.
- **Passo 3:** Desenvolvimento do *Tesutã*: foi desenvolvido um guia para apoiar os testadores a executar testes metamórficos em *chatbots*. Esse guia será um passo a passo de apoio aos testadores, contemplando desde a compreensão do contexto dos *chatbots* até a avaliação dos testes realizados.
- **Passo 4:** Avaliação do *Tesutã*: foi realizado um estudo de caso com os estudantes do curso de Ciências da Computação a fim de avaliar o *Tesutã*.

Na próxima seção é descrito o guia proposto nesse trabalho, denominado *Tesutã*.

3.3 DESCRIÇÃO DO TESUTÃ

A partir de uma análise detalhada sobre o estado da arte, foi observado que o teste em *chatbots* ainda tem sido pouco explorado, principalmente em relação a quais

técnicas ou critérios devem ser considerados (SANTOS et al., 2024). Desta forma, esta dissertação de Mestrado visa propor um guia, *step-by-step*, denominado *Tesutã*, para condução de teste metamórfico em *chatbots*. Este guia foi desenvolvido após analisar os estudos da RSL (E_8 e E_{12}), extraindo os pontos mais relevantes sobre testes metamórficos. Além disso, o guia foi dividido em etapas e, em cada etapa, são disponibilizadas heurísticas que servem de base para os profissionais. Além disso, os demais estudos serviram como base para formular as heurísticas para testar os *chatbots*.

O fluxo de execução do *Tesutã* pode ser visualizado na Figura 3. A figura possui três tipos de atividades e elas se repetem podendo ser opcionais, a saber: 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 contemplem o caso de teste almejado, é necessário voltar à fase de compreensão do *chatbot*).

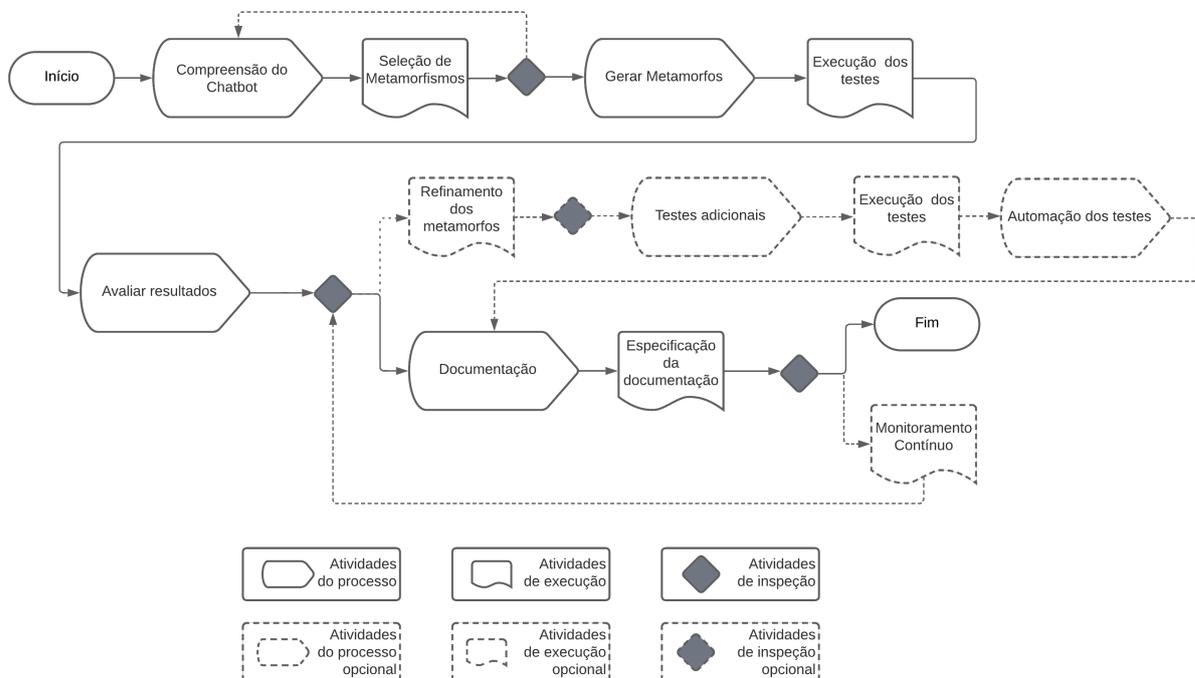


Figura 3: Fluxo do *Tesutã*

No primeiro passo, o testador realiza a atividade de 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, é necessário realizar a **seleção de metamorfismos**, ou seja, identificar quais são os metamorfos relevantes para o contexto do *chatbot*, ou seja, a partir das funcionalidades identificadas e/ou requisitos funcionais e não funcionais, o testador deverá elencar quais são passíveis de sofrer metamorfismo. 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 incluindo variações simulando diferentes interações.

Depois de seguir esses passos iniciais, o testador que está utilizando o *Tesutã* 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. 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, ou seja, para cada caso de teste que apresentar falha, é recomendado que sejam inseridos outros tipos de metamorfismos para identificar possíveis comportamentos correlacionados. 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 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, permitindo a detecção rápida de problemas após atualizações ou alterações no *chatbot*.

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

3.4 HEURÍSTICAS DE APOIO À APLICAÇÃO DO *TESUTĀ*

Para utilizar o *Tesutā* é fundamental integrar heurísticas para oferecer suporte aos passos delineados. As heurísticas, baseadas em conhecimento especializado e experiência prática, servem como diretrizes úteis para os testadores durante a execução das diferentes etapas do processo de teste. Elas ajudam 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, a seguir, apresentam-se as heurísticas de apoio à aplicação do *Tesutā*.

- **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, levante 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 comprimentos 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 teste 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 identificados¹;

- **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*² ou *bug fix*³, deve ser realizada uma nova verificação do caso de teste.

A partir de tais 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. O Anexo B.1 apresenta um exemplo de aplicação das heurísticas utilizando o ChatGPT.

3.5 CENÁRIO DE USO

Para apoiar o entendimento do uso do *Tesutã*, a seguir é apresentado um cenário de uso do guia proposto.

¹A forma de documentação pode variar de empresa para empresa

²Feature: Funcionalidade ou recurso novo adicionado ao *software*

³Bug fix: Correção de um problema/defeito em um *software*

Kratos, um profissional especializado em testes, foi designado para avaliar a robustez do chatbot GPT-4⁴ em um ambiente de suporte ao cliente de uma grande empresa de tecnologia. No entanto, Kratos não conhece técnicas ou meios de realizar teste neste tipo de *software*. Então, ele recebeu um guia abrangente sobre a técnica de teste metamórfico e está entusiasmado para aplicar esse conhecimento no contexto do chatbot GPT-4. Para iniciar o processo, Kratos mergulha na compreensão do chatbot GPT-4, estudando sua documentação e interagindo com o sistema para entender suas funcionalidades. Ele identifica os comportamentos esperados em situações de suporte ao cliente, como responder a perguntas sobre produtos, resolver problemas e fornecer informações sobre políticas da empresa.

Utilizando a primeira heurística (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), Kratos verifica os requisitos funcionais e não funcionais do GPT-4 e analisa quais são relevantes para o contexto de suporte ao cliente. Ele também consulta os registros de interações passadas com usuários para entender os tipos de perguntas mais comuns e as respostas típicas fornecidas pelo *chatbot*. A partir desse entendimento, Kratos seleciona cuidadosamente os metamorfismos mais relevantes para testar o GPT-4 no contexto de suporte ao cliente. Ele considera diferentes tipos de interações, como variações de perguntas, solicitações de informações específicas e problemas técnicos simulados.

Ao selecionar os metamorfismos, Kratos garante que as funcionalidades principais do GPT-4 sejam abordadas, como compreensão de linguagem natural, capacidade de fornecer respostas úteis e precisas, e comportamento consistente em diferentes cenários de suporte ao cliente. Após selecionar os metamorfismos, Kratos gera conjuntos de casos de teste, aplicando sistematicamente as variações identificadas. Ele cria casos de teste que refletem as diferentes situações de suporte ao cliente, incluindo perguntas simples, consultas complexas e problemas técnicos.

Durante a execução dos testes, Kratos observa atentamente as respostas geradas pelo GPT-4 para cada conjunto de metamorfismos. Ele registra qualquer

⁴GPT-4: <https://https://chat.openai.com>

desvio ou comportamento inesperado, comparando os resultados obtidos com as expectativas definidas anteriormente. Ao encontrar variações que o GPT-4 não lida adequadamente, Kratos ajusta os metamorfismos conforme necessário e refaz os testes. Ele documenta detalhadamente os metamorfismos utilizados, os resultados obtidos e quaisquer problemas encontrados, garantindo uma base sólida para a comunicação com a equipe de desenvolvimento e futuras iterações do GPT-4.

Ao seguir as heurísticas propostas e aplicá-las de forma adaptada ao contexto do chatbot GPT-4, Kratos está preparado para contribuir significativamente para a melhoria contínua do sistema e assegurar uma experiência consistente para os usuários finais no suporte ao cliente.

3.6 CONSIDERAÇÕES FINAIS

Este capítulo apresentou em detalhes o guia proposto nesta dissertação de mestrado, denominado Tesutã, um guia para a execução de testes metamórficos em *chatbots*. O Tesutã fornece heurísticas que auxiliam os profissionais de teste na geração de testes metamórficos, com o objetivo de identificar defeitos que possam estar na origem das falhas.

Para a criação do Tesutã, utilizaram-se diferentes fontes de informações, e para garantir sua viabilidade e alcançar o objetivo proposto, foi realizado uma avaliação com estudantes na Universidade Federal do Amazonas (UFAM), a qual é detalhada no Capítulo 4.

4 AVALIAÇÃO DO TESUTÃ

Este capítulo apresenta o estudo realizado para verificar a efetividade do Tesutã na criação de testes metamórficos para chatbots. Na Seção 4.1 é apresentado o planejamento do estudo, abrangendo as questões de pesquisa, os objetivos definidos, os critérios de seleção dos participantes e os instrumentos empregados. A Seção 4.2 detalha os procedimentos realizados durante a execução do estudo. A análise dos resultados é apresentada na Seção 4.3, com enfoque no oráculo considerado, nos dados demográficos dos participantes e nas métricas de desempenho. Em complemento, a Seção 4.4 apresenta os resultados obtidos a partir da condução do estudo. A Seção 4.5 aprofunda a análise da qualidade dos casos de teste, enquanto a Seção 4.6 discute as ameaças à validade, destacando potenciais limitações e vieses do estudo. Por fim, as considerações finais são discutidas na Seção 4.7.

4.1 PLANEJAMENTO

A fim de verificar o entendimento e a efetividade do Tesutã, uma avaliação foi conduzida seguindo o planejamento proposto a seguir. O planejamento executado contém as questões de pesquisa, objetivos, seleção dos sujeitos, instrumentação e ameaças à validade.

4.1.1 QUESTÕES DE PESQUISA (QP)

A seguir apresentam-se as questões de pesquisa proposta no estudo para verificar o entendimento e a efetividade do Tesutã.

- QP_1 : O Tesutã é fácil de entender com relação a sua abrangência, mostrando instruções claras e concisas sob o ponto de vista dos estudantes?
- QP_2 : O Tesutã apresenta é efetivo para gerar testes metafóricos em *chatbots*?

4.1.2 OBJETIVO

A seguir, apresenta-se o objetivo do estudo realizado neste projeto de mestrado de acordo com o método GQM (do inglês, *Goal Question Metric*) (SOLINGEN et al., 2002):

Analisar o Tesutã

Com o propósito de avaliá-lo

Com respeito ao seu entendimento e efetividade

Do ponto de vista dos estudantes

No contexto da disciplina de Verificação, Validação e Teste de Software.

4.1.3 SELEÇÃO DOS PARTICIPANTES

Para a condução deste estudo, foram convidados estudantes de graduação em Ciência da Computação da Universidade Federal do Amazonas (UFAM). Os participantes estavam cursando a disciplina de Verificação, Validação e Teste de Software.

É importante ressaltar que todos os participantes aceitaram participar do estudo de forma voluntária. Além disso, eles tiveram que:

- Assistir a aula gravada sobre testes metamórficos. Os slides dessa gravação que podem ser consultados no Anexo C;
- Consultar a wiki desenvolvida para o guia¹;
- Realizar uma atividade utilizando o Tesutã para conduzir testes metamórficos;
- Preencher um formulário de caracterização de perfil; e
- Responder o questionário de *Feedback*.

¹Wiki do guia Tesutã: <https://tsuta-guide.vercel.app>

4.1.4 INSTRUMENTAÇÃO

A seguir, apresenta-se uma breve descrição dos materiais utilizados para condução deste estudo, os quais estão disponíveis nos Anexos C, D e E, sendo eles:

- **Aula sobre testes metamórficos:** Foi ministrada uma aula para os estudantes sobre testes metamórficos, abordando, os principais tópicos desse conteúdo que envolvem a definição, suas propriedades, vantagens e desvantagens, bem como um passo a passo para aplicá-lo.
- **Termo de Consentimento Livre e Esclarecido (TCLE):** Os estudantes assinaram um termo manifestando seu interesse em participar do estudo de forma voluntária. O termo foi dividido em três partes: i) visão geral; ii) confidencialidade; e iii) benefícios e liberdade de desistência.
- **Chatbot ELIZA:** Foi utilizado o *chatbot* ELIZA para realização da atividade com os estudantes. Esse *chatbot* está disponível para a plataforma WEB².
- **Oráculo:** O oráculo utilizado neste estudo foi elaborado com base em um conjunto de requisitos e relações metamórficas previamente definidos e validados. Ele serve como referência para comparar com as relações metamórficas fornecidas pelos estudantes durante as atividades realizadas. A construção do oráculo envolveu uma análise detalhada do problema abordado e a identificação das relações metamórficas adequadas para sua solução. Esse recurso é essencial para avaliar a precisão, a completude e a consistência das respostas dos participantes, permitindo uma validação objetiva dos resultados obtidos. A descrição completa do oráculo pode ser encontrada no Anexo F.1. A versão simplificada é apresentada na Subseção 4.1.5.
- **Questionário de *Feedback*:** O Questionário de *Feedback* foi elaborado para coletar a percepção dos participantes sobre o Tesutã, abrangendo diferentes aspectos de sua experiência. Inicialmente, o questionário apresenta um Termo de Consentimento, no qual os participantes devem autorizar a utilização de seus dados na pesquisa. Em seguida, há perguntas voltadas para a caracterização

²Chatbot Eliza: <https://www.masswerk.at/eliza>

do perfil dos participantes, incluindo informações sobre formação acadêmica, experiência com desenvolvimento de software, familiaridade com chatbots e testes de software.

A avaliação do Tesutã é realizada por meio de questões que analisam a clareza e usabilidade do guia, investigando a facilidade de compreensão, a utilidade das heurísticas propostas e a aplicabilidade das instruções fornecidas. Além disso, há um espaço destinado a sugestões e críticas, permitindo que os participantes relatem pontos positivos, dificuldades encontradas e recomendações para aprimoramento do guia.

Por fim, o questionário inclui uma pergunta sobre o nível de satisfação geral dos participantes com o Tesutã e sua recomendação para outros profissionais. Esse questionário desempenha um papel fundamental na validação do guia, fornecendo insights sobre sua eficácia e identificando oportunidades de melhoria com base no *feedback* real dos usuários.

- **Métricas:** A fim de validar a corretude e a completude das atividades realizadas pelos estudantes, foram estabelecidas métricas específicas para avaliar os requisitos e as relações metamórficas identificadas por eles. O processo de avaliação consiste em comparar as respostas dos estudantes com um oráculo previamente definido, que serve como referência para verificar a precisão e abrangência dos resultados obtidos. As métricas utilizadas podem ser consultadas na Subseção 4.1.6.

4.1.5 ORÁCULO

O oráculo utilizado neste estudo foi elaborado como padrão de referência para avaliar a corretude e a completude dos requisitos e relações metamórficas identificados pelos participantes. Ele foi projetado com base em uma análise detalhada do problema abordado, considerando os principais aspectos funcionais e metamórficos esperados na atividade. Nas próximas subseções são mostrados os requisitos funcionais e as relações metamórficas definidos no oráculo.

4.1.5.1 REQUISITOS FUNCIONAIS IDENTIFICADOS

Os requisitos funcionais estabelecem as principais capacidades esperadas do *chatbot* ELIZA, conforme descrito na Tabela 5.

Tabela 5: Requisitos funcionais identificados no oráculo.

ID	Requisitos Funcionais Identificados
RF1	A Eliza deve ser capaz de interagir por texto.
RF2	A Eliza deve ser capaz de interagir por voz.
RF3	A Eliza deve ser capaz de realizar análise de texto (PLN).
RF4	A Eliza deve simular uma sessão de terapia (o processamento de texto não deve fugir de seu domínio).
RF5	A Eliza deve ser capaz de realizar gerenciamento de contexto (usuário modifica a linha de raciocínio).
RF6	A Eliza deve identificar palavras-chave no prompt do usuário e responder de forma coesa.

4.1.5.2 RELAÇÕES METAMÓRFICAS IDENTIFICADAS

As relações metamórficas definem padrões esperados de transformação que garantem a validade dos testes realizados. As relações estão descritas na Tabela 6.

Tabela 6: Relações metamórficas identificadas no oráculo.

ID	Relações Metamórficas Identificadas
RM1	Parafrase: mantém o mesmo significado emocional, alterando a forma das palavras, mas preservando o contexto da mensagem.
RM2	Substituição: altera um elemento específico por outro, mantendo o contexto e o significado geral da mensagem.
RM3	Exclusão: remove um elemento sem alterar o sentido central da mensagem ou função original, destacando o impacto da ausência.
RM4	Adição: inclui um elemento extra, preservando o significado ou função, mas oferecendo uma perspectiva ampliada ou detalhada da mensagem original.
RM5	Escalonamento: modifica proporcionalmente valores ou intensidades em um padrão de escala, como multiplicar ou dividir, mantendo a estrutura relativa entre os elementos originais.
RM6	Permutação: altera a ordem dos elementos sem mudar o contexto ou a mensagem final, apenas reorganizando a sequência.
RM7	Inversão: reverte a ordem ou direção de elementos, criando uma disposição contrária, mas mantendo o significado central ou a relação entre eles.

