

UNIVERSIDADE FEDERAL DE ALAGOAS
INSTITUTO DE COMPUTAÇÃO
PROGRAMA DE PÓS GRADUAÇÃO EM INFORMÁTICA

Felipe Moura de Jesus

**Design Thinking no Desenvolvimento de Software: Indícios e
Evidências na Indústria e na Literatura**

Maceió - AL

2024

Felipe Moura de Jesus

Design Thinking no Desenvolvimento de Software: Indícios e Evidências na Indústria e na Literatura

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal de Alagoas, como requisito para obtenção do grau de Mestre em Informática.

Orientador: Prof. Dr. Rodrigo Gusmão de Carvalho Rocha

Maceió - AL

2024

Catálogo na fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico

Bibliotecária: Helena Cristina Pimentel do Vale – CRB4/661

J58d Jesus, Felipe Moura de.
Design thinking no desenvolvimento de software : indícios e evidências na indústria e na literatura / Felipe Moura de Jesus. – 2024.
96 f.: il.

Orientador: Rodrigo Gusmão de Carvalho Rocha.
Dissertação (mestrado em Informática) – Universidade Federal de Alagoas, Instituto de Computação. Maceió, 2024.

Bibliografia: f. 78-83.
Apêndices: f. 84-96.

1. Design thinking. 2. Desenvolvimento de software. 3. Engenharia de sistemas. computacionais. 4. Mapeamento sistemático. 5. Método Survey. I. Título.

CDU: 004.4



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE ALAGOAS
INSTITUTO DE COMPUTAÇÃO
Av. Lourival Melo Mota, S/N, Tabuleiro do Martins, Maceió - AL, 57.072-970
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO (PROPEP)
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Folha de Aprovação

FELIPE MOURA DE JESUS

DESIGN THINKING NO DESENVOLVIMENTO DE SOFTWARE: INDÍCIOS E EVIDÊNCIAS NA INDÚSTRIA E NA LITERATURA

DESIGN THINKING IN SOFTWARE DEVELOPMENT: INDUSTRY AND LITERATURE EVIDENCE

Dissertação submetida ao corpo docente do Programa de Pós-Graduação em Informática da Universidade Federal de Alagoas e aprovada em 31 de outubro de 2024.

Banca Examinadora:

Documento assinado digitalmente
 **RODRIGO GUSMÃO DE CARVALHO ROCHA**
Data: 04/11/2024 18:15:37-0300
verifique em <https://validar.itl.gov.br>

Prof. Dr. RODRIGO GUSMÃO DE CARVALHO ROCHA
UFAPE – Universidade Federal do Agreste de Pernambuco
Orientador

Documento assinado digitalmente
 **RIAN GABRIEL SANTOS PINHEIRO**
Data: 11/11/2024 22:20:41-0300
verifique em <https://validar.itl.gov.br>

Prof. Dr. RIAN GABRIEL SANTOS PINHEIRO
UFAL – Instituto de Computação
Examinador Interno

Documento assinado digitalmente
 **IGOR MEDEIROS VANDERLEI**
Data: 12/11/2024 11:00:03-0300
verifique em <https://validar.itl.gov.br>

Prof. Dr. IGOR MEDEIROS VANDERLEI
UFAPE – Universidade Federal do Agreste de Pernambuco
Examinador Externo

Dedico este trabalho a todos que contribuíram para minha jornada acadêmica e pessoal. Agradeço especialmente a Nathália Teles pelo apoio, pelos conselhos valorosos, à minha mãe Nilde Moura pelo incentivo à educação, aos amigos pela motivação, e ao orientador Rodrigo G. de Carvalho Rocha pelos ensinamentos fundamentais para minha formação.

AGRADECIMENTOS

Gostaria de agradecer primeiramente a Deus, pois sempre me sustentou e guiou os meus passos. Agradeço à minha esposa Nathália, que me incentivou a iniciar essa jornada, me auxiliou e me apoiou, compreendendo momentos de dedicação e estudos até o fim dessa jornada. Agradeço à minha família, Neilde (Mãe) e Jessica (Irmã), por sempre estarem comigo e me apoiarem tanto em decisões pessoais quanto profissionais. A todos os amigos que oraram e deram palavras de apoio durante a minha vivência acadêmica.

À Deus que em sua abundante graça, nunca me desamparou.

RESUMO

Em um cenário cada vez mais globalizado e motivado pela tecnologia, a indústria de software busca atuar com máxima eficiência na criação de produtos inovadores alinhados com as expectativas dos usuários. Neste contexto, o Design Thinking (DT) tornou-se uma ferramenta fundamental para promover a inovação e melhoria da qualidade de software, pois foca na experiência do usuário com abordagens que proporcionam multidisciplinaridade, prototipação e colaboração ativa das partes interessadas. Nesse sentido, vários autores investigam como o DT é utilizado em cenários reais de desenvolvimento de software e quais são as partes essenciais neste contexto. O objetivo desta pesquisa é apresentar um conjunto de indícios e evidências sobre DT no contexto de desenvolvimento de software, considerando as perspectivas da indústria e da literatura. Identificando e mapeando as principais abordagens do uso do Design Thinking na produção de sistemas computacionais. A metodologia utilizada neste trabalho se fundamentou em dois estudos experimentais fundamentados em questões de pesquisa, o mapeamento sistemático, que explorou a literatura sob as diretrizes de Kitchenham, utilizando três bases científicas, IEEE, ACM e SCOPUS, e, após verificação e análise dos resultados, foram identificados 95 estudos primários. Em seguida, foi aplicada uma pesquisa com 105 profissionais de tecnologia de diferentes organizações, buscando coletar de profissionais da área dados relevantes sobre o contexto de Design Thinking. Este estudo evidenciou um conjunto de 116 práticas, 45 modelos, 60 papéis, 75 ferramentas e 28 desafios associados a adoção do Design Thinking no desenvolvimento de software. Esses resultados podem ajudar profissionais e pesquisadores da área a entender melhor os desafios e implementar soluções mais eficazes para melhorar suas tarefas no desenvolvimento de software usando abordagens de Design Thinking. Dessa forma, este trabalho contribui com recursos importantes para o ecossistema de construção de software, permitindo o compartilhamento de conhecimento sobre os principais elementos de DT no desenvolvimento de software.

Palavras-chaves: Design Thinking. Desenvolvimento de Software. Engenharia De Sistemas Computacionais. Mapeamento Sistemático. Survey

ABSTRACT

In an increasingly globalized and technology-driven scenario, the software industry seeks to act with maximum efficiency in creating innovative products aligned with user expectations. In this context, Design Thinking (DT) has become a fundamental tool for promoting innovation and improving software quality, as it focuses on the user experience with approaches that provide multidisciplinary, prototyping and active collaboration of stakeholders. In this sense, several authors investigate how DT is used in real software development scenarios and what are the essential parts in this context. The objective of this research is to present a set of indications and evidence about DT in the context of software development, considering the perspectives of the industry and the literature. Identifying and mapping the main approaches to the use of Design Thinking in the production of computer systems. The methodology used in this work was based on two experimental studies based on research questions, systematic mapping, which explored the literature under Kitchenham's guidelines, using three scientific databases, IEEE, ACM and SCOPUS, and, after verification and analysis of the results, 95 primary studies were identified. Then, a survey was applied to 105 technology professionals from different organizations, seeking to collect relevant data from professionals in the area about the context of Design Thinking. This study revealed a set of 116 practices, 45 models, 60 roles, 75 tools and 28 challenges associated with the adoption of Design Thinking in software development. These results can help professionals and researchers in the area to better understand the challenges and implement more effective solutions to improve their tasks in software development using Design Thinking approaches. In this way, this work contributes important resources to the software construction ecosystem, allowing the sharing of knowledge about the main elements of DT in software development.

Keywords: Design Thinking. Software development. Computer Systems Engineering. Systematic Mapping. Survey

LISTA DE ILUSTRAÇÕES

Figura 1 – Fases do Design Thinking	20
Figura 2 – Exploração do Espaço de Problemas e Soluções no Design Thinking	21
Figura 3 – Processo de experimentação	22
Figura 4 – Fluxo de etapas da pesquisa	25
Figura 5 – Etapas do mapeamento sistemático	26
Figura 6 – Classificação da qualidade dos artigos selecionados	44
Figura 7 – Número de publicações por ano dos artigos relacionados ao DT	44
Figura 8 – Tamanho da empresa dos participantes	45
Figura 9 – Distribuição de papéis	46
Figura 10 – Experiência dos participantes com o desenvolvimento de software	46
Figura 11 – Nível educacional dos participantes	47
Figura 12 – Composição dos times de tecnologia	47
Figura 13 – Nível de Conhecimento em Design Thinking	48
Figura 14 – Tempo de Experiência com Design Thinking na Produção de Software	48
Figura 15 – Adoção do Design Thinking na Produção de Software	49
Figura 16 – Fases do Desenvolvimento de Software com a abordagem Design Thinking	50
Figura 17 – Modelos e Metodologias Aplicados na Produção de Software	50
Figura 18 – Abordagens Empresariais para a Melhoria do Processo de Design Thinking	51
Figura 19 – Principais Desafios Enfrentados pelas Equipes na Utilização do Design Thinking	52
Figura 20 – Ferramentas Complementares ao Design Thinking	52
Figura 21 – Utilização das Práticas de Design Thinking em Diferentes Fases do Projeto	56
Figura 22 – Categorias de Projetos que Incorporam Design Thinking	57
Figura 23 – Papéis e Desafios na Adoção do Design Thinking	58
Figura 24 – Desafios Relacionadas a Outras Questões de Pesquisa	60
Figura 25 – Citação dos modelos nos estudos selecionados	61
Figura 26 – Citação dos Modelos Nos Estudos Selecionados	62
Figura 27 – Práticas Relacionadas a Outras Questões de Pesquisa	64
Figura 28 – Desafios Relacionadas a Outras Questões de Pesquisa	64
Figura 29 – Ferramentas Utilizadas por Diferentes Papéis no Desenvolvimento de Software com Design Thinking	65
Figura 30 – Desafios - Citações no Survey e Mapeamento Sistemático	67
Figura 31 – Práticas - Citações no Survey e Mapeamento Sistemático	68
Figura 32 – Modelos Integrados ao Design Thinking no Desenvolvimento de Software	69
Figura 33 – Design Thinking e Sua Adoção em Diversos Papéis	69
Figura 34 – Ferramentas Integradas ao Design Thinking no Desenvolvimento de Software	70

LISTA DE TABELAS

Tabela 1 – Artigos Derivados do Mestrado	17
Tabela 2 – Diferentes modelo de Design Thinking	19
Tabela 3 – String para as questões de pesquisa	28
Tabela 4 – Critérios de Inclusão/Exclusão	29
Tabela 5 – Resultados do processo de seleção dos estudos primários	36
Tabela 6 – Desafios na adoção do Design Thinking	38
Tabela 7 – Papéis no Uso do Design Thinking no Desenvolvimento de Software	39
Tabela 8 – Principais práticas utilizadas no Design Thinking	40
Tabela 9 – Principais Modelos associados ao Design Thinking	41
Tabela 10 – Principais ferramentas associados ao Design Thinking	42
Tabela 11 – Percentual de ferramentas em artigos selecionados	43
Tabela 12 – Fases e Aperfeiçoamento do Processo de Design Thinking	54
Tabela 13 – Nível de Conhecimento em Design Thinking no Desenvolvimento de Software	54
Tabela 14 – Desafios associados a práticas - Survey	55
Tabela 15 – Evidências do Design Thinking nos Trabalhos Relacionados	75
Tabela 16 – Evidências do Uso de Design Thinking nos Trabalhos Relacionados	76

LISTA DE ABREVIATURAS E SIGLAS

DT	Design Thinking;
XP	Extreme Programing;
SLR	Systematic Literature Review;
SDLC	Software Development Lifecycle;
TI	Tecnologia da Informação;

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Problematização	15
1.2	Objetivos da Pesquisa	15
1.2.1	Objetivo Geral	15
1.2.2	Objetivos específicos	16
1.3	Justificativa	16
1.4	Contribuições da Pesquisa	16
1.5	Organização do Trabalho	17
2	REFERENCIAL TEÓRICO	18
2.1	Design Thinking	18
2.1.1	Fases do Design Thinking	20
2.1.2	Engenharia de Software Experimental	22
2.1.3	Mapeamento Sistemático	23
2.1.4	Survey	23
3	MÉTODOS DE PESQUISA UTILIZADOS	25
3.1	Mapeamento Sistemático	26
3.1.1	Questões de pesquisa - Mapeamento Sistemático	27
3.1.2	Termos chaves da pesquisa	27
3.1.3	Strings de busca	28
3.1.4	Fontes de busca	28
3.1.5	Critérios de inclusão e exclusão	28
3.1.6	Seleção dos estudos primários	28
3.1.7	Extração da informação	29
3.1.8	Avaliação da qualidade do estudo	30
3.1.9	Tipos de estudo	30
3.1.10	Critérios de avaliação	31
3.2	Survey	31
3.2.1	Estruturação Survey	31
3.2.2	Hipótese	32
3.2.3	Questões de pesquisa - Survey	32
3.2.4	O Formulário	32
3.2.5	Benefícios	34
3.2.6	Desfecho Primário	34
3.2.7	Desfecho Secundário	34

3.2.8	Cronograma de execução	34
4	RESULTADOS DOS ESTUDOS EXPERIMENTAIS	36
4.1	Resultados do Mapeamento Sistemático	36
4.1.1	Desafios da utilização de DT no desenvolvimento de Software	37
4.1.2	Papéis exercidos no Design Thinking	38
4.1.3	Práticas relacionadas ao Design Thinking para produção de software	39
4.1.4	Modelos de DT e processos de desenvolvimento de software	40
4.1.5	Ferramentas para integrar o desenvolvimento de software com Design Thinking	42
4.1.6	Avaliação de qualidade e Alinhamento dos Estudos Primários	43
4.2	Resultados do Survey	45
4.2.1	Dados Gerais - Survey	45
4.2.2	Resposta a Questão de pesquisa Nº1 - Survey	53
4.2.3	Resposta a Questão de pesquisa Nº2 - Survey	54
4.2.4	Resposta a Questão de pesquisa Nº3 - Survey	55
4.2.5	Resposta a Questão de pesquisa Nº4 - Survey	56
4.2.6	Resposta a Questão de pesquisa Nº5 - Survey	57
5	ANÁLISE E DISCUSSÃO DOS PADRÕES DE EVIDÊNCIAS IDENTIFICADOS	59
5.1	Análise Geral dos Resultados	59
5.2	Análise dos Resultados dos Estudos Experimentais	62
5.3	Padrões de Evidências	66
5.4	Limitações Gerais desta pesquisa	71
6	TRABALHOS RELACIONADOS	72
7	CONSIDERAÇÕES FINAIS	77
8	REFERÊNCIAS BIBLIOGRÁFICAS	80
	APÊNDICES	86
	APÊNDICE A – FORMULÁRIO PARA EXTRAÇÃO E AVALIAÇÃO DA QUALIDADE DOS TRABALHOS	87
	APÊNDICE B – QUESTÃO DE PESQUISA Nº 1 / DESAFIOS	88
	APÊNDICE C – QUESTÃO DE PESQUISA Nº 2 / PAPÉIS ASSOCIADOS AO DESIGN THINKING	91

<i>APÊNDICE D – QUESTÃO DE PESQUISA Nº 3 / PRÁTICAS</i>	<i>92</i>
<i>APÊNDICE E – QUESTÃO DE PESQUISA Nº 4 / PRINCIPAIS MODELOS .</i>	<i>94</i>
<i>APÊNDICE F – QUESTÃO DE PESQUISA Nº 5 / FERRAMENTAS</i>	<i>96</i>

1 INTRODUÇÃO

Existem várias empresas de software que desenvolvem diversos produtos. Apesar de existirem várias empresas de desenvolvimento de software no mundo, muitos softwares ainda falham em atender aos requisitos e, eventualmente, os clientes acabam não os utilizando. Segundo IBRAIGHEETH (2020) a previsão de falha de projeto de software precocemente pode ajudar a tomar medidas de aprimoramento que podem conduzir o resultado do projeto do fracasso ao sucesso. Uma série de riscos pode afetar o projeto de software durante o processo de desenvolvimento e pode levar ao fracasso do projeto. Isso geralmente ocorre porque as equipes de software não conseguem entender completamente os requisitos do usuário. O Design Thinking (DT) é uma ferramenta que facilita a compreensão das necessidades do usuário, e tem se tornado uma boa alternativa para aqueles que buscam produtividade e satisfação do usuário com o produto (BUCHAN, 2017). Nos últimos anos, diversos estudos vêm sendo desenvolvidos para avaliar a efetividade da implantação do DT no desenvolvimento de software e como a implantação destas abordagens trazem melhorias na qualidade do software (PARIZI, 2022). A sua efetividade está atrelada ao envolvimento do usuário em várias fases do processo de desenvolvimento do produto, podendo assim, priorizar, testar e criar recursos mediante as necessidades informadas pelo usuário.

Nas últimas décadas, a participação do usuário no processo de desenvolvimento de software, foi considerada uma das perspectivas mais importantes para a obtenção do sucesso na estruturação de projetos (PEREIRA, 2021). Empresas de software vêm adotando a abordagem centrada no usuário em seus projetos com o intuito de gerar cada vez mais produtos inovadores, com valor agregado ao cliente (PERAIRE, 2019). A integração do DT na produção de software, faz com que o usuário tenha sempre pequenas entregas do produto, isto lhe possibilita avaliar os resultados de maneira constante, sem que tenha que aguardar a finalização de todo o projeto para usufruir e avaliar o produto. A estrutura do Design Thinking pode ser usada dentro do desenvolvimento de software para obter empatia do cliente e superar esse problema (MIRZA, 2020).

Este trabalho tem como foco promover evidências sobre os padrões de utilização do Design Thinking no desenvolvimento de software. Com base nisso, busca-se fornecer subsídios conceituais, juntamente com práticas da engenharia de software, para a adoção do Design Thinking em ambientes de desenvolvimento de software. Lidar com o desenvolvimento de projetos de software exige que as partes interessadas estejam bem alinhadas, além de ter muito bem definido qual o público alvo que deseja-se atingir com o projeto, quais os caminhos deverão ser percorridos e quais estratégias deverão ser utilizadas para êxito do projeto. Porém, executar tudo isto exige muita paciência e organização (SUMAN, 2021). A disposição dos assuntos nessa dissertação foi dividida em quatro partes e distribuída por 8

capítulos da seguinte forma: 1 - Introdução, 2 - Referencial Teórico, 3 - Métodos de Pesquisa Utilizados, 4 - Resultados dos Estudos Experimentais, 5 - Análise e Discussão dos Padrões de Evidências Identificados, 6 - Trabalhos Relacionados, 7 - Considerações Finais, e por fim, 8 - Referências Bibliográficas.

1.1 PROBLEMATIZAÇÃO

Uma das pesquisas realizadas pelo autor IBRAIGHEETH (2020), mostrou que uma média de 65% dos projetos em um contexto global de software falharam nas últimas décadas. O estudo também aponta que os projetos que falharam tem um ou mais das seguintes características, qualidade desejada não alcançada e não atender aos requisitos do usuário, levando ao abandono total do produto, mesmo que já tenha sido lançado. Conforme evidências contidas no estudo de OSAMA et al. (2019), onde o autor aborda sobre o aumento da demanda em agrupar as informações em ambientes digitais (Sistemas), isto promoveu que empresas refletissem sobre como entender as necessidades de seus clientes, e explorar a necessidade de desenvolver produtos inovadores para competir com o mercado, visto que os clientes estão cada vez mais exigentes, e procuram soluções rápidas e eficazes para atender às suas necessidades digitais.

O estudo realizado por DITTRICH et al. (2018), apontou que o mercado econômico global tem requisitado abordagens User Center Design (UCD) dentro do desenvolvimento de software, pois enfrentam desafios culturais, baixos recursos financeiros para a administração de projetos, problemas estruturais, e uma diversificidade literal de seus clientes. Isto faz com que as empresas busquem desenvolver produtos cada vez mais usuais e homogêneos, trazendo uma melhor interação do usuário no processo de construção dos produtos. O benefício dessa avaliação junto aos praticantes envolvidos no processo de desenvolvimento apresentou melhorias na redução de retrabalhos, usuários satisfeitos, melhor colaboração com as partes interessadas e uma compreensão próxima dos usuários e suas necessidades.

1.2 OBJETIVOS DA PESQUISA

1.2.1 Objetivo Geral

O presente trabalho tem como objetivo identificar, sintetizar e analisar padrões de evidências do emprego de abordagens de Design Thinking no desenvolvimento de software, através de um mapeamento sistemático e a execução de um survey, buscando auxiliar profissionais e pesquisadores a compreender com maior profundidade os desafios e implementar soluções mais eficazes para aprimorar suas atividades no desenvolvimento de software, utilizando abordagens de design thinking.

1.2.2 Objetivos específicos

Para alcançar o objetivo principal deste trabalho, alguns objetivos específicos foram executados, sendo eles:

- Planejamento e Execução de um mapeamento sistemático da literatura;
- Planejamento e Execução de um survey com profissionais que atuam com desenvolvimento de software;
- Avaliação de Qualidade, Extração de Informação e Análise Geral das evidências;
- Apresentar e Analisar os padrões de evidências identificados nas perspectivas da literatura e da indústria de software;
- Compreender como o Design Thinking afeta o processo de desenvolvimento de software;
- Prover evidências que auxiliam em como utilizar recursos de DT de forma satisfatória na indústria de software;

1.3 JUSTIFICATIVA

A necessidade de inovação e interação do usuário abrange desde o cenário corporativo até os ambientes educacionais, e as diversas faixas etárias. GURAN et al. (2020) relataram que as crianças que nascem no século atual (século XXI), são chamadas de nativos digitais, porém as suas habilidades digitais estão abaixo das expectativas da futura força de trabalho, que será tremendamente digital. Criar soluções que promovam a interação e auxiliem as pessoas no desenvolvimento de habilidades digitais, tem se tornado uma necessidade cada vez mais eminente no mercado tecnológico. Pesquisas sobre Design Thinking indicam que ele pode impactar positivamente as atividades de desenvolvimento de software, de forma que a sua adoção pode promover uma compreensão aprofundada do produto, dos usuários e de suas necessidades. Alguns pesquisadores recomendam a aplicação contínua do Design Thinking em todas as fases do desenvolvimento de software. Entretanto, ainda existe uma lacuna na compreensão de como essa abordagem é adotada nos processos de software, assim como sobre os recursos necessários para sua implementação (DOBRIGKEIT, 2021).

1.4 CONTRIBUIÇÕES DA PESQUISA

Durante o desenvolvimento desta pesquisa, alguns artigos foram desenvolvidos e outros estão em andamento, como pode ser visto na Tabela 1.

Tabela 1 – Artigos Derivados do Mestrado

Artigo	Revista / Conferência	Qualis	Estágio	Ano
Design Thinking Evidence in Software Development: Results From a Systematic Mapping	IEEE International Conference WETICE	A3	Publicado	2023
Design Thinking in Software Development: A Practical Approach in Technology Labs	IEEE International Conference SMC	A2	Publicado	2024
Practical Applications of Design Thinking: Experiences from Software Development Teams	IEEE International Symposium on Multimedia	A3	Submetido	2024
Design Thinking in Software Development: A Comprehensive Overview of Research	SN Computer Science	A2	Convidado	2024
A evidence roadmap about Design Thinking in Software Development	A definir	-	A submeter	2024

1.5 ORGANIZAÇÃO DO TRABALHO

A organização deste trabalho tem como objetivo apresentar os dados de estudos experimentais realizados para a compreensão de padrões de Design Thinking no desenvolvimento de software. No capítulo seguinte, serão apresentados os referenciais teóricos utilizados para a execução desses experimentos, cujos conceitos foram fundamentais para garantir que as diretrizes seguidas fossem seguras, a fim de gerar publicações consistentes com o estado da arte. No Capítulo 3, serão descritas as etapas de execução desta dissertação de mestrado, abordando as estratégias utilizadas para a realização do Mapeamento Sistemático e do Survey. Durante o mapeamento sistemático, foram seguidas dez etapas metodológicas, e, para o survey, foi estruturado um protocolo de pesquisa com base nas diretrizes apresentadas no estudo de WAGNER et al. (2020).

No Capítulo 4, serão apresentados os resultados obtidos com a realização dos estudos experimentais. No Capítulo 5, será feita uma análise geral das informações coletadas durante a execução dos estudos. O Capítulo 6 discutirá quatro estudos classificados como trabalhos relacionados, que tratam da aplicação do Design Thinking no desenvolvimento de software. Por fim, no Capítulo 7, serão apresentadas as considerações finais desta dissertação, juntamente com propostas para a continuidade desta pesquisa em estudos futuros.

2 REFERENCIAL TEÓRICO

Nesta seção, são apresentados os conceitos relevantes ao tema de pesquisa, proporcionando uma melhor contextualização. A seguir, serão abordados os conceitos de Design Thinking, Engenharia de Software Experimental, Mapeamento Sistemático e Survey, os quais foram fundamentais para a identificação de padrões no uso do Design Thinking nos processos de desenvolvimento de software. O Processo de Desenvolvimento de Software, ou Software Development Lifecycle (SDLC), é um processo que organiza as atividades de desenvolvimento em fases distintas, facilitando o planejamento e a gestão (DHANDAPANI, 2016). O conceito básico do modelo SDLC abrange atividades principais, como Requisitos, Projeto, Implementação, Teste, Depuração, Implantação e Manutenção. A maneira como essas atividades são executadas, de forma sequencial ou em paralelo, é o que torna cada abordagem única.

No processo de desenvolvimento de software, diversos modelos podem ser adotados no ambiente corporativo, e as organizações podem escolher o mais adequado com base em fatores como tempo e recursos disponíveis. Essa escolha, geralmente, é uma decisão conjunta entre os gestores e a equipe de desenvolvimento, considerando aspectos gerenciais, comerciais, técnicos e específicos de cada projeto. O desenvolvimento de software é orientado pelos escopos definidos pelos gerentes de projeto, sendo essa a fase em que os requisitos são transformados em soluções concretas no software em desenvolvimento (PETERS, 2018). É essencial que o processo de desenvolvimento de software siga as melhores práticas de engenharia de software, assim como outros requisitos centrados no usuário que foram incorporados ao design, visando construir produtos eficientes e alinhados às necessidades dos usuários.