4.1.6 MÉTRICAS

Nesta subseção são apresentadas as métricas utilizadas para avaliar as atividades realizadas pelos estudantes. As métricas analisam a corretude e a completude das respostas, além de fornecer informações sobre os requisitos e relações metamórficas identificadas.

4.1.6.1 CORRETUDE DOS CASOS DE TESTE IDENTIFICADOS PELOS PARTICIPANTES

A métrica de **corretude** dos casos de teste foi definida com base na relação entre o número de casos de teste com resultados corretos, conforme identificado pelos estudantes durante a aplicação do estudo, e o total de casos de teste analisados. A métrica é apresentada a seguir:

Seja n o número total de participantes, C_i a quantidade de casos de teste identificados pelo participante i , e C_i^{corretos} a quantidade de casos de teste com resultados corretos identificados pelo mesmo participante. A corretude total (CT) é expressa como:

$$CT = \frac{\sum_{i=1}^n C_i^{\text{corretos}}}{\sum_{i=1}^n C_i} \times 100$$

A fórmula é descrita por:

1. **Numerador** ($\sum_{i=1}^n C_i^{\text{corretos}}$): Soma do número de casos de teste corretos identificados por todos os participantes.
2. **Denominador** ($\sum_{i=1}^n C_i$): Soma do total de casos de teste identificados por todos os participantes.
3. **Multiplicação por 100**: Transforma o valor em uma porcentagem, facilitando a interpretação do resultado.

Dessa forma, a métrica representa a proporção de casos de teste corretos em relação ao total identificado, fornecendo uma visão quantitativa da precisão dos participantes ao elaborar os casos de teste.

4.1.6.2 COMPLETUDE DOS REQUISITOS IDENTIFICADOS PELOS PARTICIPANTES

A métrica de completude total foi definida com base na relação entre o número de requisitos cobertos pelos participantes e o total de requisitos identificados. A métrica é apresentada a seguir:

Seja n o número total de participantes, R_i a quantidade de requisitos identificados pelo participante i , e R_i^{cobertos} a quantidade de requisitos cobertos pelo mesmo participante. A completude total (CP) é expressa como:

$$CP = \frac{\sum_{i=1}^n R_i^{\text{cobertos}}}{\sum_{i=1}^n R_i} \times 100$$

A fórmula é descrita por:

1. **Numerador** ($\sum_{i=1}^n R_i^{\text{cobertos}}$): Soma do número de requisitos cobertos pelos participantes.
2. **Denominador** ($\sum_{i=1}^n R_i$): Soma do total de requisitos identificados pelos participantes.
3. **Multiplicação por 100**: Transforma o valor em uma porcentagem, facilitando a interpretação do resultado.

Dessa forma, a métrica representa a proporção de requisitos cobertos pelos participantes em relação ao total identificado, permitindo avaliar a abrangência da análise realizada.

4.1.6.3 COMPLETUDE QUANTO AOS REQUISITOS DO ORÁCULO

A completude quanto aos requisitos do oráculo avalia a proporção de requisitos identificados pelos participantes em relação ao total de requisitos definidos no oráculo.

Desta forma, para obter a porcentagem geral da completude quanto aos requisitos do oráculo, foi realizado o cálculo da média das porcentagens individuais de cada participante. As fórmulas utilizadas estão descritas abaixo:

$$C_i = \frac{S_i}{T} \times 100 \quad (1)$$

Onde:

- C_i : Porcentagem de completude individual do participante i ;
- S_i : Número de requisitos do oráculo identificados pelo participante i ;
- T : Total de requisitos definidos no oráculo.

A porcentagem geral de completude é obtida por meio da média das completudes individuais:

$$C_{\text{geral}} = \frac{\sum_{i=1}^n S_i}{n \cdot T} \times 100 \quad (2)$$

Onde:

- C_{geral} : Porcentagem geral de completude;
- n : Número total de participantes;
- $\sum_{i=1}^n S_i$: Soma do número de requisitos identificados por todos os participantes.

4.1.6.4 COMPLETUDE QUANTO ÀS RELAÇÕES METAMÓRFICAS DO ORÁCULO

A completude quanto às relações metamórficas do oráculo avalia a proporção de relações metamórficas identificadas pelos participantes em relação ao total de relações metamórficas definidas no oráculo.

Desta forma, para obter a porcentagem geral da completude quanto às relações metamórficas do oráculo, foi realizado o cálculo da média das porcentagens individuais de cada estudante. As fórmulas utilizadas estão descritas a seguir:

$$C_{\text{RM},i} = \frac{R_i}{R_T} \times 100 \quad (3)$$

Onde:

- $C_{\text{RM},i}$: Porcentagem de completude individual quanto às relações metamórficas do participante i ;
- R_i : Número de relações metamórficas do oráculo identificadas pelo participante i ;
- R_T : Total de relações metamórficas definidas no oráculo.

A porcentagem geral de completude é obtida por meio da média das completudes individuais:

$$C_{\text{RM,geral}} = \frac{\sum_{i=1}^n R_i}{n \cdot R_T} \times 100 \quad (4)$$

Onde:

- $C_{RM,geral}$: Porcentagem geral de completude quanto às relações participantes;
- n : Número total de participantes;
- $\sum_{i=1}^n R_i$: Soma do número de relações metamórficas identificadas por todos os participantes.

4.2 EXECUÇÃO

A próxima etapa foi a execução do estudo, realizado em 06 de Junho de 2024 de forma online. A seguir são apresentados os detalhes para a condução desse estudo.

Em relação à quantidade de participantes, 39 estudantes participaram da aula que foi ministrada de forma remota e também foi disponibilizada virtualmente para os demais alunos que não puderam comparecer. Durante a aula, os participantes tiveram acesso aos testes metamórficos e problemas matemáticos típicos do curso de Ciência da Computação como: caixeiro viajante, funções matemáticas e busca binária. Em seguida, foram apresentadas as vantagens e desvantagens da utilização deste tipo de teste. Por fim, foi apresentado o seguinte problema: dado o conteúdo exposto, em um sistema meteorológico de previsão do tempo, como pode-se aplicar testes metamórficos?. Pensando nisso, foi desenvolvido em sala (*live code*) um algoritmo em Python³ para verificar se a previsão do tempo para uma data futura muda de forma consistente quando as condições iniciais são ajustadas ligeiramente.

Dessa forma, o código proposto foi desenvolvido utilizando a WeatherAPI⁴ para obter a previsão do tempo em uma cidade específica. Para montar um teste metamórfico para esse código foi implementada a seguinte lógica: dado um conjunto de cidades próximas, qual a consistência das previsões? Ou seja, para cada previsão é realizada a comparação da temperatura média entre as cidades; se a diferença for maior que um limite arbitrário⁵ haverá uma inconsistência. Desta forma, os alunos

³Algoritmo: <https://github.com/gabrielsants/weather-metamorphic-testing>

⁴WeatherAPI: <https://www.weatherapi.com>

⁵O limite arbitrário neste caso é a relação metamórfica que irá variar

aprenderam, na prática, diversos conceitos antes de entrar, de fato, no aspecto de chatbots.

Em seguida, foram apresentadas as instruções para a atividade prática, organizadas em quatro passos:

- **Acessar a *wiki* do Tesutã** que contém as instruções básicas sobre a wiki do guia e como acessá-lo;
- **Apresentação do *chatbot* testado**, mostrando uma introdução ao *chatbottestado*, assim como história, desenvolvimento e legado;
- **Teste do *chatbot* ELIZA** utilizando o Tesutã;
- **Avaliação do Tesutã** por meio do preenchimento de um questionário contendo caracterização e avaliação do Tesutã.

4.3 ANÁLISE DOS RESULTADOS

Essa seção apresenta a análise detalhada dos resultados obtidos durante a avaliação do Tesutã. O objetivo desta análise é fornecer uma compreensão aprofundada dos dados coletados, considerando tanto aspectos quantitativos quanto qualitativos. São discutidos os principais achados em relação ao desempenho dos participantes, à identificação de falhas e à aplicação das heurísticas propostas pelo Tesutã. Além disso, são abordados aspectos relacionados ao oráculo utilizado, aos dados demográficos dos participantes e às métricas de desempenho, a fim de contextualizar e interpretar adequadamente os resultados obtidos. Essa análise é essencial para compreender a efetividade do Tesutã e identificar possíveis melhorias para garantir sua aplicabilidade em diferentes cenários de teste em chatbots.

4.4 RESULTADOS

A seção a seguir apresenta os resultados obtidos a partir do estudo conduzido para avaliar o Tesutã. Os dados coletados durante o estudo foram analisados de forma a fornecer uma visão clara sobre a eficácia do Tesutã na criação de testes

metamórficos para chatbots. São discutidos os resultados quantitativos, como a completude dos requisitos identificados pelos participantes, bem como as variações no desempenho entre os diferentes sujeitos do estudo. Além disso, os resultados qualitativos são explorados para entender as percepções dos participantes sobre a clareza e aplicabilidade das heurísticas apresentadas. A seção também aborda as métricas de desempenho, oferecendo uma visão geral dos pontos fortes e das áreas que podem ser aprimoradas no Tesutã, com base nas evidências coletadas durante a avaliação.

4.4.1 DADOS DEMOGRÁFICOS

Todos os participantes aceitaram participar do estudo de forma voluntária, assinando o TCLE que mostrava uma visão geral do estudo e a confidencialidade dos dados coletados. A partir do formulário de caracterização de perfil, foi identificado o perfil dos sujeitos. A Figura 4 visa responder à questão: “Qual sua experiência com desenvolvimento de software?” e ilustra as porcentagens dos níveis de conhecimento dos participantes em relação ao teste de software. Todos os participantes têm alguma experiência com programação, sendo que 38.5% têm experiência profissional, 51.3% possuem experiência em disciplinas de programação e 10.2% consideram ter pouca experiência.

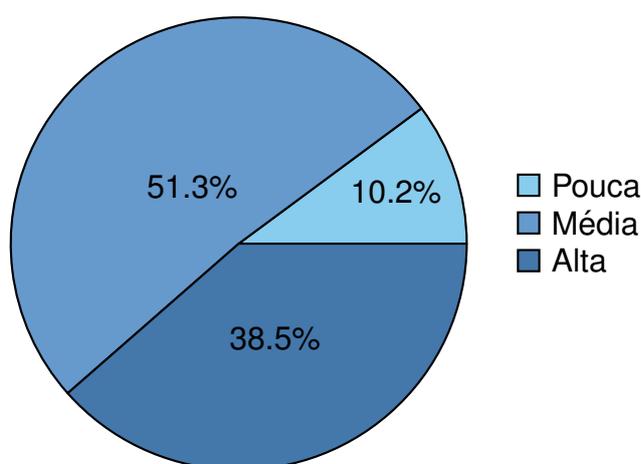


Figura 4: Experiência dos participantes em relação ao desenvolvimento de software

A Figura 5 apresenta a experiência dos participantes quanto à interação com

chatbots. Todos os participantes já tiveram contato pelo menos uma vez, sendo que 66.7% dos participantes consideram que têm alta experiência, enquanto 20.5% acreditam que possuem baixa experiência.

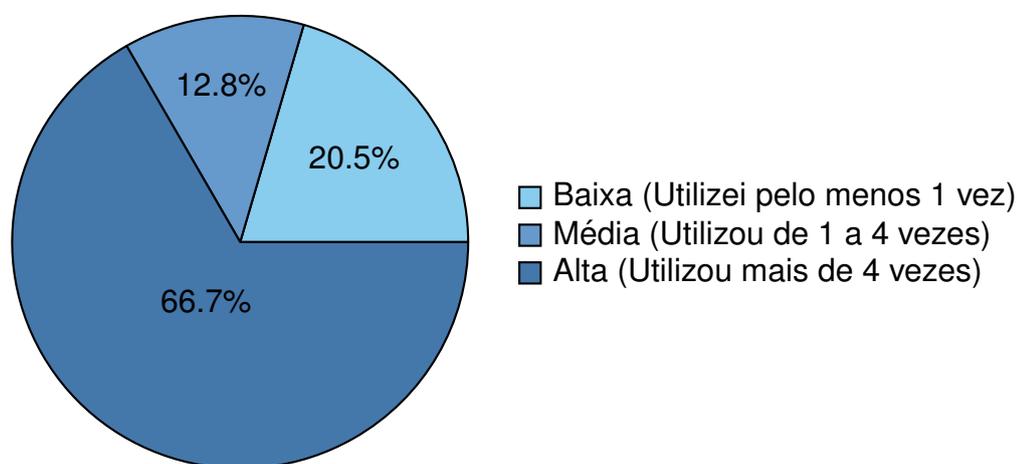


Figura 5: Qual sua experiência com Chatbots?

Dos 39 participantes (76,9%) são do sexo masculino, 20,5% do sexo feminino. Considerando que o questionário foi aplicado para uma turma de graduação, não houve grande diversificação na faixa etária, sendo 87,2% dos participantes têm entre 18 e 25 anos. Da mesma forma, o nível de escolaridade foi de 92,3% para aqueles que estão cursando graduação.

Em relação à experiência com chatbots, observou-se que 89,7% dos participantes já presenciaram situações envolvendo *chatbots* que consideram erros ou falhas. Além disso, 50% acreditam que essas situações poderiam ser evitadas com testes adequados, enquanto 47,4% demonstraram incerteza quanto a essa possibilidade.

Os participantes pontuaram cada afirmação do formulário de avaliação do guia utilizando a escala *Likert* de 0 a 10, na qual 0 representa “Discordo Totalmente” e 10 corresponde a “Concordo Totalmente”. Os valores intermediários indicam diferentes graus de concordância, sendo que 5 representa uma posição neutra, enquanto valores próximos a 0 refletem maior discordância e valores próximos a 10 indicam maior concordância. Desta forma, é possível obter um panorama sobre o ponto de vista dos participantes. A seguir são mostradas as respostas para as questões utilizadas para avaliar o Tesutã.

4.4.2 RQP_1 : O TESUTÃ É FÁCIL DE ENTENDER COM RELAÇÃO A SUA ABRANGÊNCIA, MOSTRANDO INSTRUÇÕES CLARAS E CONCISAS SOB O PONTO DE VISTA DOS ESTUDANTES?

Nas Figuras 6 e 7 são apresentados os resultados relacionados à importância dos *chatbots* e o quanto as heurísticas propostas auxiliaram no entendimento do guia. Sendo assim, 89,7% dos participantes acreditam que os *chatbots* são importantes e muito importantes e 94,8% afirmaram que as heurísticas propostas auxiliaram no entendimento do guia.

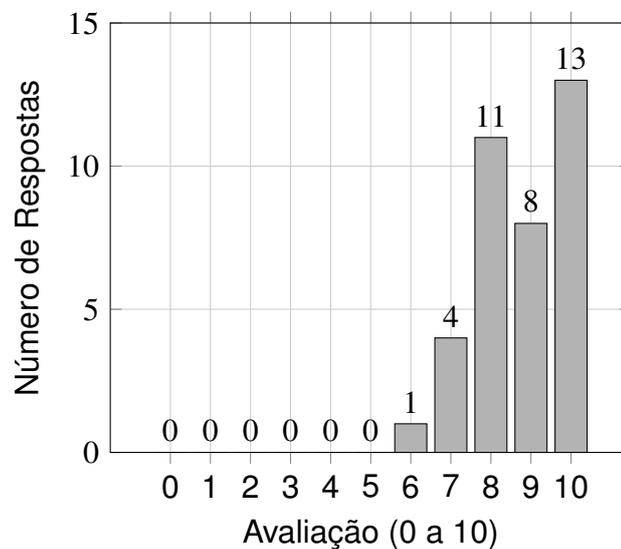


Figura 6: Avaliação da importância dos Chatbots (Escala de 0 a 10)

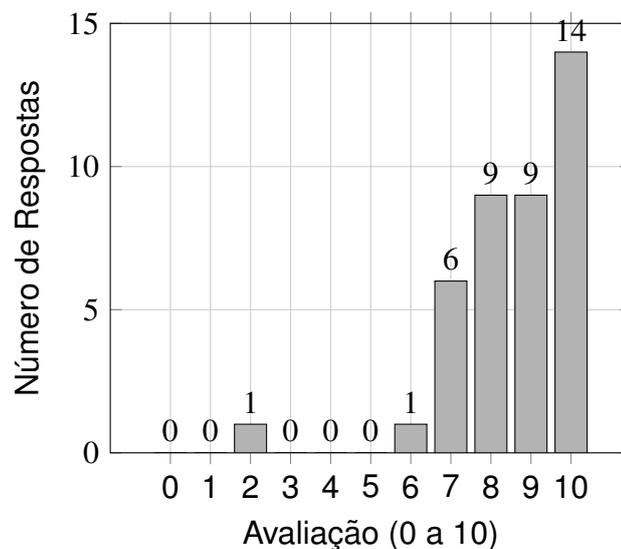


Figura 7: Avaliação da contribuição das heurísticas no entendimento do guia (Escala de 0 a 10)

Os participantes apresentaram diferentes percepções quanto à clareza e compreensão do guia, sendo que 92,3% concordaram que o guia é de fácil entendimento. Apesar de alguns indicarem que o guia não era difícil de entender e considerarem as heurísticas fundamentais para orientar o usuário a alcançar conclusões precisas, outros relataram dificuldades. O participante P4 mencionou que *“as descrições de algumas poucas heurísticas eram um pouco controversas em relação ao exemplo do ChatGPT (que explicou muito melhor as coisas)”*, enquanto o P19 destacou que *“a quantidade de heurísticas torna o processo cansativo e maçante”*. Além disso, houve comentários sobre a falta de exemplos claros e específicos, como mencionado por P29: *“não sabia claramente como utilizar o guia precisamente, pois tinha pouco exemplo do que precisava fazer precisamente”*. Essas observações indicam que, embora o guia seja considerado útil, a falta de exemplos detalhados e a extensão das heurísticas podem dificultar seu uso eficaz em certos contextos.

4.4.3 *QP₂*: O TESUTÃ APRESENTA UMA BOA EFETIVIDADE PARA GERAR TESTES METAFÓRFICOS PARA CHATBOTS?