2.1 DESIGN THINKING

O Design Thinking (DT) é um tema que vem despertando o interesse empírico sobre a sua aplicação e benefícios quando adotado dentro do processo de desenvolvimento de software (MAHÉ, 2019). As empresas de software têm utilizado o Design Thinking como uma abordagem de design centrado no usuário (User Center Design - UCD), colocando o usuário no centro do processo de desenvolvimento de software (PARIZI, 2022). Em linhas gerais, DT é uma abordagem UCD que envolve o usuário no desenvolvimento de soluções de software inovadoras (Pereira and Russo, 2018). Design Thinking ou pensamento de design, é a abstração do modelo mental adotado pelos designers para gerar ideias e resolver problemas (BROWN, 2008). O DT enfatiza a busca pela criatividade e sua implantação auxilia equipes a lidar com os mais diversos problemas, sem estruturação ou até mesmo aqueles que não

possuem definições claras e concisas (SENF, 2019). O DT foi declarado pela primeira vez em 1987 por ROWE (professor de arquitetura e design urbano na Universidade de Harvard) em seu livro *Design Thinking*. O autor defendeu que os designers têm uma forma única de abordar o seu trabalho, um processo único de resolução de problemas (KIMBELL, 2011; LUPTON, 2011; TSCHIMMEL, 2012).

Os autores BRENNER et al. (2016) sugerem que o DT como um recurso organizacional pode ser descrita em três níveis diferentes: como uma mentalidade, como um processo ou como uma caixa de ferramentas. Essa caracterização tem sido amplamente aceita na literatura (KUULA et al., 2020; LEVY e HULI, 2019; HEHN et al., 2019; MAHE et al., 2019). O desenvolvimento de uma mentalidade de Design Thinking (DT) é um conceito fundamental tanto para sua prática quanto para sua pesquisa. Certas mentalidades são particularmente valiosas ao aprender ou aplicar o DT. Por exemplo, adotar uma mentalidade empática é essencial para compreender as necessidades dos usuários e para promover a colaboração em equipes diversificadas. O DT como mentalidade reconhece que a inovação é um processo humano voltado para atender às necessidades humanas. Essa abordagem integra pensamento divergente e convergente, adota a filosofia de aprender com falhas rápidas e precoces, incentiva a criação de protótipos para experimentação e enfatiza a importância de realizar testes antecipados com os clientes. (DOBRIGKEIT, 2019). O Design Thinking, enquanto processo, é organizado como uma série de espaços de trabalho iterativos que combinam pensamento divergente e convergente. DT como uma caixa de ferramentas se refere ao uso de métodos e técnicas de design de engenharia, computação e psicologia para resolver um problema (KUULA et al., 2020). A literatura apresenta diversos processos de DT, frequentemente denominados modelos de Design Thinking (PARIZI et al., 2022). Cada modelo caracteriza o DT como um conjunto de etapas ou espaços dedicados à compreensão do problema e à criação de soluções inovadoras, a Tabela 2 destaca as variações desses processos com base nas definições apresentadas por diferentes autores.

Tabela 2 – Diferentes modelo de Design Thinking

Autor	Modelos de Design Thinking
Brown (2008)	Inspiração / Ideação / Implementação
D.School Stanford (2010)	Empatia / Definir / Idear / Prototipagem / Teste
Brenner et al. (2016)	Necessidade e Síntese / Ideação / Protótipo / Teste / (Re)definir problema

Nos últimos anos, autores têm discutido sobre a crescente demanda por sistemas de informação, e como isso levou as empresas a repensar mais as necessidades de seus clientes e a necessidade em desenvolver produtos inovadores para competir no mercado. Para alcançar este objetivo, o Design thinking se torna uma metodologia que auxilia na construção de software, pois é centrada no ser humano e que pode levar à criatividade e inovação dos processos para aqueles que a adotarem (SOHAIB, 2019; PRESTES, 2020). Com base nos artigos

classificados como objetos deste estudo, foi observado que houve um aumento significativo no desenvolvimento de estudos de DT associado ao processo de desenvolvimento de software. Aspectos como a necessidade de interação com o usuário, empreendedorismo, criação de produtos inovadores (FRANZISKA, 2019), são fatores que impactam na busca de soluções como o Design Thinking para atingir esses objetivos.

2.1.1 Fases do Design Thinking

Os autores FAWCETT et al. (2013) mencionam que o DT aplica muitas técnicas criativas e inovadoras para desenvolver produtos modernos. Uma das vantagens da aplicação do DT é que seus conceitos podem ser aplicados em qualquer cenário e fase (FARRAR, 2020).

A implementação do DT pode ocorrer de maneira interativa ou não linear (FRANZISKA, 2019). A utilização do DT auxilia as equipes a explorar alternativas e realizar novas descobertas sobre determinado tema, permitindo que sejam motivadas a refletir sobre premissas anteriormente adotadas antes da implantação do DT. A Figura 1, aborda a representação dessas atividades, considerando um ciclo segmentado em fases.

Figura 1 – Fases do Design Thinking



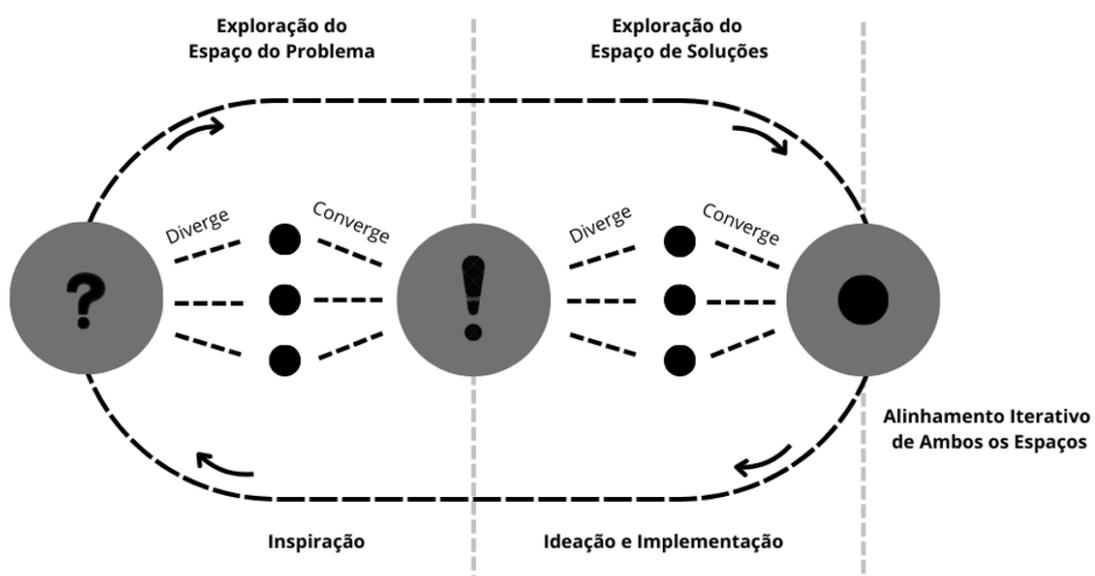
Fonte: Adaptado de Falk (2020)

Segundo as diretrizes de BROWN (2008) o processo de DT é considerado um sistema composto por três espaços sobrepostos, e não uma sequência de etapas ordenadas: inspiração, ideação e implementação. A inspiração refere-se à fase em que se identifica o problema ou oportunidade que motiva a busca por soluções; a ideação representa o processo de geração, desenvolvimento e teste de ideias; e, por fim, a implementação visa levar o projeto ao público (JIMÉNEZ, 2021). Esses espaços não seguem uma ordem fixa, permitindo que a equipe volte a qualquer um deles — inspiração, ideação ou implementação — repetidamente, à medida que as ideias são refinadas (PETERSEN et al., 2017).

Na fase de inspiração, busca-se compreender as necessidades e desafios relacionados ao problema que se deseja resolver. A coleta de informações pode ser realizada por meio de entrevistas individuais ou em grupo, de forma presencial ou digital, além de estudos na literatura sobre o tema em questão. Na fase de ideação, os dados coletados são analisados e transformados em *insights* para soluções inovadoras. A análise pode ser conduzida por meio da condensação sistemática dos dados principais, seguida da segmentação em temas preliminares. A criação de protótipos também auxilia na abstração de soluções que atendam às necessidades dos usuários.

Na fase de implementação, as ações são executadas, incluindo a realização de testes, atividades essenciais focadas nas necessidades dos usuários, validações constantes e melhorias no que já foi desenvolvido. A Figura 2 apresenta a relação entre as fases e os espaços onde são exploradas as soluções e os problemas a serem tratados com a adoção do Design Thinking.

Figura 2 – Exploração do Espaço de Problemas e Soluções no Design Thinking



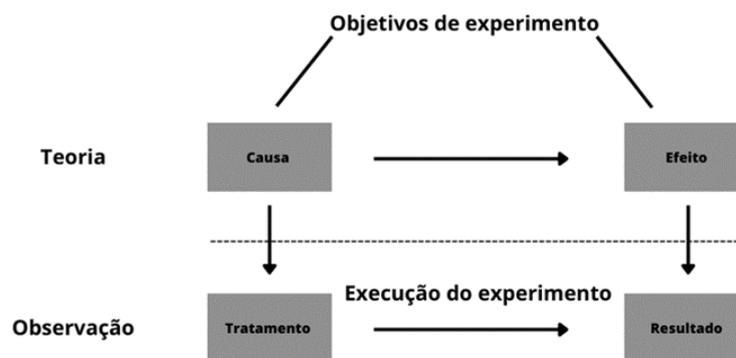
Fonte: Adaptado de Gurusamy (2016)

2.1.2 Engenharia de Software Experimental

Experimentação é o centro do processo científico. Somente experimentos verificam as teorias. Somente experimentos podem explorar os fatores críticos e dar luz ao fenômeno novo para que as teorias possam ser formuladas e corrigidas. (TRAVASSOS, 2002). O método de experimentação consiste em avaliar novas invenções, comparando com as com inovações já existentes, de modo que possuam alguma característica semelhante.

O método de engenharia de software experimental observa as soluções existentes, sugere as soluções mais adequadas, desenvolve, mede e analisa, e repete até que nenhuma melhoria adicional seja possível. Isto é uma abordagem orientada à melhoria evolutiva que assume a existência de algum modelo do processo ou produto de software e modifica este modelo com propósito de melhorar os objetos do estudo. A Figura 3 descreve o fluxo do processo de experimentação.

Figura 3 – Processo de experimentação



Fonte: Adaptado de Wohlin (2012)

Sendo assim, os elementos principais do experimento são as variáveis, os objetos, os participantes, o contexto do experimento, hipóteses e o tipo do projeto do experimento, que pode ser qualitativa, quantitativa, ou ambas (TRAVASSOS, 2008).

Conforme apresentado na Figura 3, existem dois tipos de variáveis do experimento, sendo elas as variáveis dependentes e as variáveis independentes. As variáveis dependentes, referem-se à saída do processo de experimentação, ou seja, podem representar o efeito que é causado pelos fatores do experimento, sendo comumente apresentada como o resultado da experimentação. Enquanto as variáveis independentes, referem-se à entrada do processo de experimentação, comumente chamadas de “fatores”, pois apresentam a causa que afeta o resultado do processo.

2.1.3 Mapeamento Sistemático

Uma revisão sistemática (Systematic Literature Review - SLR) é uma forma secundária de estudo que utiliza uma metodologia bem definida para identificar, analisar e interpretar toda a evidência disponível a respeito de uma questão de pesquisa específica, uma área ou um fenômeno de interesse do pesquisador. Existem dois tipos de revisões que compreendem a SLR: mapeamento sistemático (systematic mapping) e a revisão terciária (tertiary review) (KITCHENHAM, 2007).

De acordo com os autores citados acima, um mapeamento sistemático permite que a evidência seja analisada de uma forma macro, permitindo assim a identificação de clusters de evidências o que pode direcionar o foco de futuras revisões sistemáticas e identificar áreas de interesse para a condução de novos estudos primários.

PETERSEN et al. (2008) destacam que o principal objetivo de um mapeamento sistemático é fornecer uma visão geral da área de pesquisa, identificar a quantidade, os tipos de pesquisa na área, além dos resultados disponíveis nela. As principais diferenças entre mapeamentos sistemáticos e revisões sistemáticas, de acordo com KITCHENHAM (2007), são:

- Mapeamentos sistemáticos geralmente possuem questões de pesquisa mais amplas, e também podem ter várias questões de pesquisa;
- Os termos de pesquisa (search terms) em mapeamentos são menos focados que em SLR e provavelmente retornarão um grande número de documentos. Dada a natureza do mapeamento, isto não é um problema já que o objetivo aqui é uma cobertura mais ampla;
- O processo de extração dos dados no estudo de mapeamento é muito mais amplo do que em uma SLR. O objetivo nesta etapa será classificar os estudos selecionados com detalhes suficientes para responder às questões de pesquisa;
- O estágio de análise no mapeamento serve para resumir os dados que vão responder às questões de pesquisa. É improvável que sejam feitas análises detalhadas com técnicas complexas. Representação gráfica pode ser uma boa forma de expor os dados;
- Por fim, a divulgação dos resultados de um mapeamento pode ser mais limitada do que a divulgação de um SLR.

2.1.4 Survey

A pesquisa foi concebida com o objetivo de reunir evidências sobre a adoção do Design Thinking em cenários reais de desenvolvimento de software, por meio da avaliação de profissionais da área de TI. Os instrumentos de pesquisa baseados em surveys são métodos sistemáticos utilizados para coletar informações em estudos quantitativos (por meio de questionários) e qualitativos (por meio de entrevistas). Esses instrumentos permitem

avaliar percepções sobre um determinado tema dentro de uma amostra populacional, possibilitando a realização de inferências sobre a população como um todo (WOHLIN et al., 2012). Para garantir uma abordagem sólida e consistente, utilizou-se as diretrizes de pesquisa estabelecidas no guia desenvolvido por WAGNER et al. (2020) e conduzimos cuidadosamente as seguintes etapas:

Etapa 1. Design da Pesquisa. Realizou-se um mapeamento sistemático sobre a utilização do Design Thinking no desenvolvimento de software (JESUS, 2023), a fim de fornecer as bases teóricas para a formulação das perguntas e opções de resposta. A partir disso, a pesquisa inicial foi elaborada com base nas evidências preliminares do estudo mencionado, visando obter respostas específicas para atender às questões de pesquisa do *survey*.

Etapa 2. Revisão do Design da Pesquisa. A pesquisa foi revisada e ajustada com base em discussões *online* e no *feedback* registrado por pesquisadores de Engenharia de Software da Universidade Federal de Alagoas (UFAL) e da Universidade Federal do Agreste de Pernambuco (UFAPE). Em seguida, a pesquisa também foi revisada pelos autores do estudo.

Etapa 3. Avaliação inicial da estruturação da pesquisa. Esta avaliação consistiu em uma revisão preliminar realizada por entrevistados selecionados aleatoriamente. Um grupo de 10 profissionais que atuam no desenvolvimento de software foi convidado a fornecer *feedback* sobre a clareza das perguntas e a registrar o tempo de resposta. Essa fase resultou em pequenos ajustes relacionados à clareza das perguntas do formulário da pesquisa. As respostas foram descartadas antes do lançamento oficial da pesquisa.

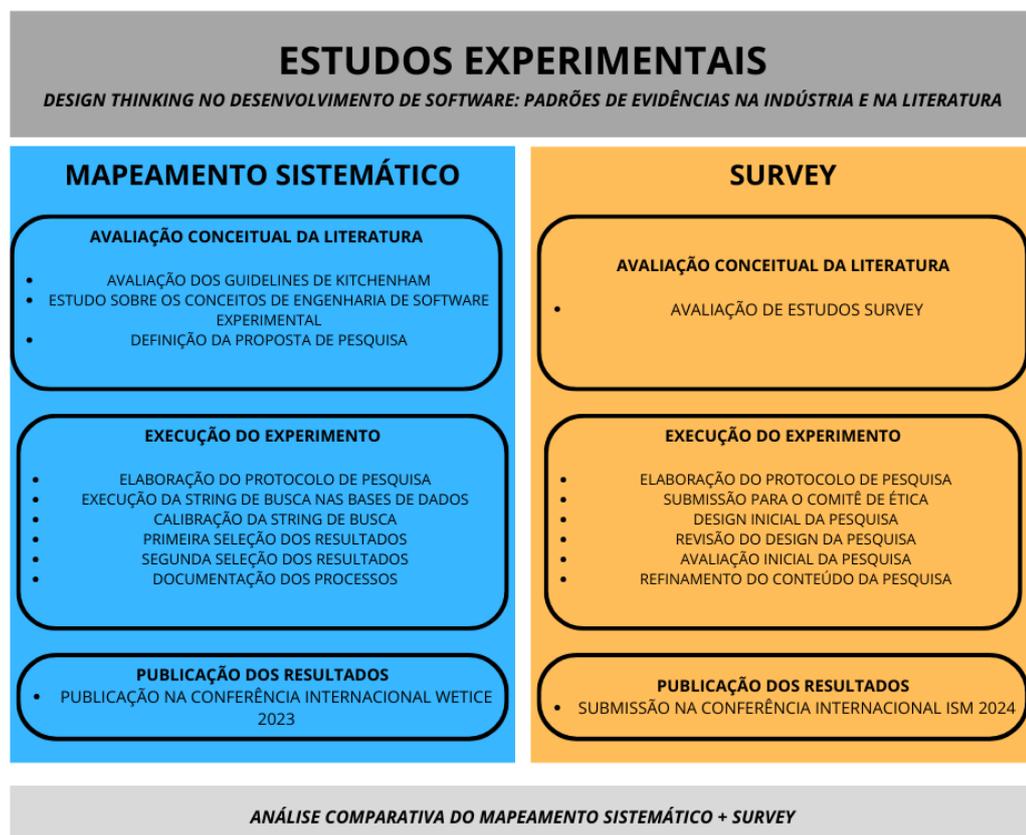
Etapa 4. Refinamento do Conteúdo da Pesquisa. Esta fase envolveu a avaliação por especialistas no tema da população-alvo. Foram selecionados quatro profissionais seniores em desenvolvimento de sistemas, que na sequência foram solicitados a responder a pesquisa e transmitirem seus respectivos *feedbacks*. Os participantes não tiveram dificuldades em responder à pesquisa, que levou, em média, cinco minutos. Após essa etapa, a pesquisa foi considerada pronta para divulgação.

3 MÉTODOS DE PESQUISA UTILIZADOS

Neste capítulo estão descritas as características dos tipos de pesquisa adotadas neste trabalho. Inicialmente, os conceitos iniciais dizem respeito a um estudo de mapeamento sistemático, e na sequência fazem referência a execução de um protocolo *survey*. Por meio do estudo teórico de mapeamento sistemático foi realizada uma análise do que se havia feito na literatura a respeito da adoção do Design Thinking em processos de software. Com base nestas evidências, um estudo aprofundado com o intuito de avaliar as evidências dos achados do mapeamento sistemático em cenários reais de desenvolvimento de software foi executado por meio de um formulário de pesquisa *online*.

A Figura 4 apresenta as macro etapas realizadas durante o desempenho desta pesquisa de mestrado englobando as fases de planejamento e execução do mapeamento sistemático e do *survey*. Para que fosse possível atender os objetivos descritos neste trabalho, as etapas descritas foram executadas de forma sequencial, com o intuito de realizar análises após cada etapa finalizada. A publicação dos resultados dos experimentos foram efetuadas em conferências internacionais conforme apresentado na Tabela 1.

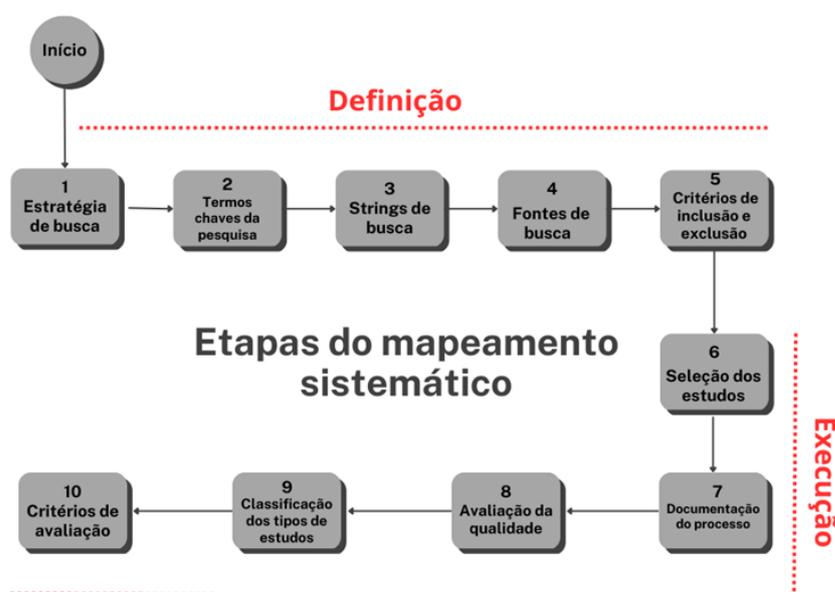
Figura 4 – Fluxo de etapas da pesquisa



3.1 MAPEAMENTO SISTEMÁTICO

O mapeamento sistemático foi realizado seguindo as diretrizes estabelecidas para a realização de revisões sistemáticas da literatura sugeridas por Kitchenham (2007), que são um meio comprovado para chegar a um censo completo das pesquisas existentes dentro de um domínio. Nesta seção são apresentados as etapas de execução do mapeamento sistemático. Quanto ao método adotado para a condução do estudo, foram utilizados os conceitos de uma revisão sistemática da literatura, a partir dos objetivos definidos no escopo definido na seção de objetivos, que visa sintetizar em um modelo de evidências a respeito de como o Design Thinking pode afetar o processo de desenvolvimento de software. Dez etapas metodológicas foram empregadas. A primeira etapa teve como objetivo definir as estratégias de busca do mapeamento sistemático. A segunda etapa consistiu em definir os termos chaves da pesquisa. A terceira envolveu a estruturação e criação da *string* de busca da pesquisa. A quarta etapa compreendeu em definir as bases de dados para a execução do mapeamento. A quinta envolveu a definição dos critérios de inclusão e exclusão dos resultados. A sexta etapa teve como ação realizar a seleção dos artigos primários encontrados. A sétima, consistiu em documentar o processo de busca, sendo as próximas etapas definidas em avaliar a qualidade do estudo, avaliar os tipos de estudos encontrados e execução dos critérios de avaliação, respectivamente. Essas etapas estão demonstradas na Figura 5 e serão descritas nas próximas seções.

Figura 5 – Etapas do mapeamento sistemático



Fonte: Adaptado de KITCHENHAM (2010)

3.1.1 Questões de pesquisa - Mapeamento Sistemático

A realização deste estudo buscou responder à seguinte questão geral de pesquisa: (QG): Como o Design Thinking afeta o processo de desenvolvimento de software? Neste sentido, também foram formuladas questões específicas de pesquisa, considerando quatro questões de pesquisa, buscando evidências que mapeiem e componham as atividades de desenvolvimento de software e DT.

Q1. Quais são os desafios da utilização do Design Thinking no processo de desenvolvimento de Software?

Q2. Quais são os papéis exercidos no Design Thinking associados durante o desenvolvimento de software?

Q3. Quais são as boas práticas utilizadas no Design Thinking em desenvolvimento de software?

Q4. Quais são os modelos de processos de Design Thinking encontrados no desenvolvimento de software?

Q5. Quais são as ferramentas de DT utilizadas no desenvolvimento de software?

3.1.2 Termos chaves da pesquisa

Após a definição do projeto, tendo como base as diretrizes de KITCHENHAM (2007), como população, intervenção, questões de pesquisa e resultados, faz-se necessário a inclusão de palavras-chave, com o intuito de facilitar a identificação da ideia central do documento.

Após a determinação dos termos principais de pesquisa, seleciona-se um conjunto de sinônimos e posteriormente realiza-se a tradução para o inglês por ser o idioma usual de revistas e artigos científicos. Além disso, palavras que possuem flexão de número (singular e plural), estão demarcadas com o símbolo asterisco (*). Os termos e sinônimos identificados são apresentados abaixo:

- **Desenvolvimento de Software:** Software development; Tailoring software process, development stages;

- **Emprego de Design Thinking no desenvolvimento de projetos:** use of design development of software projects, User-centered design, User centric design; Design Thinking, Design session;

- **Desafios:** Challenge*, Difficult*, problem;

- **Papéis:** functions, assignments, roles;

- **Boas Práticas:** Good Practice, best practice, practice;

- **Modelos:** Models, Prototypes;

- **Ferramentas:** Tool*;

3.1.3 Strings de busca

Segundo KITCHENHAM (2007), as *strings* de pesquisa são derivadas apropriadamente das questões de pesquisa. Portanto, as *strings* de busca foram geradas a partir da combinação dos operadores booleanos “OR” (ou) e “AND” (e), e as palavras chaves associadas às questões de pesquisa, englobando também os seus respectivos sinônimos. A *string* utilizada para responder às questões será listada a seguir: (Tabela 3).

Tabela 3 – String para as questões de pesquisa

Questão de pesquisa	String
Q1, Q2, Q3, Q4 e Q5	((("Design Thinking") OR ("user-centered design")) AND (("software development") OR ("software process"))) AND (("challenges") OR ("roles") OR ("models") OR ("practices") OR ("tools")))

3.1.4 Fontes de busca

Existem várias fontes de bancos de dados acadêmicos. Para definições destas fontes de dados foram levados em consideração alguns aspectos, dentre eles: Importância e relevância das fontes; Disponibilidade de consulta dados *online*; Compatibilidade de mecanismos de busca usando palavras-chave (keywords). Portanto, as fontes de pesquisa utilizadas para a busca dos estudos primários foram:

- IEEE Xplore Digital Library (<https://ieeexplore.ieee.org/>)
- ACM Digital Library (<https://portal.acm.org>)
- Scopus (<https://www.scopus.com>)

3.1.5 Critérios de inclusão e exclusão

Para garantir a seleção imparcial de artigos, foi utilizado um processo multiestágio que envolveu dois pesquisadores, onde foram documentadas as razões para inclusão/exclusão de cada artigo, conforme sugerido por KITCHENHAM (2007). A Tabela 4 apresentada abaixo, retrata os critérios de inclusão e exclusão adotados neste estudo.