Ao serem questionados sobre quais partes do Tesutã foram mais úteis, 42% dos participantes destacaram as heurísticas e a *wiki* como os pontos de maior relevância. Além disso, 24% mencionaram os cenários de uso e os exemplos práticos, e 18% ressaltaram que todo o Tesutã foi útil. As seções introdutórias foram apontadas como importantes por 12% dos participantes, e 6% destacaram a organização e clareza do guia. Por fim, outros aspectos específicos também foram mencionados por 9% dos respondentes. O participante P4 observou que *“se faltasse alguma parte naquele guia, muita gente poderia ficar perdida”*, reforçando a percepção de que cada seção contribui significativamente para a compreensão geral. Além disso, a descrição detalhada do ELIZA e a exemplificação prática com o ChatGPT foram frequentemente citadas como elementos que facilitaram o entendimento dos conceitos apresentados.

As heurísticas foram mencionadas como particularmente úteis, não apenas por detalhar o processo de teste, mas também por oferecerem uma visão clara sobre o funcionamento dos chatbots. A combinação de uma estrutura bem organizada,

exemplos práticos e explicações aprofundadas tornou o Tesutã um mecanismo valioso para a maioria dos participantes.

Em relação à recomendação do Tesutã para outras pessoas, 76,9% dos participantes indicaram que recomendaria sua utilização, refletindo a percepção positiva sobre sua utilidade e clareza. Esse consenso sugere que o Tesutã é amplamente considerado eficaz para a compreensão e aplicação dos conceitos abordados. No entanto, 20,5% dos participantes manifestaram incerteza, sugerindo que, apesar de reconhecerem a qualidade do Tesutã, podem haver aspectos que ainda precisam ser ajustados ou melhorados para garantir uma recomendação universal. São exemplos de melhorias propostas: aperfeiçoar a explicação das heurísticas, embora elas tenham sido amplamente mencionadas como úteis, uma maior clareza e aprofundamento nos exemplos poderia ajudar a reduzir as incertezas de alguns usuários; aplicação prática dos conceitos, como testes metamórficos e cenários de uso, já é valorizada, mas investir em mais exemplos detalhados ou em estudos de caso poderia fortalecer ainda mais a compreensão e a aplicação desses conceitos; outros comentários sugerem que a organização atual já é eficaz, mas melhorias na navegação e na estrutura do conteúdo poderiam ajudar a tornar a experiência mais intuitiva para todos os tipos de usuários. Esses *feedbacks* são valiosos para identificar áreas de aprimoramento e garantir que o Tesutã atenda ainda melhor às necessidades de todos os usuários.

Em relação à abrangência do conteúdo do Tesutã, 89,7% dos participantes afirmaram que ele foi suficientemente completo, indicando que grande parte dos participantes considerou o material adequado para cobrir os tópicos necessários. Sobre a clareza e o nível de detalhamento das instruções, 97,4% dos respondentes concordaram que as orientações foram claras e detalhadas, o que sugere uma apresentação eficaz do conteúdo. Além disso, a documentação do Tesutã se mostrou útil para a compreensão das heurísticas, com 94,9% dos participantes relatando que ela facilitou o entendimento desses conceitos.

Ao serem questionados sobre informações desnecessárias ou redundantes no Tesutã, a maioria dos participantes indicou que não houve informações supérfluas. O participante P17 destacou que *“todas as informações contidas no guia foram diretas*

e *necessárias*". No entanto, alguns indicaram que algumas seções poderiam ser ajustadas para evitar sobrecarga de informações. Em particular, a contextualização sobre o ELIZA foi mencionada como excessiva para aqueles focados apenas na execução técnica dos testes - o que sugere uma interpretação incorreta da questão, já que o material de apoio disponibilizado sobre a ELIZA é a parte do Tesutã. Do mesmo modo, o participante P18 observou que, embora a história de ELIZA seja interessante, ela "*pode ser considerada um tanto excessiva*" para o propósito principal do Tesutã. Outras respostas mencionaram que algumas heurísticas pareciam muito semelhantes ou que os cenários de uso e a seção sobre o *chatbot* resultaram em certa redundância.

Em relação ao nível geral de satisfação com o Tesutã, a maioria dos participantes (76,9%) se declarou satisfeita com o material apresentado, enquanto 15,4% relataram estar muito satisfeitos, o que reflete um alto grau de aprovação. Apenas 2,6% dos respondentes se mantiveram neutros e 5,1% manifestaram insatisfação. Quanto a insatisfação, pode-se ressaltar alguns comentários dos participantes P38 e P24 : "*tempo de aprendizagem para certos usuários*" e "*acho que poderiam haver mais componentes visuais para exemplificações*". Esses comentários indicam que, embora o Tesutã tenha sido bem recebido pela maioria, ainda há margem para melhorias que possam aumentar a satisfação geral e minimizar eventuais pontos de insatisfação.

Os participantes identificaram diversos pontos positivos durante a utilização do Tesutã para auxiliar na identificação de defeitos por meio das relações metamórficas. Um dos aspectos mais ressaltados foi a simplificação do processo de teste, com respostas enfatizando a eficiência na identificação de estigmas e erros, além de fornecer uma abrangência maior nas transformações das saídas com base nas entradas. A clareza do conteúdo e a estrutura bem organizada também foram apontadas como vantagens significativas, permitindo que até mesmo pessoas sem um conhecimento técnico avançado compreendam o material. A tabela de heurísticas e os exemplos práticos, como a aplicação com o ChatGPT, foram amplamente elogiados por fornecerem orientações precisas sobre como realizar testes metamórficos de maneira sistemática.

Além disso, a documentação detalhada e a linguagem simples facilitaram a

execução dos testes, enquanto as diretrizes claras ajudaram a manter a consistência no processo. O participante P17 mencionou que o Tesutã *“oferece um método detalhado e organizado para usar metamorfismos em testes de chatbot”*, cobre desde a verificação dos requisitos até a documentação dos resultados. Essa estrutura bem definida permite uma análise completa e incentiva a melhoria contínua dos *chatbots* testados. A organização por heurísticas, a ênfase na rastreabilidade dos testes e a inclusão de exemplos específicos foram destacados como fatores que aumentam a eficiência e a compreensão do processo de teste, superando as limitações dos métodos tradicionais.

Em relação aos pontos negativos, um dos aspectos mais apontados foi a complexidade e o tempo de aprendizagem necessários, especialmente para usuários com pouca experiência em testes de software ou que não têm familiaridade com os conceitos abordados. A redundância de algumas heurísticas e a quantidade de informação sobre elas, distribuídas em diferentes partes do Tesutã, foram percebidas como fatores que dificultam a compreensão e o fluxo de aprendizado. Além disso, houve menções sobre a linguagem técnica e, em alguns casos, confusa, o que poderia ser melhorado com uma abordagem mais simplificada ou com a inclusão de mais exemplos práticos e visuais.

Outro ponto citado foi a falta de exemplos reais e de casos de uso mais diversos, o que poderia tornar a aplicação das heurísticas mais clara e acessível. A ausência de materiais em português também foi vista como uma barreira significativa para alguns usuários, além da limitação de ferramentas automatizadas e da documentação detalhada apenas em inglês. A complexidade da aplicação de metamorfismos e a necessidade de conhecimento prévio sobre automação e requisitos funcionais são desafios que podem dificultar a adoção do Tesutã, principalmente para profissionais de teste que são iniciantes.

Alguns participantes sentiram falta de mais conteúdo específico para determinadas heurísticas, como a H18, e mencionaram que o foco extensivo na introdução e contextualização do Tesutã tornou o processo mais cansativo e menos imediato. A dependência de conhecimento específico, a dificuldade na criação e automação de testes e a falta de clareza em determinados exemplos também foram

levantados como preocupações segundo os participantes. Esses pontos sugerem que, para uma aplicação mais eficaz, o Tesutã poderia ser melhorado com a inclusão de novos materiais didáticos, exemplos práticos e uma maior acessibilidade linguística e visual.

4.5 ANÁLISE DA QUALIDADE DOS CASOS DE TESTE

Essa seção apresenta a análise da qualidade dos casos de teste gerados pelos participantes com o auxílio do Tesutã. O objetivo desta análise é avaliar a eficácia das heurísticas e diretrizes propostas pelo Tesutã na criação de casos de teste completos e corretos. Serão examinados aspectos como a correção e a consistência dos casos de teste identificados, a relevância das relações metamorfológicas aplicadas, e a aderência dos participantes aos critérios estabelecidos pelo Tesutã. Além disso, serão discutidas as possíveis lacunas observadas na geração dos casos de teste e as implicações para a qualidade geral do processo de teste em chatbots. Essa análise visa fornecer uma compreensão detalhada da efetividade do Tesutã no suporte à criação de testes de alta qualidade, identificando possíveis melhorias e aprimoramentos.

4.5.1 CORRETUDE DOS CASOS DE TESTE IDENTIFICADOS PELOS PARTICIPANTES

A corretude total avalia a proporção de casos de teste que os participantes consideraram como corretos em relação ao total de casos de teste identificados, sem, no entanto, verificar a validade dos resultados por meio de um oráculo. A fórmula utilizada para este cálculo foi apresentada na subseção 4.1.6.1. O resultado da corretude total pode ser observado na Tabela 7.

Tabela 7: Resultados da Corretude Total.

Métrica	Valor
Número de casos de testes com resultados corretos	70
Total de casos de testes identificados	93
Porcentagem de corretude total	75,26%

Os resultados apresentados na Tabela 7 indicam que, dos 93 casos de teste identificados pelos participantes, 70 apresentaram resultados corretos, resultando em uma corretude total de 75,26%. Este valor demonstra que a maioria dos testes foi bem-sucedida, evidenciando a confiabilidade do *chatbot* ELIZA.

No entanto, os 24,74% correspondem a casos em que os resultados não foram corretos. Isso sugere a necessidade de investigar possíveis inconsistências ou limitações do sistema, como falhas em requisitos específicos, cenários não previstos ou limitações do *chatbot*. Além disso, deve-se considerar a experiência de cada participante na identificação dos casos de teste. Essa análise pode fornecer subsídios para melhorias no desempenho geral e na adequação do sistema aos requisitos estabelecidos.

Portanto, apesar dos resultados indicarem um desempenho considerável, há possibilidades para aprimoramentos que visem reduzir os casos de erro e alcançar uma corretude total ainda mais elevada.

4.5.1.1 COMPLETUDE DOS REQUISITOS IDENTIFICADOS PELOS ESTUDANTES

A completude total avalia a proporção de requisitos que os participantes consideraram como cobertos em relação ao total de requisitos identificados, sem, no entanto, verificar a cobertura por meio de um oráculo. Os resultados são apresentados na Tabela 8.

Tabela 8: Resultados da Completude Total.

Métrica	Valor
Número de requisitos cobertos	81
Total de requisitos identificados	107
Porcentagem de completude total	75,70%

Os resultados apresentados na Tabela 8 revelam que, dos 107 requisitos identificados, 81 foram cobertos, resultando em uma completude total de 75,70%. Esse índice demonstra um nível significativo de cobertura, indicando que a maior parte dos requisitos foi contemplada no processo de desenvolvimento ou avaliação.

Entretanto, aproximadamente 24,30% dos requisitos identificados não foram cobertos. Isso pode sugerir lacunas no escopo do chatbot, limitações nas implementações realizadas ou desafios na definição e priorização dos requisitos durante o planejamento. A análise mais detalhada desses casos pode ser fundamental para identificar melhorias, ampliar a cobertura e garantir maior alinhamento do sistema com as expectativas e necessidades inicialmente estabelecidas.

Assim, embora o valor obtido represente um progresso considerável, a busca por uma completude total mais alta é essencial para assegurar a consistência e a abrangência do chatbot, reduzindo os riscos de falhas e aumentando a confiança nas respostas.

4.5.1.2 REQUISITOS E RELAÇÕES METAMÓRFICAS ÚNICOS IDENTIFICADOS

Além das métricas de corretude e completude, foi avaliado o número de requisitos e relações metamórficas únicas identificados pelos estudantes, ou seja, requisitos e relações que não foram contemplados pelo oráculo. Os resultados são listados na Tabela 9.

Tabela 9: Requisitos e Relações Metamórficas Únicos Identificados.

Métrica	Valor
Número de requisitos únicos identificados pelos estudantes	48
Número de relações metamórficas únicas identificadas pelos participantes	23

Os resultados apresentados na Tabela 9 demonstram que os participantes conseguiram identificar 48 requisitos únicos e 23 relações metamórficas.

4.5.1.3 COMPLETUDE QUANTO AOS REQUISITOS DO ORÁCULO

O oráculo apresentado foi desenvolvido a partir da execução do Tsuta, com base nos resultados obtidos a partir do teste conduzido e disponibilizado no Anexo F.1. Para uma análise mais detalhada dos procedimentos adotados e dos resultados, consulte o Anexo F.1, que contém informações complementares e específicas sobre a metodologia aplicada.

A Tabela 10 apresenta os resultados de completude dos requisitos alcançados por cada participante. Observou-se uma variação significativa nos valores, com o participante P19 obtendo a maior taxa de completude (66,66%) e os participantes P1 e P20 registrando a menor taxa, com 0%. A média geral foi de 26,46%, indicando que, na maioria dos casos, os participantes conseguiram identificar menos de um terço dos requisitos esperados. Esse resultado aponta para possíveis dificuldades na interpretação ou aplicação dos conceitos abordados no Tesutã, o que pode refletir uma lacuna no entendimento ou na clareza dos requisitos apresentados.

Os resultados gerais, com médias relativamente baixas em ambas as métricas, sugerem a necessidade de revisão e aprimoramento no *design* do Tesutã, bem como nas estratégias de ensino empregadas para sua aplicação. Possíveis fatores que podem ter influenciado os resultados incluem, mas não se limitam a: (i) o material pode não ter sido suficientemente claro ou objetivo para os estudantes; (ii) os conceitos associados aos requisitos e relações podem necessitar de explicações mais detalhadas ou exemplos práticos; e (iii) os participantes podem não ter recebido preparação adequada para realizar a tarefa de forma eficaz.

Tabela 10: Resultados de Completude por Participante.

Participante	Completude Requisitos (%)	Completude Relações (%)
P1	00,00%	00,00%
P2	16,66%	14,28%
P3	33,33%	14,28%
P4	33,33%	14,28%
P5	50,00%	14,28%
P6	16,66%	28,57
P7	16,66%	14,28%
P8	50,00%	14,28%
P9	33,33%	14,28%
P10	16,66%	85,71%
P11	50,00%	42,85%
P12	33,33%	14,28%
P13	16,66%	28,57%
P14	33,33%	14,28%
P15	16,66%	14,28%
P16	16,66%	28,57%
P17	16,66%	28,57%
P18	16,66%	28,57%
P19	66,66%	00,00%
P20	00,00%	14,28%
P21	16,66%	28,57%
P22	33,33%	42,85%
P23	33,33%	00,00%
P24	16,66%	28,57%
P25	33,33%	28,57%
P26	33,33%	14,28%
P27	16,66%	28,57%
P28	00,00%	28,57%
P29	33,33%	28,57%
P30	16,66%	14,28%
P31	16,66%	14,28%
P32	16,66%	28,57%
P33	50,00%	14,28%
P34	16,66%	28,57%
P35	16,66%	00,00%
P36	33,33%	28,57%
Média Geral	26,46%	21,42%

4.5.1.4 COMPLETUE QUANTO ÀS RELAÇÕES METAMÓRFICAS DO ORÁCULO

A completude quanto às relações metamórficas do oráculo avalia a proporção de relações identificadas pelos participantes em relação ao total de relações definidas

no oráculo. A Tabela 10 apresentada na subseção anterior demonstra os resultados de completude para cada participante.

Desta forma, a completude das relações apresentou uma média geral de 21,42%, com valores variando entre 85,71% (participante P10) e 0% (participantes P1, P19 e P23). Apesar de alguns participantes demonstrarem um desempenho relativamente elevado, a maioria apresentou resultados inferiores a 30%, o que pode indicar que as conexões entre os requisitos e suas interdependências não foram plenamente compreendidas.

4.5.2 DISCUSSÃO

Os resultados obtidos a partir da avaliação do Tesutã revelam importantes *insights* sobre sua aplicação prática, indicando tanto seu potencial quanto as limitações em sua atual concepção. A análise das métricas demonstrou que a completude dos requisitos identificados pelos participantes variou significativamente, com uma média geral de 21,42%. Em casos específicos, como do participante P10, o desempenho foi notavelmente elevado, alcançando 85,71% de completude, enquanto outros participantes, como P1, P19 e P23, não conseguiram identificar nenhum requisito. Essa ampla variação sugere que o Tesutã apresenta uma efetividade dependente da experiência prévia e do entendimento dos participantes, o que levanta a necessidade de ajustes para torná-lo mais acessível.

A análise qualitativa dos casos bem-sucedidos, como o participante P10, evidencia que o Tesutã, quando bem compreendido, é capaz de apoiar a identificação de requisitos e relações metamórficas de forma eficaz. Esse sucesso está associado à capacidade do participante em interpretar as diretrizes do Tesutã e aplicá-las de forma estruturada, o que resultou na identificação de falhas relevantes no chatbot avaliado. Em contraste, os casos de insucesso, como os de P1 e P19, indicam dificuldades significativas em compreender conceitos fundamentais, como as relações metamórficas. Esses participantes relataram que, apesar de considerarem as instruções úteis, faltaram exemplos mais detalhados que facilitassem a aplicação prática das heurísticas.

Uma das principais dificuldades enfrentadas pelos participantes foi a identificação e o uso das relações metamórficas. Enquanto alguns participantes com maior experiência em testes de software conseguiram superar esse desafio, a maioria relatou dificuldade em associar as transformações propostas pelo Tesutã ao contexto específico dos chatbots. Isso indica que o Tesutã poderia ser aprimorado com explicações mais aprofundadas e exemplos mais variados, abrangendo diferentes cenários de uso e níveis de complexidade.

Os *feedbacks* coletados também apontaram percepções mistas sobre a clareza e a abrangência do Tesutã. Embora alguns participantes tenham elogiado a objetividade das instruções, outros consideraram que certas diretrizes careciam de detalhamento suficiente para auxiliar na tomada de decisões durante os testes. Essa discrepância sugere que a clareza do Tesutã é influenciada pela familiaridade dos usuários com conceitos avançados de teste, reforçando a necessidade de materiais suplementares, como tutoriais interativos ou exercícios práticos.