3.1.6 Seleção dos estudos primários

O processo de seleção dos estudos é constituído por várias etapas, cujo a finalidade é avaliar se o resultado é condizente com o que se deseja encontrar, pontos estes que são

Tabela 4 – Critérios de Inclusão/Exclusão

Tipo	Descrição
Inclusão	Trabalhos relacionados ao DT no desenvolvimento de software
Inclusão	Artigos publicados entre 2000 e 2023
Exclusão	Trabalhos que não discutem DT no desenvolvimento de software
Exclusão	Documentos não disponíveis para download
Exclusão	Trabalhos não escritos em inglês
Exclusão	Artigos que não foram revisados por pares
Exclusão	Capítulos de livros
Exclusão	Artigos Incompletos

definidos pelas questões de pesquisa. Em seguida, definiu-se as estratégias utilizadas para a busca dos estudos primários, englobando os critérios de inclusão e exclusão. As etapas desta abordagem serão explanadas a seguir:

- Dois pesquisadores (Mestrando e Orientador) inicialmente realizam as buscas dos estudos, neste momento são levados em consideração o título dos estudos e as palavras-chaves dos artigos encontrados. Havendo compatibilidade com as questões de pesquisas, estes artigos são classificados para a próxima etapa de seleção, caso contrário são excluídos do processo;
- Após a etapa anterior foi obtida uma lista de estudos relevantes em relação às questões de pesquisas que foram definidas na composição do mapeamento sistemático. Nesta etapa, foi efetuado uma análise mais robusta dos artigos, avaliando os resumos e a seção de conclusão dos trabalhos.
- Após a análise dos resumos e conclusões, realizou-se a classificação dos artigos como aptos ou não aptos para composição dos resultados deste mapeamento sistemático.
- Na etapa final, foi efetuado a extração dos dados e avaliação da qualidade dos trabalhos realizados.

3.1.7 Extração da informação

Tendo como base as diretrizes de KITCHENHAM (2007), essa seção aborda as limitações e adaptações que ocorrem no processo de busca definido para essa revisão. Observou-se limitações quanto às fontes de busca, algumas ainda não contemplam as abordagens utilizadas e relatadas nesta revisão, e outras possuem limitações quanto ao número de operadores booleanos utilizados, tendo a necessidade de realizar adaptações nas *strings* de busca. Em complemento, a literatura científica está em constante evolução, o que pode exigir atualizações frequentes no mapeamento para incluir novos estudos publicados

após a execução inicial.

Outro desafio encontrado é que os resultados encontrados nem sempre incluem todos os dados relevantes. Tendo a necessidade de verificar com muita cautela várias etapas do artigo, por exemplo, a seção de métodos e resultados, tornando o processo de análise demorado e rigoroso.

3.1.8 Avaliação da qualidade do estudo

KITCHENHAM (2007:26-28) destaca a importância de avaliação de qualidade de trabalhos quantitativos e qualitativos, em que são apresentados critérios gerais e específicos para submeter e avaliar a relevância da pesquisa. KITCHENHAM (2007) argumenta ainda que a execução da avaliação de qualidade dos estudos incluídos é crítica por alguns motivos, que incluem: fornecer ainda mais detalhes para aplicar os critérios de exclusão/inclusão; investigar se as diferenças de qualidade explicam as diferenças dos resultados dos estudos; a avaliação de qualidade caracteriza-se como uma maneira de dar pesos na importância de estudos individuais quando os resultados são sintetizados; esta etapa pode guiar a interpretação dos resultados encontrados e determinar a força das inferências; e pode guiar a recomendação de trabalhos futuros.

No apêndice A, estão contidas as perguntas utilizadas para a avaliação da qualidade dos artigos selecionados. Foram utilizadas 8 perguntas com o intuito de avaliar se os artigos respondem a cada uma das questões de pesquisa individualmente. Para cada pergunta, as notas foram efetuadas com os seguintes critérios:

- 1.0 - Significa que o estudo cumpriu o critério COMPLETAMENTE.
- 0.5 - Significa que o estudo cumpriu PARCIALMENTE o critério
- 0.0 - Significa que o estudo NÃO cumpriu o critério.

3.1.9 Tipos de estudo

Uma Revisão Sistemática de Literatura é uma forma de identificar, avaliar e interpretar todas as pesquisas disponíveis que são relevantes para uma particular questão de pesquisa, ou área, ou fenômeno de interesse”. Segundo KITCHENHAM (2007), existem algumas características que diferenciam uma revisão sistemática de uma revisão da literatura de uma revisão convencional. As revisões sistemáticas possuem um escopo e um protocolo de execução específico, voltado à responder questões de pesquisas estabelecidas. Além de possuir critérios de inclusão e exclusão para avaliar cada estudo primário de relevância. Enquanto a convencional segue os processos mais abrangentes de análise.

KITCHENHAM (2007), relata que o objetivo dos estudos empíricos em engenharia de software é “Fornecer os meios pelos quais as melhores evidências atuais de pesquisa podem

ser integradas com a experiência prática e valores humanos no processo de tomada de decisão sobre o desenvolvimento e manutenção de software”. Neste contexto, a evidência é definida como uma síntese da melhor qualidade estudos científicos sobre um tópico específico ou questão de pesquisa. O principal método de síntese é uma revisão sistemática da literatura (SLR), que é o que foi proposto neste trabalho.

3.1.10 Critérios de avaliação

KITCHENHAM (2007) destaca a importância do uso dos critérios, que podem ser gerais referentes a escrita e apresentação da pesquisa (Exemplo: se os objetivos da pesquisa estão apresentados de forma coerente e clara; se há discussão dos resultados), e critérios mais específicos para a questão de pesquisa da revisão sistemática (Exemplo: se houve determinação clara das questões de pesquisa; explicitação e utilização dos critérios de inclusão ou exclusão). O objetivo de definir os critérios é identificar os estudos primários que fornecem evidências diretas a respeito das questões de pesquisa (KITCHENHAM, 2007). Mediante isto, foram avaliados esses aspectos absorvendo o conteúdo de cada artigo, visando distinguir se os mesmos estavam de acordo com as questões de pesquisa definidas.

3.2 SURVEY

Para confrontar os resultados a partir do Mapeamento Sistemático, foi realizado a execução de uma pesquisa com profissionais de Tecnologia da Informação, com o intuito de investigar sobre a aplicação do Design Thinking no desenvolvimento de software. A partir disso, buscou-se compreender quais são as evidências do DT nas fases do desenvolvimento de software. Esta seção descreve o protocolo utilizado para a execução de um *survey*, cujo objetivo principal é investigar os desafios da utilização do Design Thinking no processo de desenvolvimento de software por meio de profissionais da área de TI.

3.2.1 Estruturação Survey

Nesta seção são apresentados as etapas de execução do formulário de pesquisa. Quanto ao método adotado para a condução da pesquisa, foram elaborados a partir dos objetivos definidos no escopo definido na seção de objetivos, que visa identificar a aplicação do Design Thinking no processo de desenvolvimento de software. Foi utilizado o aplicativo de gerenciamento de pesquisa Google Forms para projetar e hospedar nossa pesquisa on-line, que esteve disponível para participação entre os meses de Maio a Agosto de 2024. De acordo com as diretrizes de ética, os entrevistados podem solicitar a remoção de suas respostas do estudo a qualquer momento, desde que encaminhe formalmente a solicitação de exclusão por e-mail. Para atingir o número máximo de respostas, os convites foram enviados para listas de e-mails associadas a fóruns de tecnologia e também grupos do Telegram.

No formulário, as perguntas foram estruturadas com algumas estratégias, buscando identificar a caracterização da empresa e do participante, desafios, práticas e metodologias do desenvolvimento de software, bem como desafios e ferramentas associadas ao tema.

3.2.2 Hipótese

A aplicação do Design Thinking no desenvolvimento de software pode resultar em produtos usuais, centrados no usuário e adaptáveis às necessidades em constante mudança dos clientes.

3.2.3 Questões de pesquisa - Survey

O objetivo primário desta pesquisa buscou responder a questão geral de pesquisa (QG): Como utilizar recursos de DT de forma satisfatória na indústria de software? Para tanto, as demais questões secundárias da pesquisa foram formuladas. A pesquisa parte para cinco questões de investigação mais específicas que possam responder essas perguntas, na busca por uma abordagem que apoie com práticas e ferramentas eficazes no processo de desenvolvimento de software.

Q1. Quais são as evidências de DT no cenário de desenvolvimento de software?

Q2. Quais os principais desafios encontrados na utilização de DT na produção de software?

Q3. Quais os possíveis recursos de DT costumam ser utilizados para otimizar a construção de software?

Q4. Quais são os tipos de projeto em que DT é utilizado?

Q5. Quais os papéis que costumam atuar mais com DT e software?

3.2.4 O Formulário

A indústria de software precisa de produtos inovadores que atendam cada vez mais as perspectivas dos usuários, com boa usabilidade e funcionalidades que atendam às suas reais demandas. Nesse sentido, o conceito de Design Thinking tem se consolidado como um poderoso recurso para desenvolvimento de software, trazendo vantagens competitivas e impactando diretamente na qualidade do produto. Para execução desta pesquisa, perguntas foram formuladas a partir dos objetivos definidos no escopo definido na seção de objetivos. O formulário utilizado para captura dos dados foi elaborado por meio da plataforma Google Forms, e pode ser acessado através do link <https://forms.gle/3EQ19uE1MfQvVqsB7>.

O formulário possui um total de 18 perguntas, sendo 17 obrigatórias e 1 opcional. O mesmo visa ser aplicado para pessoas que atuam profissionalmente no cenário de

desenvolvimento de software. Para caracterização da empresa e do participante as seguintes perguntas foram desenvolvidas:

- 1 - E-mail do participante (Opcional)
- 2 - Qual a localidade da empresa em que atua? (Cidade / Estado / País).
- 3 - Quantos funcionários sua empresa tem?
- 4 - Qual a sua função ou os papéis que desempenha na empresa?
- 5 - Há quanto tempo atua com desenvolvimento de software?
- 6 - Qual o seu nível de formação ?
- 7 - Quais papéis/funções costumam compor as equipes na sua empresa?
- 8 - Em uma escala de 1 a 5, qual o seu grau de conhecimento sobre Design Thinking?
1 corresponde a pouco conhecimento e 5 corresponde a muito conhecimento sobre Design Thinking na produção de software.
- 9 - Há quanto tempo atua com Design Thinking na produção de software?

Para caracterização do processo, práticas e metodologias do desenvolvimento de software as seguintes perguntas foram desenvolvidas:

10 - Em uma escala de 1 a 5, como você considera a adoção de DT na produção de software na sua empresa? Onde 1 corresponde a pouca utilização dos conceitos de DT e 5 corresponde a muita utilização dos conceitos de Design Thinking na produção de software.

11 - Em quais fases do desenvolvimento de software a empresa tem utilizado o Design Thinking?

12 - Em sua atuação com o Design Thinking, quais são as práticas mais utilizadas em seus projetos? Como a empresa busca por novas práticas, metodologias e processos para melhorar seu processo de Design Thinking?

13 - Combinadas com o Design Thinking, quais metodologias/modelos são utilizados na empresa em que atua?

14 - Como a empresa busca por novas práticas, metodologias e processos para melhorar seu processo de Design Thinking?

15 - Em seu contexto profissional, como metodologias/modelos associados ao Design Thinking atendem às necessidades do seu projeto? Onde 1 corresponde a um baixo grau de atendimento e 5 corresponde a um alto grau de atendimento.

Por fim, para a identificação dos desafios e ferramentas da adoção no Design Thinking no desenvolvimento de software as seguintes perguntas foram desenvolvidas:

16 - Quais são os principais desafios do time ao utilizar o Design Thinking?

17 - Quais ferramentas alinhadas ao DT são utilizadas em seus projetos para o desenvolvimento de softwares?

18 - Em sua atuação profissional na empresa com DT, quais são os tipos de projeto de software desenvolvidos?

A partir das respostas do formulário, uma análise de dados deverá ser executada, buscando sintetizar em um modelo de evidências a respeito de como o Design Thinking pode afetar o processo de desenvolvimento de software.

3.2.5 Benefícios

Os benefícios desta pesquisa contemplam mapear evidências de abordagens UCD (User Center Design) no desenvolvimento de software, buscando promover produtos que atendam as necessidades dos clientes, além de levantar possíveis caminhos para entender os requisitos necessários para resolução desta problemática, e como isso pode se tornar cada vez mais comum entre empresas do ramo de tecnologia da informação.

3.2.6 Desfecho Primário

De forma geral, este trabalho busca desenvolver um conjunto de evidências que enfatizam o uso de DT no processo de software, as perguntas buscam identificar e analisar quais desafios, funções, práticas, modelos e ferramentas do Design Thinking são usados no desenvolvimento de software. Este formulário foi desenvolvido por meio da plataforma do Google Forms, e tem como ênfase a aplicação em cenários reais de desenvolvimento de software através das respostas de profissionais que já atuam na área de forma profissional.

3.2.7 Desfecho Secundário

Os resultados identificados serão compilados e analisados, e a partir disto algumas discussões e análises mais profundas foram realizadas entre os pesquisadores responsáveis pela pesquisa (Mestrando e Orientador). Este corpo de evidências cobre uma literatura atual sobre DT e os resultados poderão ajudar profissionais e pesquisadores da área de desenvolvimento de software para entender melhor os desafios e implementar soluções mais eficazes para melhorar suas tarefas usando abordagens de design thinking.

3.2.8 Cronograma de execução

Além de promover a organização, o cronograma é uma ferramenta indispensável para gerenciar o tempo de um projeto. O cronograma apresenta a relação das atividades a serem realizadas, indicando-se a previsão de tempo necessário para a execução de cada etapa da pesquisa. As fases de execução das etapas desta pesquisa consistiram na elaboração

do formulário de pesquisa, desenvolvido no Google Forms, estruturação do protocolo de pesquisa a ser desempenhado, aplicação do *survey*, análise dos dados, escrita dos resultados finais e publicação no estado da arte.

4 RESULTADOS DOS ESTUDOS EXPERIMENTAIS

O propósito deste capítulo é apresentar os dados obtidos durante a execução dos estudos experimentais. Os dados expostos consistem em informações provenientes do Mapeamento Sistemático e do Survey. Inicialmente, são apresentadas as principais características do Mapeamento Sistemático (Seção 4.1) e as principais características do *Survey* (Seção 4.2). Em seguida, os resultados obtidos serão apresentados, discutidos e correlacionados, além de serem debatidos à luz da literatura.

4.1 RESULTADOS DO MAPEAMENTO SISTEMÁTICO

A seguir, serão apresentados as informações sobre os desafios da utilização do Design Thinking no processo de desenvolvimento de Software. A segunda seção apresenta os resultados referente aos papéis exercidos no Design Thinking durante o desenvolvimento de software. A Seção 4.1.1 fornece os resultados referente aos desafios encontrados na utilização do Design Thinking durante o processo de desenvolvimento de software, enquanto a Seção 4.1.2 apresenta os papéis exercidos no Design Thinking. A Seção 4.1.3 apresenta as práticas relacionadas ao Design Thinking para produção de software, enquanto a Seção 4.1.4 apresenta o relacionamento dos modelos de DT na produção de software. A Seção 4.1.5 apresenta as ferramentas utilizadas no desenvolvimento de software relacionadas com o Design Thinking, e por fim, a seção 4.1.6 apresenta a avaliação da qualidade dos artigos selecionados.

Mediante a execução da string de busca, foram selecionadas 890 publicações para análise inicial (1º seleção). As buscas pelos estudos primários foram realizadas mediante o protocolo apresentado na seção de métodos. Após a realização para seleção dos estudos, 95 artigos foram selecionados como relevantes ao tema. A Tabela 5 apresenta os resultados do processo de seleção.

Tabela 5 – Resultados do processo de seleção dos estudos primários

Fonte	Resultados da 1ª seleção	Irrelevante	Resultados da 2ª seleção	Percentual de inclusão
ACM	167	157	10	10,53%
IEEE Explore	475	428	47	49,47%
Scopus	248	209	38	40,00%
Total	890	794	95	100,00%

A primeira coluna aborda as bases de dados acadêmicos utilizadas neste estudo, onde

foram realizadas as buscas manuais. A segunda coluna apresenta o número total de artigos encontrados no processo de busca. A terceira coluna mostra o número de artigos selecionados após a primeira etapa do processo de seleção, que consistiu na avaliação do título, resumo e palavras-chave para excluir estudos irrelevantes para este mapeamento sistemático. A quarta, apresenta o número de artigos excluídos após a 1ª etapa do processo de seleção. Por fim, a última coluna apresenta o percentual do número de artigos incluídos no mapeamento de acordo com a base de dados acadêmicos. A seguir, apresentamos os resultados encontrados para as questões de pesquisa propostas neste estudo.

4.1.1 Desafios da utilização de DT no desenvolvimento de Software

Alguns autores informam que a implementação do DT no processo de desenvolvimento de software é um desafio, pois em muitos casos, é necessário lidar com diferentes culturas, idiomas, costumes, fatores econômicos e entre outros fatores (PEREIRA, 2021). A abordagem se torna ainda mais complexa quando o DT é implementado a outros métodos (CARROLL, 2016), onde os agentes participantes precisam ter domínio a respeito do conceito do DT em conjunto com outros modelos. Fatores associados à dificuldade em identificar os interesses dos usuários durante as fases do DT também foram encontrados durante a execução deste estudo (SURESH et al., 2021). A Tabela 6 apresenta os principais desafios encontrados durante o mapeamento sistemático. Foram encontrados 29 desafios, sendo a sua maioria, com uma forte relação ao aspecto de desenvolvimento de software, podendo observar que existem muitos problemas estruturais que podem ser encontrados desde a fase de planejamento até a fase de desenvolvimento e testes. Embora os desafios associados ao processo de desenvolvimento sejam maioria, existem fatores ligados à abordagem UCD, como por exemplo a transparência na comunicação entre as partes interessadas (Stakeholders) e os times, fatores culturais, senso de urgência e relevância e entre outros. A listagem completa dos desafios encontrados na realização deste mapeamento sistemático pode ser apreciada no apêndice B deste trabalho.

Tabela 6 – Desafios na adoção do Design Thinking

ID	Desafio	Descrição
1	O envolvimento do cliente	Envolver o cliente no processo de desenvolvimento, visando compreender as suas necessidades.
2	Participação das Partes Interessadas	A falta de interação das partes interessadas no processo de desenvolvimento pode se tornar um desafio.
3	Implementação de Design Thinking com outros métodos.	Implementação de Design Thinking com outros métodos.
4	Identificar hábitos/interesses de consumo do usuário	Identificação de hábitos e interesses dos usuários.
5	Valor agregado	Envolver o cliente no processo de desenvolvimento, visando entender suas necessidades.
6	Senso de Urgência e Relevância	A falta de senso de urgência e priorização de requisitos pode se tornar um desafio.
7	Envolvimento do Cliente	Envolver o cliente e chamar sua atenção para algo que ele deseja é um processo desafiador.
8	Orçamento limitado	Orçamentos limitados podem se tornar desafios iminentes no processo de desenvolvimento de software.
9	Desafios Geográficos e Culturais	Interação entre pessoas geograficamente distantes e diversidades culturais podem se tornar um desafio.
10	Separação do trabalho de UX Designer e Desenvolvedores	Segmentar trabalhos associados a designers e trabalhos relacionados a desenvolvedores.

4.1.2 Papéis exercidos no Design Thinking

A Tabela 7 retrata os principais papéis encontrados ao implementar DT no processo de Software, o contexto Ágil tem se tornado cada vez comum no processo de desenvolvimento de software, em virtude da sua capacidade de mudanças durante as fases dos projetos (Dobrigkeit 2019), associados a esta metodologia, temos as figuras do Scrum Master, Product Owner, e Development Teams, que buscam cada vez mais entender as necessidades de seus respectivos usuários e implementá-las aos projetos. UX Designer são papéis cada vez mais comuns no processo de software, visto que a prototipagem faz parte de uma das suas principais atribuições. Neste mapeamento também encontramos papéis que estavam diretamente ligados ao desenvolvimento de software, sendo eles desenvolvedores, QA Specialist, Software Engineer, e entre outros. Apesar de não haver nenhum impedimento destes papéis em utilizar o design thinking durante as suas atividades, isto ainda é uma variante, pois estão mais ligados ao desenvolvimento software do que as necessidades do usuário, ao todo foram encontrados 60 papéis que possuem relação com o Design Thinking e o desenvolvimento de software, estas evidências estão contidas no apêndice C deste documento.

Tabela 7 – Papéis no Uso do Design Thinking no Desenvolvimento de Software

ID	Papel	Descrição
1	Designer	O designer é o profissional responsável pela concepção de um produto. Ele compreende os conceitos artísticos e funcionais, projeta e idealiza um objeto utilitário com fins definidos.
2	Development Team	Responsáveis por realizar o trabalho de desenvolvimento do produto de ponta a ponta.
3	Usuário	Responsável pela utilização do produto.
4	Scrum Master (SM)	O Scrum Master é o responsável por garantir que o Time Scrum se oriente pelos valores e práticas do Scrum.
5	Product Owner (PO)	O Product Owner representa os interesses de todos os envolvidos (Stakeholders), define as funcionalidades do produto e prioriza os itens de Product Backlog.
6	Product Manager	Responsável por planejar e executar a estratégia por trás do produto, determinar suas funcionalidades e viabilizar o acesso aos recursos necessários para a sua produção.
7	Project Manager	Responsável por estar à frente de um projeto, liderando e coordenando todas as atividades. Trata-se do líder do projeto, que conduz a equipe na busca pelo resultado.
8	Software Engineer	Responsável por projetar e guiar o desenvolvimento de sistemas, aplicativos e programas
9	User experience Designer	Responsável por conduzir a pesquisa de usuário, prototipagem e outros processos para garantir o máximo de usabilidade, utilidade e acessibilidade na experiência do usuário.
10	Marketing Analyst	Responsável por implementar todas as técnicas que farão o consumidor se atrair por uma marca e querer comprar o produto ou serviço ofertado.

4.1.3 Práticas relacionadas ao Design Thinking para produção de software

O Design Thinking possui uma abordagem centrado no usuário, logo busca compreender as necessidades dos mesmos, autores como BEKELE (2019) e DA SILVA (2011) abordaram três práticas importantes ao serem adotadas no processo de desenvolvimento de software, sendo elas: Entender e especificar o contexto de uso do produto, especificar os requisitos do usuário, produzir soluções de design para atender aos requisitos do usuário e efetuar avaliação do design em relação aos requisitos. Através dos resultados do nosso estudo, foi possível listar um conjunto das principais práticas associadas a DT que podem ser implementadas no processo de desenvolvimento de software, conforme retrata a Tabela 8 deste documento. No apêndice D foram listadas o conjunto de práticas encontradas durante a realização deste estudo. É importante destacar que tivemos ao todo 116 práticas encontradas durante a execução do mapeamento sistemático, sendo estas, de grande impacto ao desenvolvimento de software.

Tabela 8 – Principais práticas utilizadas no Design Thinking

ID	Prática	Descrição
1	Prototipagem	A prototipagem é a criação de um protótipo, que permite que uma ideia de produto seja colocada em prática.
2	Estudos de Usabilidade	É uma técnica de avaliação e pesquisa que analisa se a experiência é boa, ruim, fácil, difícil, lenta ou demorada por meio de tarefas simples que o usuário precisa realizar.
3	Feedbacks	Comunicação feita entre duas ou mais pessoas, em que uma delas é avaliada pelas demais em relação às suas ações, comportamentos, tarefas, entre outros.
4	Entrevistas preliminares	Estas são as primeiras entrevistas feitas no processo de criação e idealização de projetos/produtos.
5	Personas	Persona é a representação fictícia do cliente ideal de um negócio.
6	Brainstorming	É uma atividade destinada a explorar o potencial criativo de um indivíduo ou de um grupo.
7	Storyboards	São organizadores gráficos como uma série de ilustrações ou imagens dispostas em sequência com o objetivo de visualizar um filme, animação ou gráfico animado, incluindo elementos interativos em sites.
8	Foco nas necessidades reais do usuário	Para projetar e criar ótimas experiências de usuário, precisamos entender as necessidades do usuário e, para isso, focar nelas. Com isso em mente, podemos aprender, desenvolver e alcançar o fluxo de trabalho perfeito para os usuários.
9	Gamificação	É o uso de técnicas de design de jogos que utilizam a mecânica do jogo e o pensamento orientado para o jogo para enriquecer diversos contextos normalmente não relacionados aos jogos.
10	Coleta de dados com usuários	A coleta de dados é o processo de coletar, medir e analisar dados de várias fontes para obter insights. Os dados podem ser coletados através de várias fontes
11	Objetivos da experiência do usuário	Uma meta de experiência do usuário é uma escolha feita por sua equipe de produto sobre que tipo de experiência você deseja que seus usuários tenham com seu produto ou serviço.
12	Histórias de usuários com critérios de aceitação	Estes são os principais formatos de documentação de requisitos. Embora as histórias de usuário descrevam exatamente o que o usuário deseja que o sistema faça, o objetivo dos critérios de aceitação é explicar as condições que uma história de usuário específica deve satisfazer.

4.1.4 Modelos de DT e processos de desenvolvimento de software

Com o avanço da necessidade de interação global para as práticas de desenvolvimento de software, o modelo ágil tem se tornando cada vez mais evidente neste cenário (MARRU,

2021), mas o DT não fica limitado apenas a este modelo, ele também pode ser implementado a projetos com outros modelos, e projetos que utilizam a abordagem UCD (User Center Design) para desenvolvimento de suas tarefas (GURAN, 2020). Durante a execução deste estudo, 47 modelos foram encontrados, a Tabela 9 destaca os principais modelos encontrados no mapeamento sistemático. No apêndice E, representamos todos os modelos encontrados durante a realização deste estudo.