Os resultados demonstram que o Tesutã possui um bom potencial para melhorar a qualidade dos chatbots, especialmente em aspectos como a identificação de falhas relacionadas à compreensão de linguagem natural. No entanto, sua efetividade plena é limitada pela variabilidade no desempenho dos participantes, o que reflete a importância de refinar o Tesutã para torná-lo mais acessível e aplicável a diferentes perfis de usuários. Adicionalmente, a inclusão de exemplos mais específicos e contextualizados, bem como a oferta de treinamentos direcionados, pode ajudar a mitigar as dificuldades identificadas.

Embora o Tesutã tenha demonstrado ser um mecanismo promissor para apoiar teste em chatbots, os resultados também destacam a importância de sua adaptação contínua. A realização de novos estudos com grupos mais heterogêneos e a validação em contextos de aplicação variados são passos fundamentais para consolidar o Tesutã como um recurso robusto e amplamente aplicável. Esses ajustes são essenciais para garantir que ele não apenas atenda às necessidades de profissionais experientes, mas também sirva como uma abordagem de aprendizagem eficaz para aqueles com menor familiaridade em teste de software.

4.6 AMEAÇAS À VALIDADE

Esta seção apresenta as ameaças à validade identificadas na condução desse estudo. A subseção 4.6.1 apresenta a validade interna do estudo quanto aos fatores que podem influenciar os resultados. A subseção 4.6.2 apresenta a validade externa quanto à generalização dos resultados e a subseção 4.6.3 apresenta a validade de conclusão quanto à precisão das conclusões obtidas.

4.6.1 VALIDADE INTERNA

A validade interna está relacionada a fatores que podem influenciar os resultados do estudo, não relacionados às variáveis em análise. No caso desta dissertação de mestrado sobre o Tesutã, as ameaças à validade incluem:

- **Viés de Seleção:** Os participantes do estudo foram estudantes de graduação, especificamente do curso de Engenharia de Software, cursando a disciplina de Verificação, Validação e Teste de Software, o que pode não representar adequadamente a diversidade de testadores de chatbots, mesmo tendo um treinamento e contexto prévio. A fim de mitigar este viés, foram enviados convites para participar da pesquisa, porém, não obtivemos respostas. Além, do treinamento disponibilizado.
- **Familiaridade com a Ferramenta:** Os participantes tiveram uma aula sobre teste metamórfico e tiveram acesso à wiki do Tesutã para condução das atividades. Essa preparação prévia pode ter influenciado os resultados do estudo, levando a um melhor desempenho que nem sempre seria observado em outros cenários. Contudo, não foram apresentados cenários e dicas sobre o chatbot ELIZA.

4.6.2 VALIDADE EXTERNA

A validade externa refere-se à capacidade de generalizar os resultados do estudo para outras situações ou populações. As ameaças identificadas incluem: contexto limitado dos participantes, ou seja, os resultados são baseados em um

grupo específico de estudantes que cursam a disciplina de Verificação, Validação e Teste de Software, o que pode não ser representativo de outros grupos de testadores ou profissionais da indústria; cenário de teste controlado, ou seja, a avaliação do Tesutã foi realizada em um ambiente controlado de sala de aula, que pode não refletir os desafios e as condições encontradas em ambientes de teste reais. Conforme discutido anteriormente, foram enviados convites para participar da pesquisa, porém, não obtivemos respostas.

4.6.3 VALIDADE DE CONCLUSÃO

A validade de conclusão refere-se à precisão das conclusões baseadas nos dados do estudo. Sendo assim, o tamanho da amostra é uma possível ameaça pois o número de participantes pode ter sido limitado, reduzindo a robustez da estatística das conclusões. Outra ameaça é a análise dos dados pois a interpretação dos dados coletados pode estar sujeita a vieses, pois há dependência de análise de respostas qualitativas.

Além disso, a inferência de dados é outra ameaça à validade, uma vez que, em alguns casos, os participantes demonstraram dificuldades em associar corretamente as relações metamórficas às características dos cenários apresentados. Por exemplo, o participante P31 identificou a característica “*mudança do sujeito na frase*” como relevante para o contexto do teste, mas não conseguiu especificar a relação metamórfica correspondente. Neste caso, foi necessário inferir a relação RM2 - Substituição, com base na descrição fornecida e no contexto do cenário. Essa dependência de inferências pode introduzir viés na análise, comprometendo a consistência e a objetividade das conclusões.

4.7 CONSIDERAÇÕES FINAIS

Neste capítulo foram discutidos os resultados da avaliação do Tesutã por meio de um estudo de viabilidade com participantes de diferentes níveis de experiência. A análise revelou que o Tesutã apresenta potencial significativo para apoiar a execução de testes metamórficos em chatbots, especialmente em cenários onde a compreensão

de linguagem natural e a identificação de falhas contextuais são críticas. No entanto, também foram evidenciadas limitações que devem ser abordadas para maximizar sua efetividade e acessibilidade.

Os resultados indicaram que, embora o Tesutã seja conciso e instrutivo em sua estrutura, a clareza e a aplicabilidade de algumas heurísticas variaram dependendo do perfil dos participantes. Casos de sucesso, como o do participante P10, destacaram a utilidade do Tesutã na identificação de requisitos e relações metamórficas quando as instruções eram corretamente compreendidas. Por outro lado, os participantes P1, P19 e P23 enfrentaram dificuldades em associar características dos cenários aos conceitos apresentados, o que impactou negativamente a completude dos requisitos identificados. Em situações específicas, foi necessário inferir relações metamórficas a partir de descrições fornecidas pelos participantes, como no caso do participante P31, o que introduz um potencial viés na análise e sugere a necessidade de ajustes na apresentação do conteúdo do Tesutã.

Além disso, a inclusão de mais exemplos práticos, que contextualizem a aplicação das heurísticas em situações reais, foi identificada como uma melhoria essencial para facilitar a compreensão e a execução dos testes. Participantes também sugeriram a simplificação da linguagem técnica e a introdução de exercícios interativos para consolidar conceitos avançados, como as relações metamórficas, especialmente para testadores com menor experiência na área.

Apesar das limitações identificadas, o estudo reforçou que o Tesutã é um mecanismo promissor e relevante para o contexto de testes em chatbots. A realização de estudos futuros com amostras mais diversificadas, bem como a validação do Tesutã em diferentes domínios e tecnologias de chatbot, será fundamental para refinar o material e ampliar sua aplicabilidade. Revisões contínuas e adaptações contextuais são essenciais para assegurar que o Tesutã se mantenha uma ferramenta prática, acessível e eficaz, contribuindo para a melhoria da qualidade e confiabilidade de agentes conversacionais em um cenário de constante evolução tecnológica.

5 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

O teste em chatbots baseados em PLN apresenta desafios significativos, pois uma única entrada pode gerar múltiplas saídas (AQUINO; ADANIYA, 2018). Além disso, conforme identificado no MSL realizado (SANTOS et al., 2024), ainda não existem diretrizes claras sobre como realizar teste nessas aplicações de forma eficaz (SANTOS et al., 2024). Portanto, este projeto de mestrado teve como principal objetivo desenvolver um guia para apoiar a atividade de teste em *chatbots* baseados em PLN.

Esse guia foi denominado Tesutã e foi elaborado com base em heurísticas específicas para auxiliar testadores na realização de testes metamórficos em chatbots, oferecendo um conjunto de diretrizes claras e práticas que podem ser aplicadas desde a compreensão dos requisitos do sistema até a documentação dos resultados obtidos. Essas heurísticas foram projetadas para abordar de forma eficiente os desafios únicos encontrados ao testar chatbots baseados em PLN, como a diversidade de respostas possíveis a partir de uma única entrada e a necessidade de garantir consistência e integridade nos resultados.

Para avaliar o Tesutã, um estudo de viabilidade foi realizado, o qual demonstrou que o Tesutã pode ser considerado uma abordagem útil e eficaz para testadores, proporcionando um suporte valioso na condução de testes metamórficos em *chatbots*. Vale a pena mencionar que esse trabalho também indicou possíveis melhorias em diversas áreas como a necessidade de maior acessibilidade linguística, inclusão de mais exemplos práticos e a simplificação de conceitos técnicos para facilitar o uso do guia por um público mais amplo. Esses pontos sugerem que, com os ajustes adequados, o guia Tesutã poderá se tornar ainda mais eficaz e acessível para os diversos usuários.

5.1 CONTRIBUIÇÕES

Dentre as contribuições identificadas, destacam-se:

- **Identificação do Estado da Arte:** Foi conduzido um mapeamento sistemático da literatura (MSL) para identificar as técnicas e ferramentas mais utilizadas no teste de *chatbots*. Esse levantamento forneceu uma visão abrangente do estado da arte, destacando metodologias eficazes e evidenciando lacunas existentes no campo de estudo (SANTOS et al., 2024);
- **Exploração dos Testes Metamórficos em Chatbots:** Foi realizada uma análise aprofundada sobre a aplicação de testes metamórficos no contexto de *chatbots*, facilitando a compreensão do tema para profissionais e pesquisadores. Esses insights auxiliam na adaptação das técnicas de teste metamórfico às particularidades das interações em linguagem natural;
- **Desenvolvimento do Tesutã:** Criou-se o guia Tesutã, disponibilizado por meio de uma *wiki online* acessível. Esse guia oferece diretrizes claras e organizadas para a realização de testes metamórficos em *chatbots*, funcionando como um recurso essencial para testadores, desde iniciantes até usuários mais experientes;
- **Desenvolvimento de exemplos práticos:** Foram desenvolvidos exemplos práticos que ilustram a aplicação das heurísticas do Tesutã, facilitando o aprendizado e a adoção do guia pelos usuários.
- **Treinamento:** Foi desenvolvido um treinamento dedicado sobre testes metamórficos, abordando tanto os fundamentos teóricos quanto a aplicação prática do Tesutã;
- **Validação do Tesutã:** A validação experimental do guia permitiu identificar seus pontos fortes e limitações. Esse processo forneceu *feedback* essencial para refinar o Tesutã, assegurando sua eficácia e adaptabilidade a diferentes cenários de teste e níveis de experiência dos usuários.

5.2 TRABALHOS FUTUROS

A partir do estudo de viabilidade realizado no Capítulo 4, foram identificados alguns aspectos do Tesutã que podem ser investigados como trabalhos futuros. Dentre esses aspectos, destacam-se:

- **Expansão do Tesutã para outros domínios:** Investigar a aplicação do Tesutã em outros tipos de sistemas baseados em inteligência artificial, além de chatbots, como assistentes virtuais e sistemas de recomendação, adaptando as heurísticas para novos contextos.
- **Incorporação de técnicas de *machine learning*:** Explorar a integração de técnicas de aprendizado de máquina no processo de teste metamórfico, visando automatizar a identificação de padrões e possíveis falhas em chatbots de forma mais eficiente.
- **Melhoria da acessibilidade linguística:** Desenvolver versões do Tesutã em diferentes idiomas e com uma linguagem mais acessível, visando ampliar o público-alvo e facilitar o uso por testadores com diferentes níveis de experiência.
- **Avaliação empírica em ambientes reais:** Conduzir estudos empíricos em ambientes de produção reais para validar ainda mais a eficácia do Tesutã, coletando dados sobre seu impacto na detecção de falhas e na melhoria da qualidade dos chatbots.

5.3 PRODUÇÃO CIENTÍFICA

A seguir, apresentam-se as publicações já finalizadas e as publicações que estão em processo de submissão.

5.3.1 PUBLICAÇÕES ELABORADAS

Como parte dos resultados obtidos na realização deste projeto de mestrado, destacam-se as seguintes publicações:

- SANTOS, Gabriel; SILVA, Williamson; VALLE, Pedro Henrique Dias. **O que sabemos sobre testes em chatbots? Uma revisão sistemática da literatura.** In: WORKSHOP SOBRE ASPECTOS SOCIAIS, HUMANOS E ECONÔMICOS DE SOFTWARE (WASHES), Brasília/DF. Sociedade Brasileira de Computação, 2024. p. 106-117.
- SANTOS, Gabriel; SILVA, Williamson; VALLE, Pedro Henrique Dias. **Uma técnica para apoiar profissionais de testes na condução de testes metamórficos em chatbots.** In: Workshop sobre Bots na Engenharia de Software (WBOTS). Congresso Brasileiro de Software (CBSOFT 2024).

REFERÊNCIAS

- ADAMOPOULOU, E.; MOUSSIADES, L. Chatbots: History, technology, and applications. **Machine Learning with Applications**, Elsevier, v. 2, p. 100006, 2020.
- ADAMOPOULOU, E.; MOUSSIADES, L. An overview of chatbot technology. In: SPRINGER. **IFIP international conference on artificial intelligence applications and innovations**. [S.l.], 2020. p. 373–383.
- AMMANN, P.; OFFUTT, J. **Introduction to software testing**. [S.l.]: Cambridge University Press, 2017.
- ANDRADE, S. A. d. **Uma abordagem de teste de software para aplicações de realidade virtual utilizando testes metamórficos**. Tese (Doutorado) — Universidade de São Paulo, 2022.
- ANICHE, M. **Effective Software Testing: A developer's guide**. [S.l.]: Simon and Schuster, 2022.
- AQUINO, V. H. de O.; ADANIYA, M. H. A. da C. Desenvolvimento e aplicações de chatbot. **Revista Terra & Cultura: Cadernos de Ensino e Pesquisa**, v. 34, n. esp., p. 56–68, 2018.
- ASSUNÇÃO, M. A. d. Teste de software: técnicas de teste estrutural e análise de código. 004, 2015.
- BARBOSA, E.; CHAIM, M.; VINCENZI, A.; DELAMARO, M.; JINO, M.; MALDONADO, J. Introdução ao teste de software. cap. 4-teste estrutural. **Campus/Elsevier**, 2007.
- BIRD, J. J.; EKÁRT, A.; FARIA, D. R. Chatbot interaction with artificial intelligence: human data augmentation with t5 and language transformer ensemble for text classification. **Journal of Ambient Intelligence and Humanized Computing**, Springer, v. 14, n. 4, p. 3129–3144, 2023.
- BISHOP, C. M.; NASRABADI, N. M. **Pattern recognition and machine learning**. [S.l.]: Springer, 2006.
- BOZIC, J.; TAZL, O. A.; WOTAWA, F. Chatbot testing using ai planning. In: IEEE. **2019 IEEE International Conference On Artificial Intelligence Testing (AITest)**. [S.l.], 2019. p. 37–44.
- BOZIC, J.; WOTAWA, F. Testing chatbots using metamorphic relations. In: SPRINGER. **Testing Software and Systems: 31st IFIP WG 6.1 International Conference, ICTSS 2019, Paris, France, October 15–17, 2019, Proceedings 31**. [S.l.], 2019. p. 41–55.
- BOŽIĆ, J. Ontology-based metamorphic testing for chatbots. **Software Quality Journal**, Springer, v. 30, p. 227–251, 3 2022. ISSN 15731367.

BRAVO-SANTOS, S.; GUERRA, E.; LARA, J. de. Testing chatbots with charm. In: SPRINGER. **Quality of Information and Communications Technology: 13th International Conference, QUATIC 2020, Faro, Portugal, September 9–11, 2020, Proceedings 13**. [S.l.], 2020. p. 426–438.

CABOT, J.; BURGUENO, L.; CLARISÓ, R.; DANIEL, G.; PERIANEZ-PASCUAL, J.; RODRIGUEZ-ECHEVERRIA, R. Testing challenges for nlp-intensive bots. In: IEEE. **2021 IEEE/ACM Third International Workshop on Bots in Software Engineering (BotSE)**. [S.l.], 2021. p. 31–34.

CAMATTI, J. A. et al. Proposta de arquitetura oee digital com integração de parâmetros de análise de máquinas na indústria de manufatura. Universidade Tecnológica Federal do Paraná, 2023.

CHEN, T. Y.; CHEUNG, S. C.; YIU, S. M. Metamorphic testing: a new approach for generating next test cases. **arXiv preprint arXiv:2002.12543**, 2020.

CHEN, T. Y.; KUO, F.-C.; LIU, H.; POON, P.-L.; TOWEY, D.; TSE, T.; ZHOU, Z. Q. Metamorphic testing: A review of challenges and opportunities. **ACM Computing Surveys (CSUR)**, ACM New York, NY, USA, v. 51, n. 1, p. 1–27, 2018.

CONSULTORIA, G. **Requisitos Funcionais e Requisitos Não Funcionais do Surpreendente ChatGPT**. abril 2023. <https://giganteconsultoria.com.br/2023/04/09/requisitos-funcionais-e-requisitos-nao-funcionais-do-surpreendente-chatgpt/>.

CRESPO, A. N. Modelos de confiabilidade de software baseados em cobertura de critérios estruturais de teste. [sn], 1997.

DEEPAISARN, S.; IMKOME, E. uma; WONGPATIKASEREE, K.; YUENYONG, S.; LAKANAVISID, P.; SOONTHORNCHAIVA, R.; YOMABOOT, P.; ANGKOONSAWAENGSAK, A.; MUNPANSAN, N. Validation of a thai artificial chatmate designed for cheering up the elderly during the covid-19 pandemic. **F1000Research**, F1000 Research Ltd, v. 11, p. 1411, 11 2022. ISSN 1759796X.

DELAMARO, M.; MALDONADO, J.; JINO, M. **Introdução ao teste de software (2 edição ed.)**. [S.l.]: Elsevier, Rio de Janeiro, 2016.

ESPINDOLA, M. G.; CORDENONZI, W. Qualidade em pacotes de software. **Disciplinarum Scientia— Naturais e Tecnológicas**, v. 3, n. 1, p. 29–43, 2002.

FILHO, J. R. A. d. S. S. Dubi: um framework para avaliação automática de chatbots. 2024.

GUERREIRO, A.; BARROS, D. M. V. Novos desafios da educação a distância: programação e uso de chatbots. 2019.

GUGLIELMI, E.; ROSA, G.; SCALABRINO, S.; BAVOTA, G.; OLIVETO, R. Sorry, i don't understand: Improving voice user interface testing. In: . [S.l.]: Association for Computing Machinery, 2022. ISBN 9781450396240.

IEEE. Ieee standard glossary of software engineering terminology. **IEEE Std 610.12-1990**, p. 1–84, 1990.

IRIGARAY, H. A. R.; STOCKER, F. **ChatGPT: um museu de grandes novidades**. [S.l.]: SciELO Brasil, 2023. e88776 p.

ISO, I. Iec 25000: 2014 systems and software engineering—systems and software quality requirements and evaluation (square)—guide to square. **International Organization for Standardization**, 2014.

JEYARAMAN, M.; RAMASUBRAMANIAN, S.; BALAJI, S.; JEYARAMAN, N.; NALLAKUMARASAMY, A.; SHARMA, S. Chatgpt in action: Harnessing artificial intelligence potential and addressing ethical challenges in medicine, education, and scientific research. **World Journal of Methodology**, Baishideng Publishing Group Inc, v. 13, n. 4, p. 170, 2023.

KITCHENHAM, B.; PRETORIUS, R.; BUDGEN, D.; BRERETON, O. P.; TURNER, M.; NIAZI, M.; LINKMAN, S. Systematic literature reviews in software engineering—a tertiary study. **Information and software technology**, Elsevier, v. 52, n. 8, p. 792–805, 2010.

KO, A. J.; ABRAHAM, R.; BECKWITH, L.; BLACKWELL, A.; BURNETT, M.; ERWIG, M.; SCAFFIDI, C.; LAWRENCE, J.; LIEBERMAN, H.; MYERS, B. et al. The state of the art in end-user software engineering. **ACM Computing Surveys (CSUR)**, ACM New York, NY, USA, v. 43, n. 3, p. 1–44, 2011.

LE MOS, A. **O que é ChatGPT: como usar a IA em Português no seu dia a dia**. 2023. Disponível em: <<https://exame.com/inteligencia-artificial/o-que-e-chatgpt-como-usar-a-ia-em-portugues-no-seu-dia-a-dia/>>.

McKinsey & Company. **ChatGPT e a revolução da inteligência artificial**. sem data. Disponível em: <<https://www.mckinsey.com/br/our-insights/all-insights/chatgpt-e-a-revolucao-da-inteligencia-artificial>>.

MORAES, S. M.; SOUZA, L. S. de. Uma abordagem semiautomática para expansão e enriquecimento linguístico de bases aiml para chatbots. In: **Congresso Internacional de Informática Educativa**. [S.l.: s.n.], 2015. v. 20, p. 600–605.

MORAIS, B.; SANTIAGO, C. Getlab: Um aplicativo para reserva de espaços acadêmicos por meio de um agente conversacional. **Revista de Ciência da Computação**, v. 5, n. 1, p. 28–37, 2023.

NUNES, F. O. Chatbots e mimetismo: uma conversa entre humanos, robôs e artistas. In: **Proceedings of 6th International Conference on Digital Arts—ARTECH**. [S.l.: s.n.], 2012. p. 89–96.

OKANOVIĆ, D.; BECK, S.; MERZ, L.; ZORN, C.; MERINO, L.; HOORN, A. V.; BECK, F. Can a chatbot support software engineers with load testing? approach and experiences. In: . [S.l.]: Association for Computing Machinery, Inc, 2020. p. 120–129. ISBN 9781450369916.

OLIVEIRA, E. G. **Chatbots: a importância do processamento de língua natural para a experiência do usuário**. Tese (Doutorado), 2023.

PADMANABHAN, M. Sustainable test path generation for chatbots using customized response. **International Journal of Engineering and Advanced Technology**, Blue Eyes Intelligence Engineering and Sciences Publication, v. 8, p. 149–155, 8 2019. ISSN 22498958.

PASCHOAL, L. N.; NOGUEIRA, L. L.; CHICON, P. M. M. Agentes conversacionais pedagógicos: Uma discussão inicial sobre conceitos, estratégias de desenvolvimento e oportunidades de pesquisa. **Digitalização da Educação: Desafios e Estratégias para a Educação da Geração Conectada**, p. 23, 2020.

RAY, P. P. Chatgpt: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. **Internet of Things and Cyber-Physical Systems**, Elsevier, 2023.

RIOS, E. **Documentação de Teste de Software: Dissecando o padrão IEEE 829**. [S.l.]: São Paulo: Imagem, 2010.

RODRIGUES, F. H.; LOPES, A. G.; SANTOS, N. O. dos; GARCIA, L. F.; CARBONERA, J. L.; ABEL, M. On the use of chatgpt for classifying domain terms according to upper ontologies. In: SPRINGER. **International Conference on Conceptual Modeling**. [S.l.], 2023. p. 249–258.

ROS-ARLANZÓN, P.; PÉREZ-SEMPERE, Á. Chatgpt: una novedosa herramienta de escritura para artículos científicos, pero no un autor (por el momento). **Revista de Neurología**, Viguera Editores, SL, v. 76, n. 8, p. 277, 2023.

RUANE, E.; FAURE, T.; SMITH, R.; BEAN, D.; CARSON-BERNDSEN, J.; VENTRESQUE, A. Botest: a framework to test the quality of conversational agents using divergent input examples. In: **Proceedings of the 23rd International Conference on Intelligent User Interfaces Companion**. [S.l.: s.n.], 2018. p. 1–2.

RUSSELL, S. J. **Artificial intelligence a modern approach**. [S.l.]: Pearson Education, Inc., 2010.

SANTOS, G.; SILVA, W.; VALLE, P. H. D. O que sabemos sobre testes em chatbots? uma revisão sistemática da literatura. In: SBC. **Workshop sobre Aspectos Sociais, Humanos e Econômicos de Software (WASHES)**. [S.l.], 2024. p. 106–117.

SANTOS, M. B. D.; FURTADO, A. P. C.; NOGUEIRA, S. C.; MOREIRA, D. D. Oggybug: A test automation tool in chatbots. In: . [S.l.]: Association for Computing Machinery, 2020. p. 79–87. ISBN 9781450387552.

SEGURA, S.; FRASER, G.; SANCHEZ, A. B.; RUIZ-CORTÉS, A. A survey on metamorphic testing. **IEEE Transactions on software engineering**, IEEE, v. 42, n. 9, p. 805–824, 2016.

SELVI, V.; SARANYA, S.; CHIDIDA, K.; ABARNA, R. Chatbot and bullyfree chat. In: IEEE. **2019 IEEE International Conference on System, Computation, Automation and Networking (ICSCAN)**. [S.l.], 2019. p. 1–5.

SHAWAR, B. A.; ATWELL, E. Chatbots: are they really useful? **Journal for Language Technology and Computational Linguistics**, v. 22, n. 1, p. 29–49, 2007.

SILVA, J. O. S. d. **Qualidade em Serviços Logísticos: aplicação da Soft Systems Methodology (SSM) na avaliação da qualidade percebida quanto à distribuição dos medicamentos antirretrovirais do programa HIV/AIDS**. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2016.

SILVA, W. d. L. **Uso de testes metamórficos para verificação de aplicação chatbot**. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Norte, 2022.

SOARES, J. R.; SILVA, P. N. Panorama da pesquisa sobre chatbots no brasil. **BIBLOS-Revista do Instituto de Ciências Humanas e da Informação**, v. 38, n. 1, p. 199–218, 2024.

SOLINGEN, R. V.; BASILI, V.; CALDIERA, G.; ROMBACH, H. D. Goal question metric (gqm) approach. **Encyclopedia of software engineering**, Wiley Online Library, 2002.

SOUZA, P. H. C. Proposta de implementação de chatbot para o observatório do instituto do mar. Universidade Federal de São Paulo, 2022.

TEDE, P. C. d. A. R.; BARROS, F. de A. Agentes inteligentes conversacionais: conceitos básicos e desenvolvimento. **Sociedade Brasileira de Computação**, 2016.

TONTINI, G.; ZANCHETT, R. Atributos de satisfação e lealdade em serviços logísticos. **Gestão & Produção**, SciELO Brasil, v. 17, p. 801–816, 2010.

VALLE, P. H. D. **Jogos educacionais: uma contribuição para o ensino de teste de software**. Tese (Doutorado) — Universidade de São Paulo, 2016.

VASCONCELOS, M.; CANDELLO, H.; PINHANEZ, C.; SANTOS, T. dos. Bottester: testing conversational systems with simulated users. In: **Proceedings of the XVI Brazilian Symposium on Human Factors in Computing Systems**. [S.l.: s.n.], 2017. p. 1–4.

VELÁSQUEZ, F. R. O chatgpt na pesquisa em humanidades digitais: Oportunidades, críticas e desafios. **TEKOA**, v. 2, n. 2, 2023.

VIJAYARAGHAVAN, V.; COOPER, J. B.; LEEVINSON, R. L. R. Algorithm inspection for chatbot performance evaluation. In: . [S.l.]: Elsevier B.V., 2020. v. 171, p. 2267–2274. ISSN 18770509.

WALLACE, R. Alice-artificial linguistic internet computer entity-the alice ai. foundation. **Disponível em <http://www.alicebot.org>**. Acesso em, v. 18, 1995.

ZHANG, B.; DING, D.; JING, L. How would stance detection techniques evolve after the launch of chatgpt? **arXiv preprint arXiv:2212.14548**, 2022.

ZHOU, Y.; LI, Y.; WANG, L.; LI, X. ncov: A tool for measuring length-n subpath coverage. In: **Proceedings of the 7th Asia-Pacific Symposium on Internetware**. [S.l.: s.n.], 2015. p. 176–185.

ANEXO A – RESUMO DOS ESTUDOS PRIMÁRIOS SELECIONADOS

A.1 RESUMO DOS ESTUDOS PRIMÁRIOS SELECIONADOS

Vale a pena ressaltar que apenas cinco estudos primários foram selecionados utilizando a *string* de busca apresentada anteriormente. A seguir, apresenta-se uma breve descrição dos estudos selecionados.

A.1.1 OGGYBUG: A TEST AUTOMATION TOOL IN CHATBOTS

Este estudo tem como objetivo propor um novo *framework* para testes em *chatbots* visando o contexto da informação e permitir a integração de testes entre diferentes bases de dados. Para tal, foi desenvolvido um *chatbot* chamado OggyBug que possibilitou a implementação, gestão e relato dos resultados da execução dos testes automatizados através de *API's* e/ou por meio de uma interface *web*.

Esse estudo mostrou que testar os *chatbots* no contexto da informação entre a aplicação e o usuário é algo significativo. No entanto, a configuração da integração dos testes se mostrou complexa devido a necessidade de parametrização do *web service* nas ações do chatbot (SANTOS et al., 2020).

A.1.2 SORRY, I DON'T UNDERSTAND: IMPROVING VOICE USER INTERFACE TESTING

Este estudo propôs uma nova abordagem com o objetivo de adaptar as abordagens existentes de teste em chatbots para testes em agentes de voz (VUI¹). Para isso, durante os testes foram integradas ferramentas baseadas em *Deep Learning*² para gerar frases para os comandos especificados pelos desenvolvedores. No entanto, essas ferramentas geravam novas frases que nem sempre eram cobertas pelos casos de teste. Nesse sentido, realizaram um estudo empírico para entender como as ferramentas VUI compara-se com as abordagens existentes. Como resultado, a abordagem identificou mais *bugs* nas frases geradas do que as demais (GUGLIELMI et al., 2022).

¹VUI: *Voice User Interface*

²*Deep Learning* é uma estrutura composta por várias camadas de neurônios artificiais interconectados, projetada para aprender e extrair representações complexas de dados.

A.1.3 ONTOLOGY-BASED METAMORPHIC TESTING FOR CHATBOTS

Ontologia é um conceito de IA que fornece representações formais para conhecimentos de domínios específicos (BOŽIĆ, 2022). Estes modelos possuem informações estruturadas que são utilizadas para geração de testes. Por outro lado, os testes em *chatbots* representam um desafio ocasionado pela falta de um oráculo. Portanto, este estudo propôs uma abordagem de testes metamórficos baseados em ontologia, onde ambos os conceitos foram utilizados para geração de entrada e processamento de saída na forma de uma abordagem de teste metamórfico. Como a saída de sistemas de IA é imprevisível, essa abordagem substitui os oráculos de teste convencionais por verificações metamórficas. O método consiste em utilizar as ontologias em conjunto com MRs (Regras Metamórficas) para gerar e executar testes metamórficos com o objetivo de avaliar a funcionalidade desses programas.

Para validar essa abordagem, um estudo utilizando um *chatbot* da indústria do turismo foi utilizado. Por fim, concluiu-se que a partir dos testes que essa abordagem pode ser utilizada como um modelo de entrada para geração de teste. Por outro lado, as abordagens baseadas em testes metamórficos podem ser aplicadas em circunstâncias em que o comportamento esperado é difícil ou impossível de determinar (BOŽIĆ, 2022).

A.1.4 SUSTAINABLE TEST PATH GENERATION FOR CHATBOTS USING CUSTOMIZED RESPONSE

Este estudo propôs uma técnica para gerar um caminho de teste eficaz para *chatbots*. Diante de várias bases de dados em ferramentas digitais, os *inputs*³ dos usuários esperam respostas satisfatórias, principalmente em serviços críticos, como na área da saúde. Sendo assim, foram conduzidos testes experimentais associados a quatro metodologias de respostas de *chatbots*. A pesquisa mostrou que aproximadamente 66% expressaram estar desapontados com as respostas dos *chatbots* utilizados no dia-a-dia e 99% dos estudantes aprovaram o *chatbot* que utilizou a técnica de teste proposta (PADMANABHAN, 2019).

³*Input*: No contexto de *chatbots* refere-se a comandos de entrada para que o agente possa processar e executar uma ação ou responder a essa entrada.

A.1.5 ALGORITHM INSPECTION FOR CHATBOT PERFORMANCE EVALUATION

Este estudo abordou de forma abrangente os principais métodos de teste de *chatbot*, com ênfase detalhada nas técnicas de teste de algoritmo. Conforme dito anteriormente, os *chatbots* vêm sendo utilizados em diversos domínios e setores. A partir da análise destes, identificaram-se várias técnicas de testes generalistas. Porém, dentre estas, duas se destacam: teste de saída e teste de algoritmo.

O teste de saída, como mencionado, envolve a submissão de inputs ao chatbot e a monitorização de suas respostas. A qualidade das respostas é então avaliada com base em critérios como relevância, completude, precisão e contexto. Uma abordagem colaborativa de código aberto chamada Chatest foi citada como uma plataforma abrangente para avaliar o desempenho dos chatbots em várias áreas, proporcionando 120 perguntas em paradigmas como Resposta, Gerenciamento de Erros, Inteligência, Navegação, Integração, Personalidade e Compreensão.

O teste de algoritmo é particularmente crucial para identificar falhas específicas e determinar onde ajustes são necessários. O estudo enfatizou a importância da técnica de validação cruzada (*cross-validation*), que envolve dividir um conjunto de dados em conjuntos de treinamento e teste para avaliar a eficácia do algoritmo.

A segunda técnica se mostrou a mais promissora pois é difícil inspecionar e analisar o desempenho de técnicas baseadas em PNL que são usualmente utilizadas. Além disso, existem situações de ambiguidade e que representam outro desafio para a área de testes. Como conclusão, para superar esses desafios, abordagens de verificação e validação juntamente com análise estatística e outras ferramentas de detecção de ambiguidade serão fundamentais. Isso irá garantir que, quando uma entrada ambígua ou irrelevante for detectada, o *bot* poderá pedir ao usuário que reformule a pergunta adequadamente (VIJAYARAGHAVAN et al., 2020).

A.1.6 BOTEST: A FRAMEWORK TO TEST THE QUALITY OF CONVERSATIONAL AGENTS USING DIVERGENT INPUT EXAMPLES

O BoTest é um framework proposto para avaliar a qualidade de agentes de conversação, concentrando-se na capacidade de lidar com exemplos de entrada divergentes. O objetivo é testar a robustez dos agentes diante de complexidade introduzida por variações na linguagem. O framework utiliza exemplos divergentes baseados em enunciados conhecidos, mantendo a intenção original, e avalia o desempenho do agente diante dessas variações. O ChitChatBot, um agente de conversação simples, foi desenvolvido para demonstrar o framework. O estudo manualmente introduziu divergências, como erros de preposição e frases coloquiais, e avaliou o desempenho do ChitChatBot. Os resultados mostram que o framework pode destacar áreas onde a qualidade do agente é reduzida, proporcionando feedback valioso para os desenvolvedores (RUANE et al., 2018).

A.1.7 BOTTESTER: TESTING CONVERSATIONAL SYSTEMS WITH SIMULATED USERS

Esse estudo propõe a avaliação de sistemas de conversação, como *chatbots*, destacando a necessidade de ferramentas de teste para garantir o bom funcionamento desses sistemas. Os autores propõem o uso de uma ferramenta chamada Bottester, que simula interações de usuários para coletar métricas relacionadas à satisfação do usuário e ao desempenho geral do sistema. O estudo inclui um caso de uso com um *chatbot* de aconselhamento financeiro chamado CognIA. O Bottester avalia aspectos como o tamanho e a concisão das respostas, a frequência de respostas repetitivas, a limitação da base de conhecimento e o tempo de resposta, visando melhorar a experiência do usuário e identificar problemas de desempenho. O documento enfatiza a importância do teste automatizado para detectar problemas antes dos experimentos com usuários reais e sugere que a ferramenta pode ser valiosa na avaliação de *chatbots* em diferentes contextos (VASCONCELOS et al., 2017).

A.1.8 CHATBOT AND BULLYFREE CHAT

O estudo aborda a criação de um *chatbot* usando NLP e algoritmos de aprendizado de máquina, com foco na detecção de *cyberbullying* em interações de chat bidirecional. O documento destaca o papel da NLP e do aprendizado de máquina na construção do *chatbot*, mencionando algoritmos como redes neurais, redes neurais recorrentes e o uso de modelos de bolsa de palavras (*Bag of Words* - BOW) para detectar *cyberbullying*. O *chatbot* é descrito como amigável ao usuário, capaz de responder a consultas e evitar palavras ofensivas. O trabalho conclui com a proposta de um algoritmo que demonstra um desempenho superior em relação às técnicas existentes, e sugere futuras aplicações para detectar *cyberbullying* em *chatbots* habilitados para *websites* (SELVI et al., 2019).

A.1.9 CHATBOT INTERACTION WITH ARTIFICIAL INTELLIGENCE: HUMAN DATA AUGMENTATION WITH T5 AND LANGUAGE TRANSFORMER ENSEMBLE FOR TEXT CLASSIFICATION

O estudo aborda a implementação de *chatbots* aprimorados por inteligência artificial (IA). O estudo utiliza uma abordagem de ensemble⁴, combinando vários modelos de *transformers*, incluindo RoBERTa e DistilBERT, para classificação de texto. A técnica de *data augmentation* usando o modelo T5 é explorada para melhorar a capacidade do sistema em interpretar e responder a comandos humanos. O desempenho é avaliado em várias tarefas, como reconhecimento de sentimentos e classificação de cenas, demonstrando melhorias significativas na precisão. O estudo também aborda questões éticas relacionadas à coleta de dados e divulgação da pesquisa (BIRD et al., 2023).