Tabela 9 – Principais Modelos associados ao Design Thinking

ID	Modelo	Descrição
1	Design Thinking Model	Abordagem centrada no ser humano para a inovação - ancorada na compreensão das necessidades do cliente, prototipagem rápida e geração de ideias criativas.
2	UCD - User Centred Design	Coleção de processos que se concentram em colocar os usuários no centro do design e desenvolvimento do produto.
3	Iterative Model	Desenvolvimento iterativo e incremental é qualquer combinação de design iterativo ou método iterativo e modelo de construção incremental para desenvolvimento.
4	Human Computer Interaction (HCI)	Ele se concentra no design da tecnologia de computador e, em particular, na interação entre humanos (os usuários) e computadores.
5	Agile Model	É um processo de desenvolvimento de software incremental e iterativo onde cada iteração é considerada um “quadro” curto de tempo, que normalmente dura de duas a quatro semanas.
6	Model-based Systems Engineering (MBSE)	É a aplicação formal de modelagem para dar suporte aos requisitos do sistema, projeto, análise, verificação e atividades de validação começando na fase de projeto conceitual e continuando através do desenvolvimento e estágios posteriores do ciclo de vida.
7	Design And Development Research (DDR)	O foco do DDR está no uso, design, desenvolvimento, implementação e avaliação de produtos, ferramentas, programas e modelos usando modelos e estruturas de design instrucional.
8	ADDIE Model	É uma estrutura de design de sistemas instrucionais que muitos designers instrucionais e desenvolvedores de treinamento usam para desenvolver projetos.
9	Nordstrom Model	Combinação de Design Thinking, Metodologia Ágil e Lean Startup.
10	Model-driven engineering (MDE)	Criação de modelos para artefatos da engenharia de software, utilizados para analisar, simular e raciocinar sobre as propriedades do sistema em desenvolvimento.

4.1.5 Ferramentas para integrar o desenvolvimento de software com Design Thinking

Tabela 10 – Principais ferramentas associados ao Design Thinking

ID	Ferramenta	Tipo	Descrição
1	Figma	Prototipagem	Figma é um editor gráfico de vetor e prototipagem de projetos de design baseado principalmente no navegador web.
2	Sketch	Prototipagem	Sketch é um editor de gráficos vetoriais para o sistema operacional macOS.
3	Adobe Xd	Design / Edição de Mídia	Adobe XD é uma ferramenta de design de experiência do usuário baseada em vetores para aplicativos da web e aplicativos móveis.
4	Illustrator	Design / Edição de Mídia	Adobe Illustrator é um aplicativo de gráficos vetoriais padrão do setor que permite criar logotipos, ícones, desenhos, tipografia e ilustrações complexas para qualquer meio.
5	Photoshop	Design / Edição de Mídia	É um software desenvolvido pela Adobe Systems, para edição de imagens.
6	Microsoft Visual Studio	IDE	O Visual Studio permite que você conclua todo o ciclo de desenvolvimento em um único lugar. Por exemplo, você pode editar, depurar, testar, controlar a versão e implantar na nuvem
7	Phyton	Linguagem de Programação	O Python é uma linguagem de programação amplamente usada em aplicações da Web, desenvolvimento de software, ciência de dados e machine learning (ML).
8	Google Apresentações	Gestão de Tarefas	O Apresentações Google é um programa de apresentação incluído como parte do pacote gratuito de editores de documentos do Google baseado na Web oferecido pelo Google.
9	Miro	Gestão de Tarefas	O Miro é uma plataforma de comunicação visual que funciona como uma lousa interativa digital.
10	Javascript	Linguagem de Programação	JavaScript é uma linguagem de programação interpretada estruturada, de script em alto nível com tipagem dinâmica fraca e multiparadigma. Juntamente com HTML e CSS, o JavaScript é uma das três principais tecnologias da World Wide Web.

Após executar o mapeamento sistemático, um conjunto de 75 ferramentas foram encontradas, incluindo ferramentas para a criação de protótipos, ferramentas de comunicação, gerenciador de tarefas, editores de texto e mídias, etc. Todas essas categorias de ferramentas podem ser utilizadas para o desenvolvimento de atividades de DT. Na Tabela 10, destacamos

as principais ferramentas encontradas. A maior parte das ferramentas estão associadas ao processo de prototipação, fase esta que serve para validar com os usuários finais, se os passos a serem desenvolvidos satisfazem os interesses das partes interessadas. Em seguida, as linguagens de programação aparecem compondo 13% dos resultados de citação nos estudos selecionados, elas são as precursoras no processo de desenvolvimento de software. Ferramentas de Edição de Mídia e Design possuem também um alto grau de relevância, compondo 33% de citação nos resultados obtidos através do mapeamento sistemático. As IDE's (Integrated Development Environment), compõem 8% das citações dos artigos, e estão relacionadas unicamente à fase de desenvolvimento de software. As ferramentas que não se encaixam em nenhuma dessas classificações, compõem 13% das citações dos artigos relacionados, sendo estas, associadas a editores de textos, ferramentas de gestão de tarefas, softwares educacionais e etc. A Tabela 11 descreve a porcentagem de citação de ferramentas nos resultados deste estudo.

Tabela 11 – Percentual de ferramentas em artigos selecionados

Tipo de Ferramenta	Citação em artigos selecionados(%)
Prototipagem	33%
Linguagem de programação	13%
Design / Edição de Mídia	33%
IDE	8%
Outros	13%

4.1.6 Avaliação de qualidade e Alinhamento dos Estudos Primários

Esta seção descreve a avaliação dos artigos selecionados, bem como as suas respectivas classificações. Em nosso estudo tivemos um quantitativo de 95 artigos selecionados, estes foram classificados em Regular, Bom ou Ótimo. Os artigos classificados como “Regular” possuem notas compreendidas de 0 até 3.5, enquanto os artigos classificados como “Bom” possuem notas compreendidas entre 4 até 5.5, por fim, os artigos classificados como “Ótimo”, possuem notas de 6 até 8, sendo 8 a pontuação máxima para a classificação. A Figura 6 apresenta o quantitativo dos artigos mediante a sua classificação.

As notas foram atribuídas mediante a resposta do formulário de extração, que tem como objetivo avaliar individualmente se o artigo responde às questões de pesquisas definidas, além de classificar a qualidade de cada um deles. O apêndice A apresenta o formulário para avaliação dos artigos.

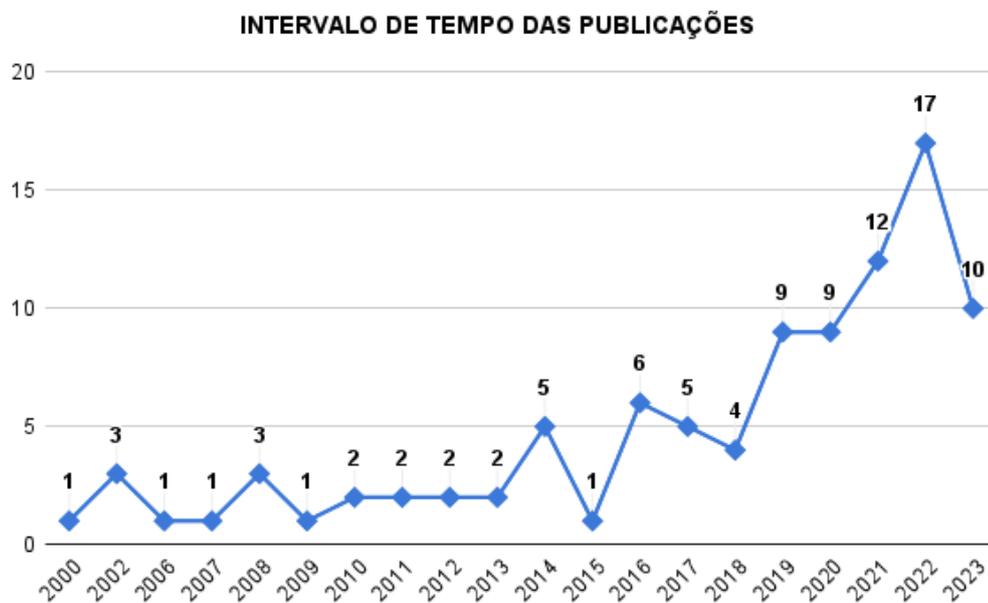
Em relação ao número de publicações de estudos primários relacionados ao Design Thinking, a imagem a seguir (Figura 7) aborda, mediante os resultados encontrados neste trabalho, a quantidade de artigos publicados nos últimos 23 anos que apresentam o DT no desenvolvimento de software.

Figura 6 – Classificação da qualidade dos artigos selecionados

Classificação	Nota	Quantidade
Regular	0 até 3,5	12
Bom	4 até 5,5	17
Ótimo	6 até 8	57

Apesar de uma pequena amostragem baseada nos artigos selecionados deste estudo, pode-se observar um aumento no número de artigos publicados que empregam conceitos e práticas de Design Thinking em suas atividades, além de apresentar quais ferramentas, modelos, práticas e desafios encontrados ao adotar DT no desenvolvimento ágil de software. Este resultado confirma que os usuários de softwares estão cada vez mais em busca de recursos e serviços que atendam às suas necessidades digitais de maneira eficiente. Um dos meios de atender estas expectativas é através do desenvolvimento de soluções centradas nas necessidades do usuário, o que comumente neste estudo é referenciado como abordagens UCD (User Center Design).

Figura 7 – Número de publicações por ano dos artigos relacionados ao DT



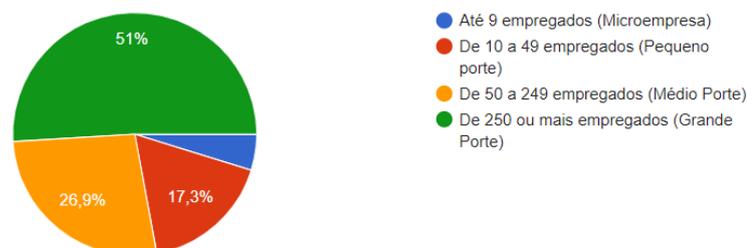
4.2 RESULTADOS DO SURVEY

Os resultados obtidos a partir da realização do *survey* em empresas de desenvolvimento de software forneceram uma visão sobre práticas, modelos, papéis, ferramentas e desafios enfrentados na adoção do Design Thinking na produção de software. Essa abordagem permitiu avaliar a adoção de metodologias e práticas, e compreender como diferentes papéis utilizam o Design Thinking nas organizações. Os dados coletados possibilitam análises detalhadas, contribuindo para o entendimento do uso do DT no desenvolvimento de software. A Seção 4.2.1 apresenta os resultados gerais obtidos com a aplicação do *survey*. A Seção 4.2.2 aborda as respostas à Questão de Pesquisa N°1, enquanto a Seção 4.2.3 discute as respostas à Questão de Pesquisa N°2. As Seções 4.2.4, 4.2.5 e 4.2.6 detalham as respostas às Questões de Pesquisa N°3, N°4 e N°5, respectivamente.

4.2.1 Dados Gerais - Survey

Esta seção apresenta os resultados gerais da pesquisa realizada. As respostas às perguntas são fornecidas em formatos textuais e gráficos, facilitando uma compreensão abrangente dos dados. Em relação à primeira pergunta relacionada às localizações dos participantes, a maioria das respostas foi do Brasil, com apenas um pequeno número de participantes indicando sua atividade profissional em outros países. Em relação ao tamanho da empresa, os dados indicaram que a maioria dos entrevistados eram empregados por organizações com mais de 250 funcionários. Em seguida, vieram empresas com 50 a 249 funcionários. Além disso, uma proporção menor de participantes trabalhavam em empresas com 10 a 49 funcionários, enquanto apenas uma minoria estava associada a organizações com até 9 funcionários, conforme apresentado na Figura 8. Essa distribuição de tamanhos de empresa fornece um contexto valioso para a compreensão dos ambientes organizacionais onde o Design Thinking pode ser utilizado.

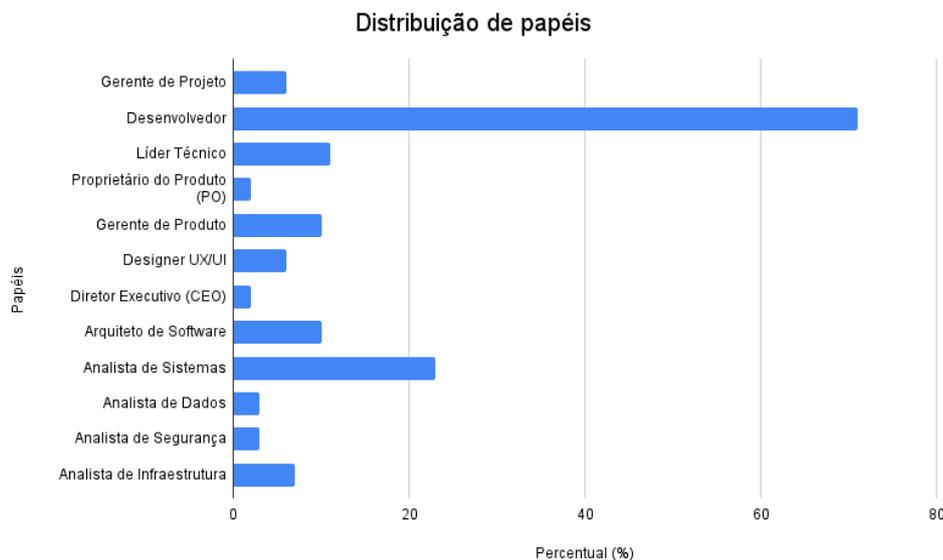
Figura 8 – Tamanho da empresa dos participantes



A análise de dados do questionário pré-eliminatório revelou as funções mais proeminentemente representadas em ambientes de desenvolvimento de software que empregam metodologias de Design Thinking. Os resultados são ilustrados na Figura 9. A questão foi elaborada para obter múltiplas respostas para obter uma compreensão abrangente das

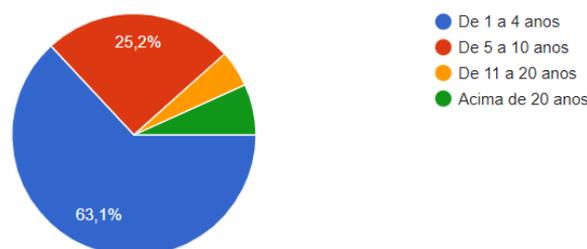
várias funções que cada participante ocupava. Os resultados revelaram que a maioria dos entrevistados se identificou como desenvolvedores, seguidos por analistas de sistemas, líderes técnicos e arquitetos de software.