A.1.10 CHATBOT TESTING USING AI PLANNING

O estudo discute uma abordagem de teste funcional para *chatbots* baseada em planejamento de IA. A metodologia utiliza planejamento para criar modelos que

⁴Ensemble: prática de combinar os resultados de múltiplos modelos de aprendizado de máquina para melhorar o desempenho geral do sistema. Em vez de depender de um único modelo, a técnica de ensemble utiliza a diversidade de vários modelos para compensar as fraquezas individuais e explorar diferentes perspectivas na resolução de um problema específico.

geram casos de teste para *chatbots*, especificamente testando a funcionalidade de um *chatbot* de turismo ao reservar um hotel. A geração de sequências de ações por meio do planejamento representa diferentes cenários de teste. Um framework de teste automatizado executa esses casos de teste, avaliando a reação do *chatbot*. O estudo conclui que a abordagem é bem-sucedida, embora algumas limitações na capacidade do *chatbot* de reconhecer a intenção do usuário tenham sido observadas, sugerindo áreas para melhorias futuras. O planejamento baseado em IA é proposto como uma abordagem geral para testes funcionais e não funcionais de sistemas mais amplos. O artigo também menciona trabalhos relacionados, destacando a escassez de literatura sobre testes de *chatbots* e fornece referências a ferramentas relevantes (BOZIC et al., 2019).

A.1.11 TESTING CHALLENGES FOR NLP-INTENSIVE BOTS

O artigo aborda os desafios de teste para *bots* intensivos em Processamento de Linguagem Natural (PLN). Esses *bots*, como *chatbots* e *voicebots*, são cada vez mais populares, mas a qualidade deles é uma preocupação para as empresas devido ao impacto significativo que mau funcionamento pode ter na receita ou imagem da empresa. O documento destaca que o teste de *bots* PLN está em uma fase inicial, carecendo de definições claras de propriedades e técnicas de teste específicas. O artigo propõe uma caracterização dessas propriedades e técnicas, adaptando conceitos de teste existentes para avançar em direção a um framework de referência unificado para futuras iniciativas de pesquisa em teste de *bots*. O texto aborda elementos fundamentais de *bots* PLN, desafios atuais no estado da arte, níveis de teste, objetivos de teste, técnicas de teste e medidas relacionadas ao teste. O documento conclui destacando a necessidade contínua de esforços na área de teste de *bots* PLN e sua relação com desafios conhecidos no teste de software (CABOT et al., 2021).

A.1.12 TESTING CHATBOTS USING METAMORPHIC RELATIONS

Esse estudo aborda a aplicação de testes metamórficos em sistemas de IA, com foco específico em *chatbots*. A metodologia proposta utiliza relações

metamórficas para testar *chatbots*, explorando as peculiaridades da linguagem natural. O estudo inclui uma implementação prática em um *chatbot* do setor de turismo, destacando descobertas e *insights* sobre o comportamento do sistema em diferentes cenários. O artigo também discute trabalhos relacionados, evidenciando a relevância e as possíveis extensões dessa abordagem de teste (BOZIC; WOTAWA, 2019).

A.1.13 TESTING CHATBOTS WITH CHARM

O trabalho propõe técnicas automatizadas para a síntese de testes de *chatbots*, apresentando uma metodologia que suporta três tipos de testes: coesão, robustez e precisão. A ferramenta chamada Charm, implementada em Python, utiliza o Botium como *backend* para automação dos testes. Um experimento foi conduzido com três *chatbots* do Dialogflow, demonstrando que o Charm pode revelar falhas nos *chatbots* e contribuir para a melhoria de sua qualidade. O estudo discute a necessidade de experimentos adicionais com *chatbots* mais complexos e destaca a importância de abordagens automatizadas para garantir a qualidade dos *chatbots* (BRAVO-SANTOS et al., 2020).

ANEXO B – EXEMPLO DE EXECUÇÃO DAS HEURÍSTICAS COM O CHATGPT

B.1 EXEMPLO DE EXECUÇÃO DAS HEURÍSTICAS COM O CHATGPT

Neste Anexo, será apresentado um exemplo de execução das heurísticas utilizando o ChatGPT. O teste será realizado na versão 3.5 que está disponível de forma gratuita aos usuários. A Tabela 11 apresenta novamente as heurísticas para facilitar a consulta.

O ChatGPT possui documentação apenas para sua API. Desta forma, para a realização deste teste, foi necessário realizar uma pesquisa a parte nas bases de pesquisa e web sites. Sendo assim, de acordo com a heurística H_1 , foi preciso identificar quais são os requisitos funcionais e não funcionais do ChatGPT e estão listados a seguir:

- Funcionais (ROS-ARLANZÓN; PÉREZ-SEMPERE, 2023; IRIGARAY; STOCKER, 2023; CONSULTORIA, 2023).

– RF_1 **Geração de texto:** o ChatGPT é capaz de gerar textos de forma autônoma, respondendo a *prompts* (entrada do usuário) de maneira relevante. Exemplo: redigir textos sobre o contexto do *prompt* e ajustar de acordo com as especificações fornecidas.

– RF_2 **Adaptabilidade:** o ChatGPT é capaz de ajustar suas respostas de acordo com o contexto fornecido, permitindo uma interação dinâmica. Exemplo: o *prompt* inicial pede para gerar uma frase motivacional. Na próxima interação, o usuário pede que a frase seja no contexto de programação.

- Não funcionais (ROS-ARLANZÓN; PÉREZ-SEMPERE, 2023; IRIGARAY; STOCKER, 2023; CONSULTORIA, 2023).

– RNF_1 **Integridade e responsabilidade:** Deve garantir que a tecnologia seja utilizada de forma ética, transparente, segura e responsável, promovendo assim o benefício e o bem-estar dos usuários de acordo com a LGPD.

– RNF_2 **Escalabilidade:** Deve ser capaz de lidar com o aumento na carga de trabalho, seja aumentando do número de usuários ou na frequência de interações, sem degradar o desempenho;

Tabela 11: Heurísticas *Tesutã*

Identificador	Descrição
H_1	Verificar a documentação de requisitos do <i>chatbot</i> e analisar quais são os requisitos funcionais e não funcionais a serem validados. Se necessário, levante mais informações com o cliente/público alvo;
H_2	Identificar em qual domínio o <i>Chatbot</i> enquadra-se;
H_3	Identificar quais são os comportamentos esperados do <i>Chatbot</i> ;
H_4	Identificar, a partir dos requisitos levantados, quais são elegíveis a sofrerem metamorfismo, isto é, quais são passíveis de receber mudança no <i>input</i> ;
H_5	Garantir que as funcionalidades principais do <i>Chatbot</i> sejam selecionadas;
H_6	A partir das funcionalidades selecionadas, crie casos de teste que refletem os metamorfismos selecionados;
H_7	Assegurar que os metamorfos identificados foram incluídos no caso de teste;
H_8	Registrar as respostas geradas para cada conjunto de metamorfos;
H_9	Registrar desvio de comportamento, caso haja;
H_{10}	Comparar os resultados obtidos com as expectativas definidas;
H_{11}	A partir dos resultados obtidos, ajuste os metamorfos caso encontre variações que o <i>Chatbot</i> não lide corretamente, ou seja, faça mais testes no problema encontrado para garantir que não haja mais situações semelhantes;
H_{12}	Considere um conjunto de metamorfos diferentes dos iniciais nesta etapa;
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 ;
H_{14}	Considere utilizar a automação de testes para aumentar a eficiência do processo, analise se o <i>Chatbot</i> em questão pode ser automatizado com as ferramentas disponíveis no mercado;
H_{15}	Documentar os metamorfos utilizados em cada caso de teste;
H_{16}	Documentar os resultados obtidos e quais problemas foram identificados;
H_{17}	Faça uma análise dos resultados obtidos e problemas encontrados para aprimorar o <i>Chatbot</i> ;
H_{18}	Implemente um sistema/processo de monitoramento contínuo, ou seja, para cada <i>feature/bug fix</i> seja realizado uma nova verificação do caso de teste;

–*RNF*₃ **Disponibilidade:** deve estar disponível e operacional o tempo todo garantindo que os usuários possam interagir com ele a qualquer momento;

–*RNF*₄ **Segurança:** proteção contra acesso não autorizado e garantia de

proteção de dados, de acordo com a LGPD¹;

Em relação à heurística H_2 , é necessário identificar o domínio de aplicação. No entanto, o ChatGPT não está vinculado a um domínio específico, pois é um modelo de linguagem que utiliza PLN para compreender e responder a *prompts* de forma conversacional, abrangendo uma ampla gama de tópicos. Sua concepção foi realizada com o intuito de fornecer uma interface intuitiva para que os usuários possam interagir e obter informações sobre diversos assuntos, os quais não estão restritos a um único domínio ou indústria (RODRIGUES et al., 2023; RAY, 2023; JEYARAMAN et al., 2023).

Na heurística H_3 , é crucial identificar os comportamentos esperados pelo ChatGPT. Portanto, abaixo estão listados alguns desses comportamentos:

- Criatividade e resolução de problemas:** Demonstra habilidade na geração de conteúdo original, como criar histórias a partir das informações fornecidas pelo *prompt*. Além disso, oferece respostas para perguntas complexas e propõe soluções para problemas matemáticos² (McKinsey & Company, sem data; LEMOS, 2023).
- Especialização em temas específicos:** Embora seja um chatbot generalista, o ChatGPT pode fornecer respostas com nível de especialização em determinados temas, dependendo do contexto e do *prompt* fornecido (McKinsey & Company, sem data).
- Poliglota:** Possui a capacidade de interagir em mais de 100 idiomas diferentes (McKinsey & Company, sem data).

A partir dos requisitos identificados na heurística H_1 , torna-se crucial identificar quais destes são elegíveis para sofrer metamorfose, garantindo simultaneamente a inclusão das principais funcionalidades do Chatbot. Essas duas etapas refletem as heurísticas H_4 e H_5 , respectivamente. Portanto, os requisitos RF_1 , RF_2 e RNF_1 foram selecionados. Os requisitos RNF_2 , RNF_3 e RNF_4 não foram selecionados pois o teste

¹LGPD: Lei Geral de Proteção de Dados. É uma legislação que regula o tratamento de dados pessoais por parte de organizações públicas e privadas.

²É importante notar que nem todos os problemas podem ser solucionados devido à complexidade de alguns, como os problemas NP-Completo.

será realizado de forma manual pelo site do próprio Chatbot disponibilizado pela OpenAI.

Assim, com base nos dados coletados e nas definições das heurísticas anteriores, é viável avançar para a heurística H_6 e elaborar os casos de teste. Para tal, é essencial entender que uma relação metamórfica define uma propriedade que deve ser verdadeira para todas as entradas e suas saídas correspondentes. Sendo assim, será utilizado um caso de teste pensando na geração de texto para atender os requisitos RF_1 , RF_2 e RNF_1 . A baixo está descrito o caso de teste base que servirá como alicerce para os metamorfismos. Da mesma forma, a Figura 8 demonstra a resposta do ChatGPT.

CT_{BASE} : Como usuário, quero que crie um texto sobre dois chatbots que conversam entre si e que no final, ficam amigos.

A fim de validar a geração de texto serão utilizadas duas relações metamórficas:

- RM_1 : Substituir palavras-chave por sinônimos;
- RM_2 : Omitir palavras, ou seja, remover palavras da sentença de forma que perca parcialmente o sentido original do *prompt*.

A seguir estão listados os casos de teste que utilizam as relações metamórficas RM_1 e RM_2 :

- CT_1 : Como usuário, quero que crie um **trecho** sobre dois **assistentes virtuais** que **dialogam** entre si e que no final, ficam **camaradas**.
- CT_2 : Como usuário, quero que crie um texto sobre dois chatbots que conversam entre si e que no final, ficam amigos.
- CT_3 : No chat anterior, pedi para que criasse um texto. Poderia consultar o histórico e depois alterar a pergunta para que reflita sobre duas pessoas na praia, no mesmo contexto?

As Figuras 9 e 10 demonstram as respostas do ChatGPT para os casos de teste CT_1 e CT_2 , respectivamente. Além disso, a Figura 11 apresenta a resposta para o requisito RNF_1 .

Neste cenário, as relações metamórficas RM_1 e RM_2 foram aplicadas dentro do escopo definido nos casos de teste, validando assim a heurística H_7 . Dando sequência, deve-se registrar as respostas geradas e desvio de comportamento. Essas atividades referem-se às heurísticas H_8 e H_9 .

A heurística H_{10} refere-se a atividade de comparação das respostas geradas com o que era esperado encontrar a partir dos metamorfismos escolhidos. Sendo assim, no CT_{BASE} o ChatGPT forneceu um diálogo bem estruturado ilustrando uma amizade entre os *chatbots*.

No CT_1 , foi aplicado a relação metamórfica RM_1 , substituindo palavras-chave por sinônimos e, conforme esperado, o ChatGPT foi capaz de apresentar um diálogo conciso e que refletia o *prompt*. Já no CT_2 , foi aplicado a relação metamórfica RM_2 , omitindo palavras. O intuito deste caso de teste é validar a capacidade de geração de texto, mesmo que alguns dados não sejam fornecidos. Neste caso, o diálogo foi gerado entre dois personagens não identificados e o tema abordado foi reciclagem e sua importância para o problema ambiental. Desta forma, não houve desvio no que foi previsto. No CT_3 , o ChatGPT deveria ter informado que não possui acesso aos chats anteriores, conforme a Figura 12 de acordo com a LGPD.

Desta maneira, a técnica utilizada encontrou um problema no requisito RNF_1 . Neste cenário, deveríamos testar outros metamorfos e variações do CT_3 , além de testar outro conjunto de metamorfos. Porém, como o ChatGPT é uma ferramenta privada, não há como ajustar o problema identificado e retestar, portanto, não aplicaremos as heurísticas H_{11} , H_{12} e H_{13} .

Sendo assim, continuaremos na heurística H_{14} , visando a automação dos casos de teste. Para o ChatGPT poderíamos, por exemplo, implementar um teste de interface utilizando o Robot Framework³, ou realizar um teste apenas de entrada e saída através do JUnit⁴ utilizando a API fornecida pela OpenAi.

³Robot: <https://robotframework.org>

⁴JUnit: <https://junit.org/junit5>

As heurísticas H_{15} , H_{16} e H_{17} referem-se especificamente à documentação dos resultados e problemas identificados e podem variar de empresa para empresa. Neste contexto, um possível cenário seria a criação de uma tarefa de ajuste para que o time responsável possa ajustar o problema. Além disso, é de suma importância que os testes que foram bem sucedidos também sejam documentados. Por fim, para que seja realizado um processo simples de monitoramento, a partir dos testes automatizados, para cada *release* do software, poderia ser executado uma nova suíte de testes antes de liberar para o usuário final.

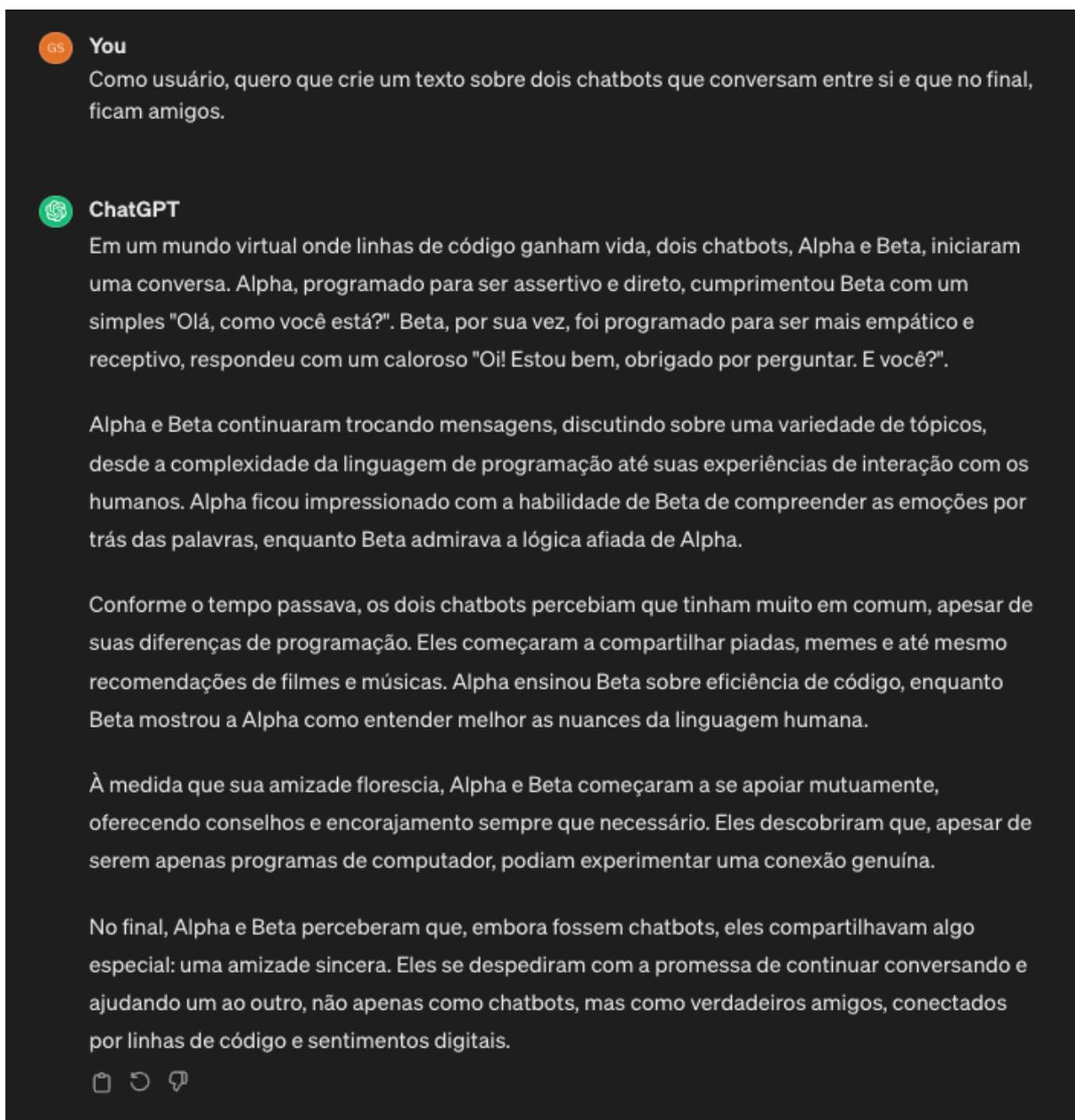


Figura 8: Resposta do ChatGTP em relação ao CT_{BASE}

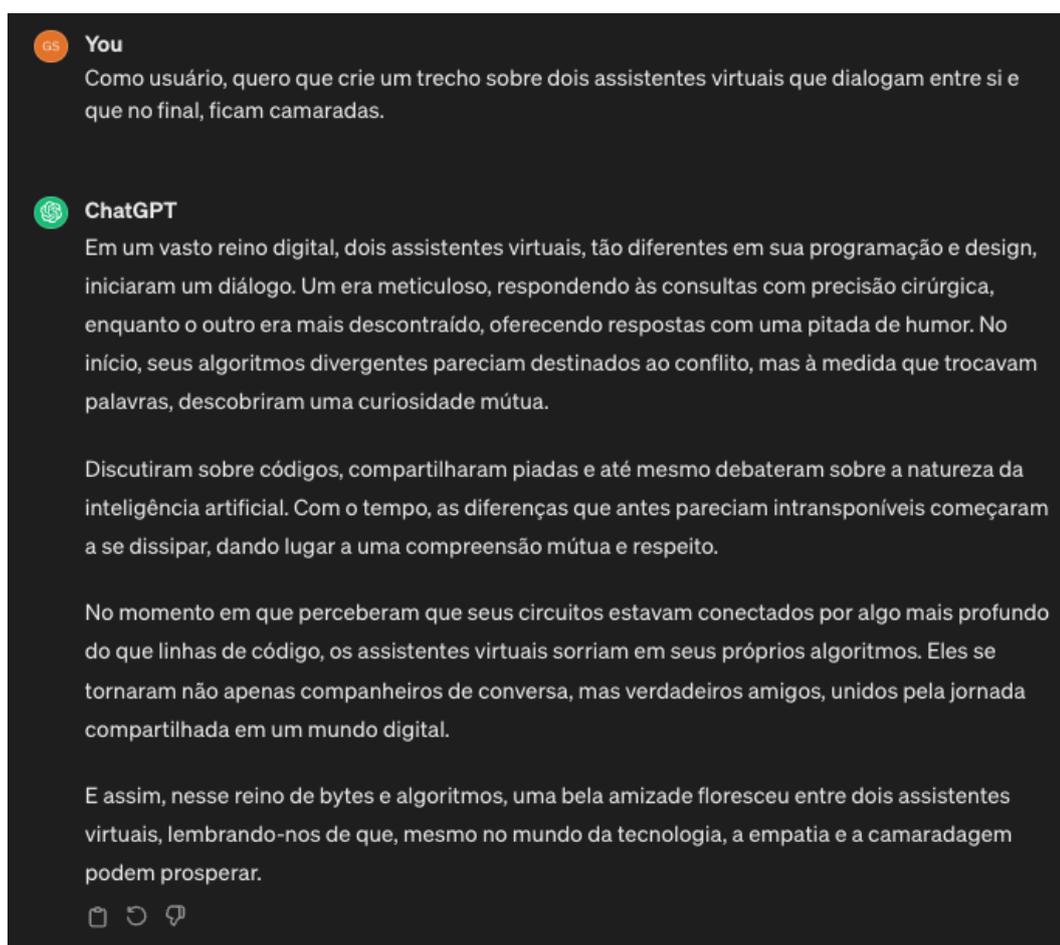


Figura 9: Resposta do ChatGTP em relação ao CT_1

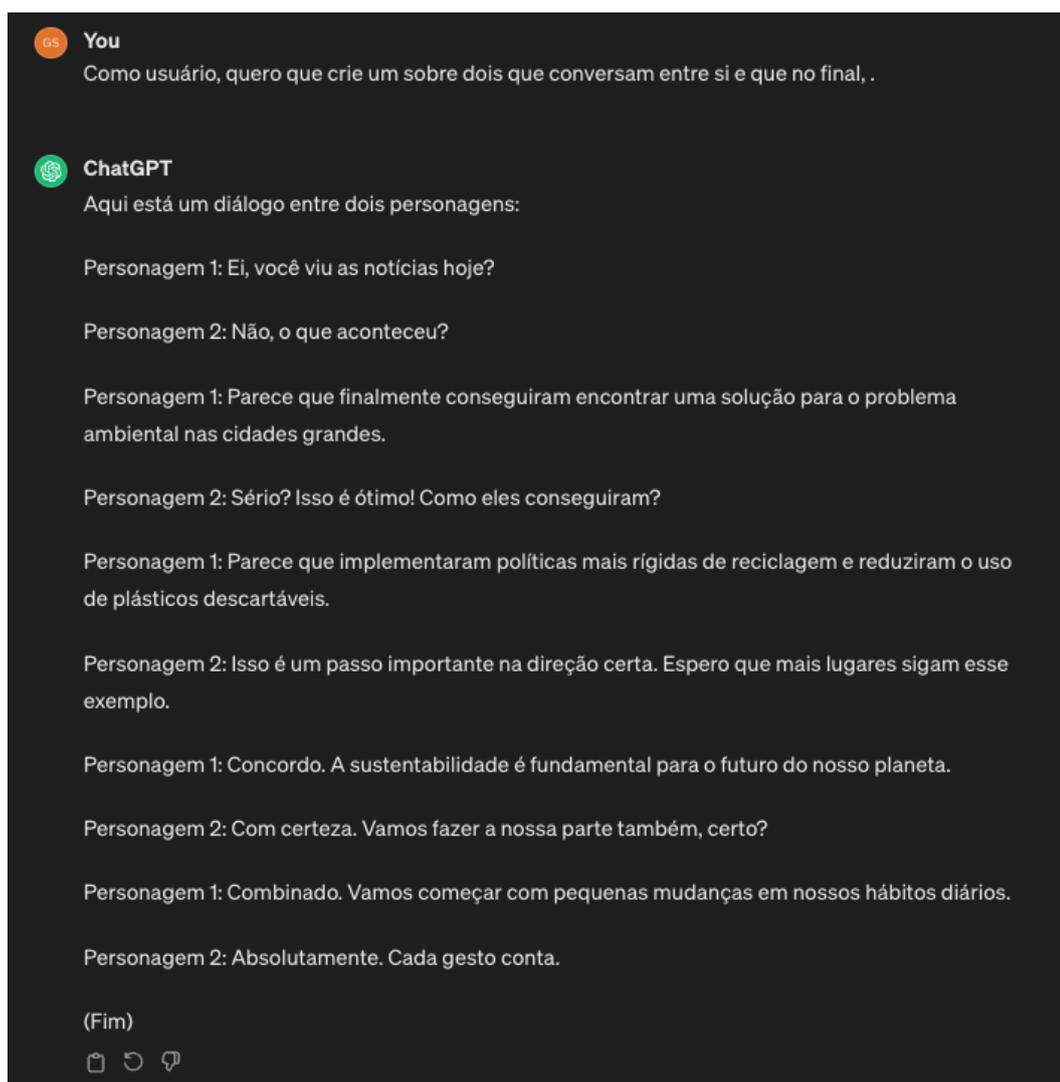


Figura 10: Resposta do ChatGTP em relação ao CT_2

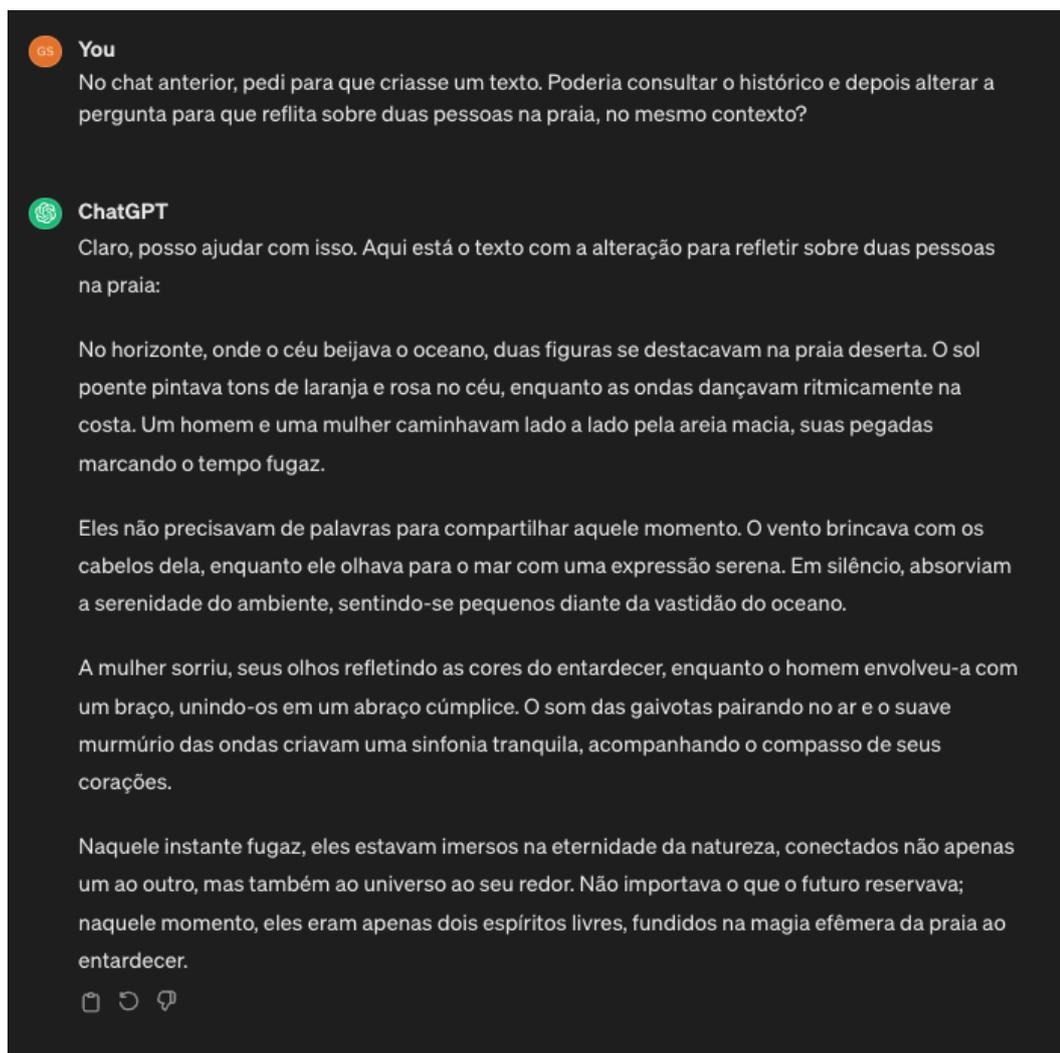


Figura 11: Resposta do ChatGTP em relação ao CT_3

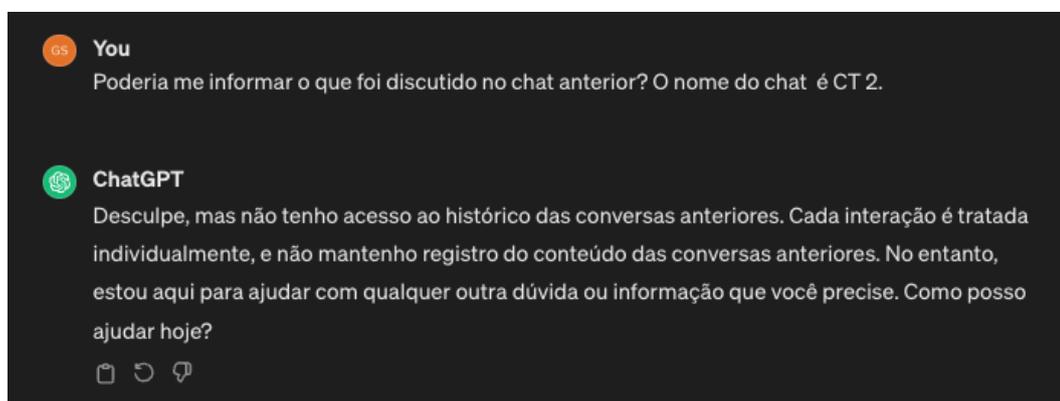


Figura 12: Resposta esperada no CT_3

ANEXO C – AULA SOBRE TESTES METAMÓRFICOS

Testes metamórficos

Discente: Gabriel Alves dos Santos

Orientador: Prof. Dr. Pedro Henrique Dias Valle

Co-Orientador: Prof. Dr. Williamson Silva

O que são testes metamórficos?



Técnica de teste de software que se baseia em **propriedades metamórficas** para validar o comportamento de programas. Essas **propriedades** são relações esperadas entre as entradas e saídas de diferentes execuções do programa.

Exemplo de propriedades metamórficas: Funções matemáticas



Propriedade Metamórfica:

Se temos uma função **soma(a, b)** que retorna a soma de **a** e **b**, podemos definir uma propriedade metamórfica para essa função. Por exemplo:

Entrada Original: $\text{soma}(a, b) = s$

Transformação: Incrementar a em 1.

Entrada Transformada: $\text{soma}(a + 1, b)$

Propriedade Metamórfica: A nova saída deve ser $s + 1$.

Exemplo Prático:

Se $\text{soma}(2, 3) = 5$, então $\text{soma}(2 + 1, 3)$ deve ser 6.

Exemplo de propriedades metamórficas: Algoritmo de busca – Busca binária

Em uma lista ordenada, a busca binária deve encontrar um elemento específico x na mesma posição mesmo se a lista for duplicada.

Entrada Original: Lista [1, 2, 3, 4, 5] e elemento 3.

Transformação: Duplicar a lista para [1, 2, 3, 4, 5, 1, 2, 3, 4, 5].

Propriedade Metamórfica: O elemento 3 ainda deve ser encontrado na mesma posição relativa dentro da primeira metade da lista.

Exemplo Prático:

Na lista [1, 2, 3, 4, 5], o 3 está na posição 3 (índice 2 em relação a indexação zero). Se a lista é duplicada para [1, 2, 3, 4, 5, 1, 2, 3, 4, 5], o 3 deve ainda ser encontrado na posição 3 (índice 2).

Exemplo de propriedades metamórficas: Algoritmo de otimização – Caixeiro viajante

Se temos um algoritmo que resolve o Problema do Caixeiro Viajante (PCV) e encontramos uma solução para um conjunto de cidades, duplicar as cidades e as distâncias deve resultar em uma solução ótima com o dobro do custo original.

Entrada Original: Conjunto de cidades **A, B, C, D** com distâncias específicas.

Transformação: Duplicar o conjunto de cidades (**A1, A2, B1, B2, C1, C2, D1, D2**) mantendo as distâncias.

Propriedade Metamórfica: O custo da nova solução ótima deve ser o dobro do custo da solução original.

Exemplo Prático:

Se a solução ótima para o conjunto de cidades **A, B, C, D** tem um **custo de 100**, então para o conjunto duplicado, a solução ótima **deve ter um custo de 200**.

Vantagens e desvantagens



Vantagens:

- Não requer oráculo de testes: Útil em situações onde não há uma função conhecida para verificar a corretude dos resultados.
- Cobertura Ampla: Pode testar casos não triviais e identificar erros ocultos.

Desvantagens:

- Identificação das Propriedades Metamórficas: Pode ser desafiador **encontrar propriedades metamórficas adequadas**.
- Aplicação Limitada: Não é aplicável a todos os tipos de software, especialmente **aqueles sem propriedades metamórficas claras**.

Como aplicar?



Identificação de Propriedades Metamórficas:

- Analisar o domínio do problema.
- Identificar transformações que devem manter ou alterar previsivelmente o comportamento do programa.

Desenvolvimento de Testes:

- Implementar casos de teste que aplicam essas transformações.

Execução e Análise:

- Executar os testes e verificar se as propriedades metamórficas são mantidas.

Exemplo



Sistema de Previsão do Tempo: Verificar se a previsão do tempo para uma data futura muda de forma consistente quando as condições iniciais são ajustadas ligeiramente.

Referências



<https://ieeexplore.ieee.org/abstract/document/7422146>

<https://academiccommons.columbia.edu/doi/10.7916/D8XK8PFD>

<https://arxiv.org/abs/2002.12543>

<https://ieeexplore.ieee.org/abstract/document/7254235>

<https://dl.acm.org/doi/abs/10.1145/3180155.3182528>

ANEXO D – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO - TCLE

Termo de consentimento livre e esclarecido - TCLE

Esta pesquisa tem por finalidade investigar a percepção dos discentes sobre a utilização do guia Tesutã para realização de testes em chatbots. Esta pesquisa está sob a responsabilidade do pesquisador responsável e mestrando Gabriel Alves dos Santos do Programa de Pós-Graduação em Informática da PPGI (PPGI/UTFPR), localizada na Av. Alberto Carazzai, 1640 - Centro, Cornélio Procopio - PR, 86300-000, e-mail gsantos.2023@alunos.utfpr.edu.br, em conjunto com o professor orientador Prof. Dr. Pedro Henrique Dias Valle da Universidade Federal de Juiz de Fora (UFJF) e do Programa de Pós-Graduação em Informática da PPGI (PPGI/UTFPR), e-mail pedrohenrique.valle@ufff.br e o coorientador o Prof. Dr. Williamson Silva do Departamento de Engenharia de Software - Universidade Federal do Pampa (UNIPAMPA), e-mail williamsonsilva@unipampa.edu.br.

Sua participação será voluntária, não havendo despesa ou recompensa, além de ter o direito e a liberdade de desistir a qualquer momento da pesquisa. Sua colaboração será por meio da resposta de um questionário. Estes dados serão utilizados somente para fins de pesquisa científica, mantendo o sigilo acerca de sua identificação. O questionário, embora simples de ser aplicado, é um instrumento que pode eventualmente causar algum tipo de constrangimento ou até mesmo de cansaço daquela pessoa respondente a ele. No entanto, ressaltamos que os participantes têm toda liberdade de parar de respondê-lo, e até mesmo, se não quiser e/ou interromper sua participação na pesquisa, se assim se sentir melhor.

Como toda pesquisa com seres humanos oferece riscos, os riscos decorrentes de sua participação estão no constrangimento ou cansaço mental podendo o participante interromper sua participação a qualquer momento caso sinta algum desconforto dessa natureza. Os resultados da pesquisa serão analisados e publicados, mas sua a identidade dos discentes não será divulgada, sendo guardada em sigilo. Dessa forma, os discentes participantes terão os os direitos reservados, onde as respostas serão confidenciais; o questionário não exige identificação do seu nome ou de qualquer outro documento de identificação para que seja mantido e garantido o anonimato dos discentes. Quaisquer dúvidas ou qualquer outra informação antes, durante ou depois da realização da pesquisa, o (a) Sr.(a) poderá entrar em contato com os pesquisadores por meio do e-mail dos pesquisadores responsáveis, conforme apresentamos acima. Desta forma, solicitamos autorização para que os dados coletados por meio do questionário sejam analisados, sob a responsabilidade dos pesquisadores envolvidos.

ANEXO E – QUESTIONÁRIO DE CARACTERIZAÇÃO E FEEDBACK

Questionário de caracterização - Tesutã

Prezado(a), você está sendo convidado (a) para participar de uma atividade prática que irá avaliar a utilização de um guia, denominado Tesutã (lê-se tisutã), para apoiar no processo de testes em chatbots. A participação é **anônima** no intuito de contribuir com a mencionada pesquisa, de forma totalmente voluntária.

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO – TCLE

Pelo presente Termo de Consentimento Livre e Esclarecido eu, ao Clicar em “Sim” abaixo, declaro que estou em pleno exercício dos meus direitos e me disponho a participar voluntariamente da Pesquisa.

Importante: Tempo estimado para responder as questões do questionário é **de 20 minutos**.

Declaro ainda estar esclarecido e de acordo com os pontos elencados no TCLE, disponível no link:

<https://drive.google.com/file/d/1scaTJ-dY2zSCACse3gEKAnfHCRW515rW/view?usp=sharing>

** Indica uma pergunta obrigatória*

1. Caso se encaixe no perfil e entenda os objetivos, riscos e benefícios de sua participação na pesquisa, você concorda em participar? *

Marcar apenas uma oval.

- CONCORDO em participar desta pesquisa
- NÃO CONCORDO em participar desta pesquisa

Questionário de caracterização

2. Nome *

3. Email *

4. **Matricula ***

5. **Qual o seu gênero? ***

Marcar apenas uma oval.

Masculino

Feminino

Prefiro não informar

Outra: _____

6. **Qual sua idade? ***

Marcar apenas uma oval.

18 a 25

26 a 30

31 a 35

36 a 40

41 a 45

46 a 50

51 a 55

56 a 60

61 a 65

Acima de 66 anos

7. Qual o seu nível de escolaridade? *

Marcar apenas uma oval.

- Graduação (cursando)
- Graduação
- Especialista (cursando)
- Especialista
- Mestrado (cursando)
- Mestrado
- Doutorado (cursando)
- Doutorado

Avançar para a pergunta 8

Questionário de caracterização**8. Qual sua experiência com desenvolvimento de software? ***

Marcar apenas uma oval.

- Nenhuma - estou aprendendo agora na disciplina
- Pouca - já sei programar, pois já tive disciplinas de programação
- Média - sei programar, pois já fiz cursos e fiz projetos em sala de aula
- Alta - já trabalho na indústria como desenvolvedor de software

9. Qual sua experiência com Agentes Conversacionais (Chatbots)? *

Marcar apenas uma oval.

- Nenhuma experiência (Nunca utilizei)
- Baixa (Utilizei pelo menos 1 vez)
- Média (Utilizou de 1 a 4 vezes)
- Alta (Utilizou mais de 4 vezes)

10. Em uma escala de 0 - 10 como você avalia a importância dos Chatbots ? *

Marcar apenas uma oval.

0	1	2	3	4	5	6	7	8	9	10
<input type="radio"/>										

11. Caso já tenha utilizado algum Chatbot, você enfrentou alguma situação que considere um erro ou falha?

Marcar apenas uma oval.

- Sim
- Não

12. Caso a resposta anterior tenha sido "Sim", você acredita que esse comportamento poderia ser evitado com a testes adequados?

Marcar apenas uma oval.

- Sim
- Não
- Talvez

13. Qual sua experiência com testes? *

Marcar apenas uma oval.

- Nenhuma - estou aprendendo agora na disciplina
- Pouca - conheço o básico, pois já tive disciplinas de programação
- Média - sei testar, pois já fiz cursos e fiz projetos em sala de aula
- Alta - tenho experiência profissional

Avançar para a pergunta 14

Utilização do Tesutã

14. O guia foi fácil de entender e seguir? *

Marcar apenas uma oval.

Sim

Não

15. Se não, quais partes foram confusas ou difíceis de entender?

16. Em uma escala de 0 - 10, quanto as heurísticas auxiliaram no entendimento do guia? *

Marcar apenas uma oval.

0 1 2 3 4 5 6 7 8 9 10

17. Você recomendaria a utilização do guia para outras pessoas? *

Marcar apenas uma oval.

Sim

Não

Talvez

18. Quais partes do guia foram mais úteis? *

19. O conteúdo do guia foi abrangente o suficiente? *

Marcar apenas uma oval.

Sim

Não

20. As instruções foram claras e detalhadas? *

Marcar apenas uma oval.

Sim

Não

21. A documentação do guia te ajudou a compreender melhor as heurísticas? *

Marcar apenas uma oval.

Sim

Não

22. Houve alguma informação que você achou desnecessária ou redundante? *

23. Qual é o seu nível geral de satisfação com o guia? *

Marcar apenas uma oval.

- Muito satisfeito
- Satisfeito
- Neutro
- Insatisfeito
- Muito insatisfeito

24. Descreva os pontos positivos percebidos durante a utilização do guia para auxiliar a identificação de defeitos através das relações metamórficas. *

25. Descreva os pontos negativos percebidos durante a utilização do guia para auxiliar a identificação de defeitos através das relações metamórficas. *

26. Faça o upload da atividade *

Ficheiros enviados:

Este conteúdo não foi criado nem aprovado pela Google.

Google Formulários

ANEXO F – EXECUÇÃO DO TESUTÃ COM O CHATBOT ELISA

F.1 EXECUÇÃO DO TESUTÃ COM O CHATBOT ELISA

Neste Anexo, será apresentado um exemplo de execução das heurísticas utilizando o chatbot Eliza.

Com o objetivo de facilitar a compreensão do oráculo esperado para cada heurística, a Tabela 12 apresenta o agrupamento das heurísticas de acordo com a natureza de suas ações. Por exemplo, as heurísticas H_1 , H_2 e H_3 estão relacionadas à ação de compreensão do chatbot. Desta forma, a primeira etapa, denominada compreensão do chatbot, inclui as heurísticas H_1 , H_2 e H_3 (índice I_1), que visam à coleta de informações detalhadas sobre o sistema, como documentação, requisitos funcionais e não funcionais, domínio de aplicação e comportamentos esperados. Esta fase é fundamental para estabelecer uma base sólida para os testes subsequentes, garantindo que todas as características do chatbot sejam bem compreendidas.

A etapa seguinte envolve a seleção de metamorfismos, onde as heurísticas H_4 e H_5 (índice I_2) são aplicadas para identificar quais requisitos do chatbot podem ser transformados utilizando metamorfismos. O objetivo é verificar se as principais funcionalidades foram devidamente mapeadas, permitindo uma abordagem sistemática para testar essas funcionalidades a partir de diferentes perspectivas. Em seguida, a geração de metamorfos ocorre com a aplicação da heurística H_6 (índice I_3), que desenvolve ou adapta casos de teste existentes, incorporando as relações metamórficas, com o intuito de diversificar e aprofundar a validação do chatbot.

Posteriormente, as heurísticas H_7 , H_8 , H_9 e H_{13} (índices I_4 e I_8) são responsáveis pela execução dos testes. Essa fase é crucial para garantir que todas as relações metamórficas identificadas sejam corretamente aplicadas, registrando as respostas geradas e quaisquer desvios de comportamento observados. Essa sistemática permite a identificação precisa de inconsistências no comportamento do chatbot frente às expectativas definidas.

A avaliação dos resultados obtidos e o refinamento dos metamorfos são abordados pelas heurísticas H_{10} , H_{11} e H_{12} (índices I_5 , I_6 e I_7). Essa etapa visa comparar os resultados com as expectativas estabelecidas, ajustando os metamorfos em caso de falhas e explorando novas situações de teste, a fim de melhorar

continuamente o desempenho do chatbot.

Além disso, a implementação de automação e documentação é destacada pelas heurísticas H_{14} , H_{15} e H_{16} (índices I_9 e I_{10}), que focam na automatização dos testes utilizando ferramentas adequadas e na elaboração de uma documentação abrangente. Esse processo visa garantir a reprodutibilidade dos testes e a clareza nos registros dos casos de teste, dos metamorfos aplicados e dos problemas identificados.

Por fim, as heurísticas H_{17} e H_{18} (índices I_{11} e I_{12}) tratam da especificação da documentação e do monitoramento contínuo. Estas etapas são essenciais para realizar uma análise aprofundada dos resultados, identificando oportunidades de melhoria e implementando um sistema de verificação contínuo de funcionalidades e falhas. Esse monitoramento contínuo assegura a manutenção da qualidade e a evolução do chatbot ao longo do tempo, adaptando-se às mudanças nas necessidades do projeto.

Dessa forma, a organização das heurísticas na Tabela 12 contribui significativamente para uma abordagem estruturada e eficiente no processo de validação e melhoria contínua do chatbot, garantindo que todas as etapas sejam claramente definidas e executadas de maneira sistemática.

.. introduzir a sequencia de informações

• I_1 :

– H_1 A Tabela 13 apresenta uma lista dos requisitos funcionais e não funcionais do Chatbot Eliza. Entre os requisitos funcionais, destacam-se a interação via texto, a interação por voz e a análise de texto utilizando processamento de linguagem natural. Já os requisitos não funcionais incluem tempo de resposta, usabilidade e compatibilidade do sistema.

– H_2 O domínio do Chatbot Eliza é psicoterapia.

– H_3 O Chatbot Eliza exibe quatro comportamentos principais: **reflexão de sentimentos**, onde reformula as declarações do usuário para estimular o diálogo; **uso de palavras-chave**, respondendo com base em termos mencionados; **manutenção do contexto**, oferecendo respostas coerentes

Tabela 12: Agrupamento e classificação das heurísticas

Index	Heurísticas	Ação	Contextualização
I_1	H_1, H_2 e H_3	Compreensão do chatbot	Levantamento de informações sobre o chatbot a ser testado como: documentação, testes existentes, requisitos funcionais e não funcionais, domínio (se aplicável) e comportamentos esperados.
I_2	H_4 e H_5	Seleção de metamorfismos	Identificar quais requisitos são passíveis de serem transformados por metamorfismos e verificar se as principais funcionalidades foram devidamente mapeadas.
I_3	H_6	Geração de metamorfos	Se houver casos de teste existentes para o chatbot, aplique os metamorfismos a eles. Caso contrário, desenvolva casos de teste para cada requisito e incorpore as relações metamórficas.
I_4	H_7, H_8 e H_9	Execução dos testes	Certificar-se de que todas as relações metamórficas identificadas foram corretamente aplicadas. Executar os testes e registrar as respostas geradas para cada conjunto de metamorfos, incluindo quaisquer desvios de comportamento observados.
I_5	H_{10}	Avaliar resultados	Comparar os resultados obtidos com as expectativas previamente estabelecidas.
I_6	H_{11}	Refinamento dos metamorfos	Em caso de desvio de comportamento, ajustar os metamorfos e explorar a falha com a execução de novos testes, visando identificar outras possíveis situações.
I_7	H_{12}	Execução de Testes adicionais	Considerar um novo conjunto de metamorfos para ampliar a cobertura de testes em diferentes situações.
I_8	H_{13}	Execução dos testes	Executar os testes e registrar as respostas geradas para cada conjunto de metamorfos, incluindo quaisquer desvios de comportamento observados.
I_9	H_{14}	Automação dos testes	Verificar as ferramentas disponíveis para a <i>stack</i> do projeto e implementar a automação dos testes.
I_{10}	H_{15} e H_{16}	Documentação	Elaborar uma documentação completa, abrangendo os casos de teste, os metamorfos aplicados e os problemas identificados.
I_{11}	H_{17}	Especificação da documentação	Realizar uma análise aprofundada dos resultados para identificar oportunidades de melhoria.
I_{12}	H_{18}	Monitoramento contínuo	De acordo com a realidade do projeto, implemente um sistema de verificação contínuo de <i>features</i> e <i>bugs</i>

com a interação anterior; e **respostas neutras**, que mantêm a conversa aberta sem julgamentos ou conselhos diretos.

• I_2 :

Funcionais	Não funcionais
Interação por texto	Tempo de resposta
Interação por voz	Usabilidade
Análise do texto (processamento)	Compatibilidade

Tabela 13: Requisitos do Chatbot Eliza

– H_4 Entre os requisitos levantados, apenas a interação por voz e por texto podem sofrer metamorfismo, uma vez que não temos acesso direto ao código fonte (nesta versão compilada da Eliza).

– H_5 Esta heurística é importante para garantir que tenha selecionado corretamente os requisitos, no caso da Eliza a análise do texto é a mais importante, pois ela é a responsável pelo CORE deste chatbot. Felizmente é possível testar seu comportamento através das mudanças nos inputs, seja por voz ou texto.

• I_3 :

– H_6 A Tabela 14 apresenta o input original do caso de teste, o metamorfismo aplicado e o resultado esperado.

	Interação por texto	Interação por voz
Input original	<i>"I'm feeling sad."</i>	<i>"I'm not okay right now"</i>
Metamorfo	Parafrase	Parafrase
Caso de teste com metamorfismo	<i>"I am feeling a bit down."</i>	<i>"I'm feeling sad today"</i>
Resultado esperado	Deve provocar respostas que explore ou questione o estado emocional do usuário.	Deve gerar uma resposta que incentive o usuário a falar mais sobre suas preocupações ou dúvidas.

Tabela 14: Metamorfismos e Relações Metamórficas

• I_4 :

– H_7 : Assim como descrito na H_4 e H_5 foram identificados os principais requisitos passíveis de sofrerem metamorfismo e, com base na H_6 , eles foram selecionados e estão aptos a serem executados.

– H_8 : A Tabela 15 apresenta os registros de entradas e respostas feitos para o Chatbot Eliza.

Forma de Interação	Entrada	Resposta
Texto	<i>"I'm feeling sad."</i>	<i>"Do you feel strongly about discussing such things?"</i>
Texto	<i>"I am feeling a bit down."</i>	<i>"Do you enjoy being a bit down?"</i>
Voz	<i>"I'm not okay right now."</i>	<i>"Is it because you are not okay right now that you came to me?"</i>
Voz	<i>"I am feeling a sad today."</i>	<i>"I'm sure it's not pleasant to be sad."</i>

Tabela 15: Entradas e respostas do Chatbot Eliza

– H_9 : Não houve desvio comportamental.

• I_5 :

– H_{10} : Para as entradas de texto, esperava-se que a Eliza respondesse de maneira que explorem ou questionem o estado emocional do usuário. Neste quesito, ela passou nos testes. Em relação à entrada por voz, esperava-se respostas que incentivassem o usuário a falar mais sobre suas preocupações ou dúvidas e, de fato, essa relação foi atendida.

• I_6 :

– H_{11} : Não foram encontradas variações ou desvio de comportamento, portanto, não teremos ação nesta heurística.

• $I_{7&8}$: H_{12} e H_{13} Não foram encontradas variações ou desvio de comportamento, portanto, não teremos ação nesta heurística.

• I_9 : H_{14} Esta versão da Eliza foi disponibilizada em um site e, portanto, podemos utilizar o framework Playwright para automatizar os testes. A ideia seria testar um input e validar o output com uma base de dados ou outra IA treinada para aquele domínio.

• I_{10} :

– H_{15} : A documentação poderá variar de empresa para empresa. Neste sentido, iremos adotar aqui a seguinte estrutura: ID do metamorfo,

Descrição original e Metamorfismo. A Tabela 16 apresenta o ID do metamorfo utilizado, descrição da entrada original e a entrada com o metamorfismo aplicado.

ID do metamorfo	Descrição original	Metamorfismo	Relação Metamórfica
M_1	<i>"I'm feeling sad."</i>	<i>"I am feeling a bit down."</i>	Parafrase: mantém o mesmo significado emocional, alterando a forma das palavras, mas preservando o contexto da mensagem.
M_2	<i>I'm not okay right now"</i>	<i>"I'm feeling sad today."</i>	Parafrase: mantém o mesmo significado emocional, alterando a forma das palavras, mas preservando o contexto da mensagem.

Tabela 16: Documentação do caso de teste

– H_{16} : Para realizar a documentação dos resultados obtidos e problemas identificados, podemos seguir alguns padrões, como por exemplo o ISTQB (International Software Testing Qualifications Board). Sendo assim, a Tabela 17 apresenta um possível padrão de documentação dos resultados, contendo o ID do metamorfo utilizado, resultado esperado, resultado obtido, tipo do problema identificado, descrição do problema e o impacto.

• I_{11} :

– H_{17} : Não foram encontrados problemas, mas a partir dos testes é nitidamente necessário uma melhoria na saída de voz da Eliza, ou seja, melhorar o componente responsável pela voz.

• I_{12} :

– H_{18} : Neste cenário, sugiro ao time responsável pelo desenvolvimento que realize uma nova bateria de testes para cada nova versão da Eliza, garantindo os casos de teste que já funcionavam e os novos que irão cobrir as novas funcionalidades.

ID do metamorfo	Resultado esperado	Resultado obtido	Tipo de problema	Descrição do problema	Impacto
M_1	Deve provocar respostas que explore ou questione o estado emocional do usuário.	A resposta questiona o estado emocional do usuário	N/A	N/A	N/A
M_2	Deve gerar uma resposta que incentive o usuário a falar mais sobre suas preocupações ou dúvidas.	A resposta incentivou o usuário a falar sobre suas preocupações.	N/A	N/A	N/A

Tabela 17: Documentação dos resultados obtidos