Figura 9 – Distribuição de papéis



Além disso, este estudo buscou entender o nível de experiência dos participantes em empresas de desenvolvimento de software, retratadas na figura 10. Os níveis de senioridade dos participantes são uma consideração crítica para aqueles que buscam implementar os modelos ou práticas de Design Thinking. A maioria dos participantes relatou ter de 1 a 4 anos de experiência em desenvolvimento de software, enquanto um número significativo indicou entre 5 e 10 anos de experiência. Além disso, um grupo menor possuía de 11 a 20 anos de experiência, e apenas uma minoria tinha mais de 20 anos na área. Essa distribuição de experiência fornece um contexto valioso para analisar as perspectivas e insights compartilhados pelos participantes em relação ao uso do Design Thinking no desenvolvimento de software.

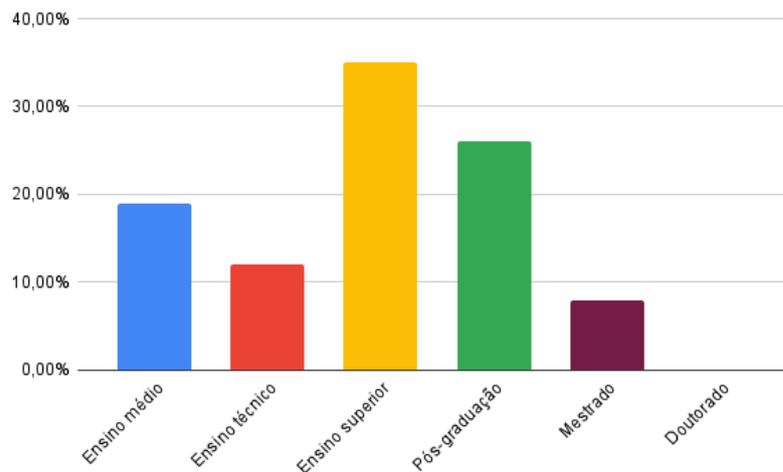
Figura 10 – Experiência dos participantes com o desenvolvimento de software



Os dados indicam que a maioria dos entrevistados concluiu o ensino superior, seguido por profissionais com qualificações de pós-graduação. Além disso, é digno de nota que alguns

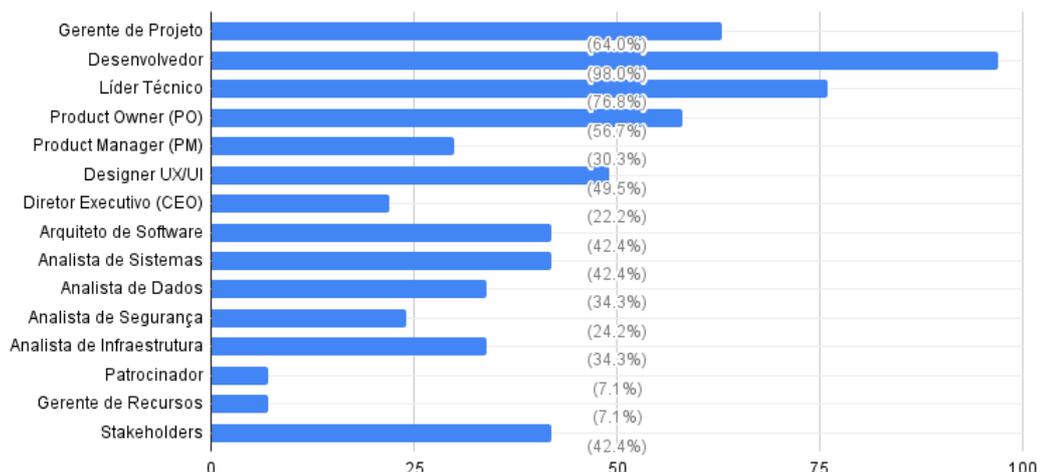
participantes possuíam apenas o diploma de ensino médio, enquanto outros concluíram o diploma de nível técnico. Essa variação nas formações educacionais ressaltam as diversas qualificações dos participantes, o que pode influenciar em suas perspectivas e experiências neste contexto. Essas informações são apresentadas na Figura 11 e essa composição de papéis destaca a natureza colaborativa da dinâmica de equipe nas organizações e os diversos conjuntos de habilidades que contribuem para o sucesso do projeto.

Figura 11 – Nível educacional dos participantes



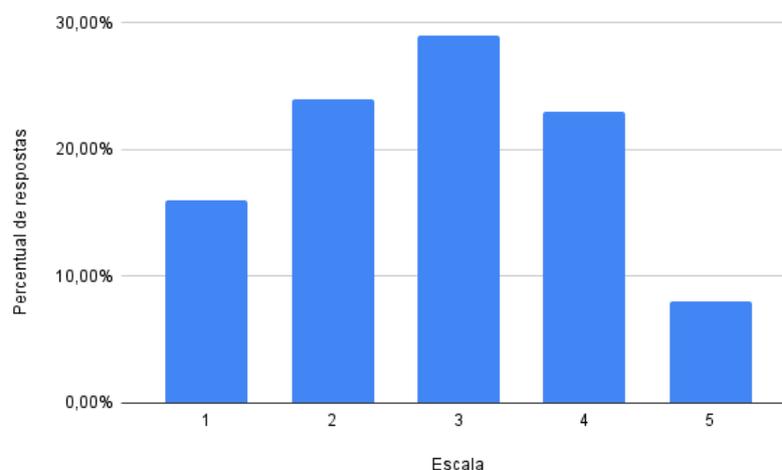
Além disso, em resposta à pergunta “Quais papéis/funções normalmente compõem as equipes em sua empresa?”, as descobertas indicaram que a maioria das equipes incluía as seguintes funções: Gerente de Projeto, Desenvolvedor, Líder Técnico, Product Owner (PO), Product Manager (PM), Designer, Diretor Executivo, Arquiteto de Software, Analista de Sistemas, Analista de Dados, Analista de Segurança da Informação, Analista de Infraestrutura, Patrocinador, Gerente de Recursos e Garantia de Qualidade de Software, entre outros. A figura 12 apresenta o conjunto de respostas dos participantes para este assunto.

Figura 12 – Composição dos times de tecnologia



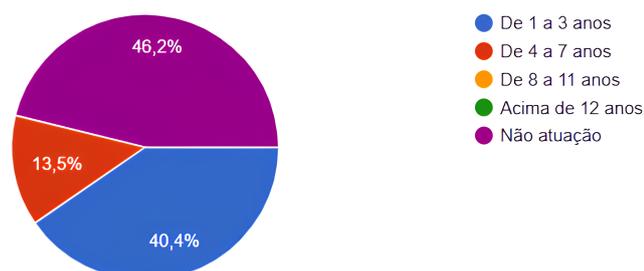
O estudo em questão também revelou o grau de conhecimento sobre Design Thinking entre os participantes da pesquisa. Os respondentes tiveram a oportunidade de indicar, em uma escala de 1 a 5, seu nível de familiaridade com o tema, sendo 1 equivalente a pouco conhecimento e 5 a um conhecimento elevado. Observa-se que a maioria dos participantes indicou possuir um nível intermediário de conhecimento em Design Thinking, conforme apresentado na figura a seguir (figura 13). Essa distribuição indica uma familiaridade variada com o conceito entre os participantes, o que pode influenciar sua capacidade de se envolver e implementar metodologias de Design Thinking em suas respectivas funções.

Figura 13 – Nível de Conhecimento em Design Thinking



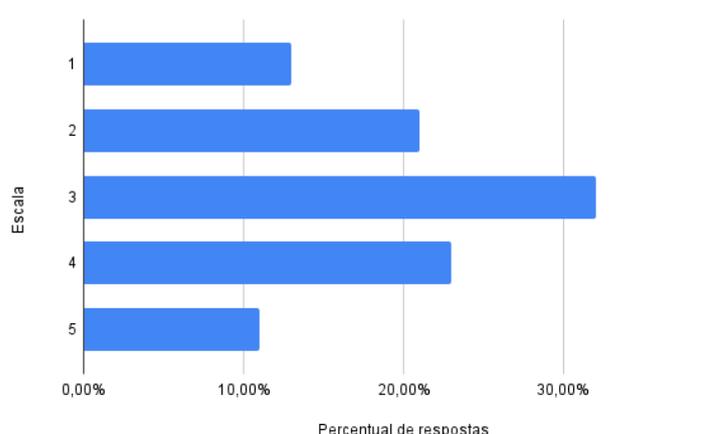
Em resposta à pergunta “Há quanto tempo você trabalha com Design Thinking na produção de software?”, a figura 14 apresenta que aproximadamente 46,2% dos participantes indicaram que não se envolvem com Design Thinking em suas funções atuais. Enquanto isso, 13,5% relataram ter de 4 a 7 anos de experiência usando Design Thinking, e 40,4% declararam que aplicaram os princípios do Design Thinking em seu trabalho por 1 a 3 anos. Essa distribuição destaca uma proporção significativa de participantes que não estão utilizando Design Thinking atualmente, ao mesmo tempo em que mostra uma gama diversificada de experiência entre aqueles que o fazem.

Figura 14 – Tempo de Experiência com Design Thinking na Produção de Software



De forma semelhante à questão que avalia o grau de conhecimento sobre Design Thinking entre os participantes, a realização do estudo também buscou compreender como eles percebem a adoção dessa metodologia na produção de software nas empresas em que atuam. O objetivo desta pergunta foi verificar se há uma alta utilização dos conceitos de Design Thinking na produção de software. Os participantes responderam a essa questão em uma escala de 1 a 5, sendo 1 correspondente a pouca utilização dos conceitos de Design Thinking e 5 a uma utilização intensa desses conceitos na produção de software, a figura 15 destaca percepções variadas da adoção do Design Thinking em diferentes organizações.

Figura 15 – Adoção do Design Thinking na Produção de Software

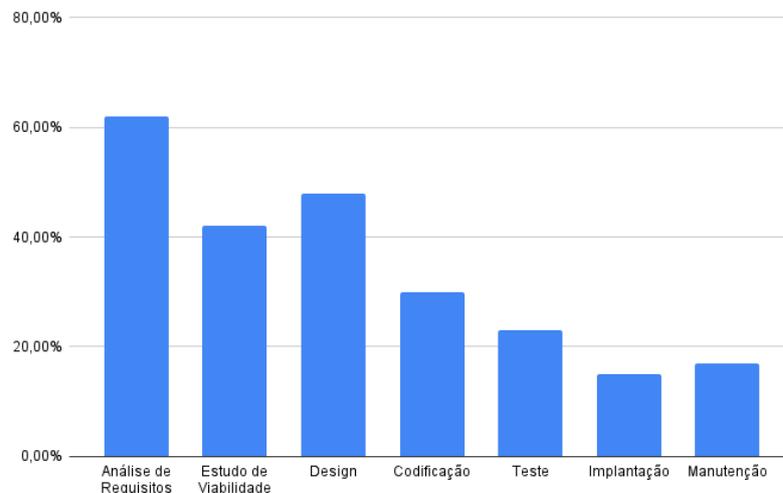


A questão subsequente abordou as fases do desenvolvimento de software nas quais a empresa implementou o Design Thinking, confirmando que o Design Thinking é comumente aplicado em todos os estágios do processo de desenvolvimento. Os resultados indicam que as fases de Análise de Requisitos e Design são as mais frequentemente associadas ao Design Thinking, seguidas de perto pelas fases de Estudo de Viabilidade e Codificação. Essa aplicação generalizada do Design Thinking em vários estágios ressalta seu papel integral no ciclo de vida do desenvolvimento de software. Essas informações são apresentadas na Figura 16.

Em resposta à questão “Em sua atuação com o Design Thinking, quais são as práticas mais utilizadas em seus projetos?”, tivemos como resultado as seguintes práticas: Prototipação (64%), Estudos de Usabilidade (49,5%), Cultura de Feedback (47,6%), Brainstorming (43,7%), Foco nas Necessidades do Cliente (50,5%), Entrega Contínua de Versões (45,6%), Definição de Objetivos de Negócio (33,0%), Planejamento Adequado (29,1%), Storyboards (23,3%), Personas (23,3%), Entrevistas Preliminares (23,3%), Suporte Executivo (6,8%), Gamificação (4,9%), Grupo Focal (1,0%) e “Não é utilizada” (3,0%). Este resultado destaca os termos que mais se destacam entre os participantes e reflete a ênfase deles nas principais práticas de Design Thinking.

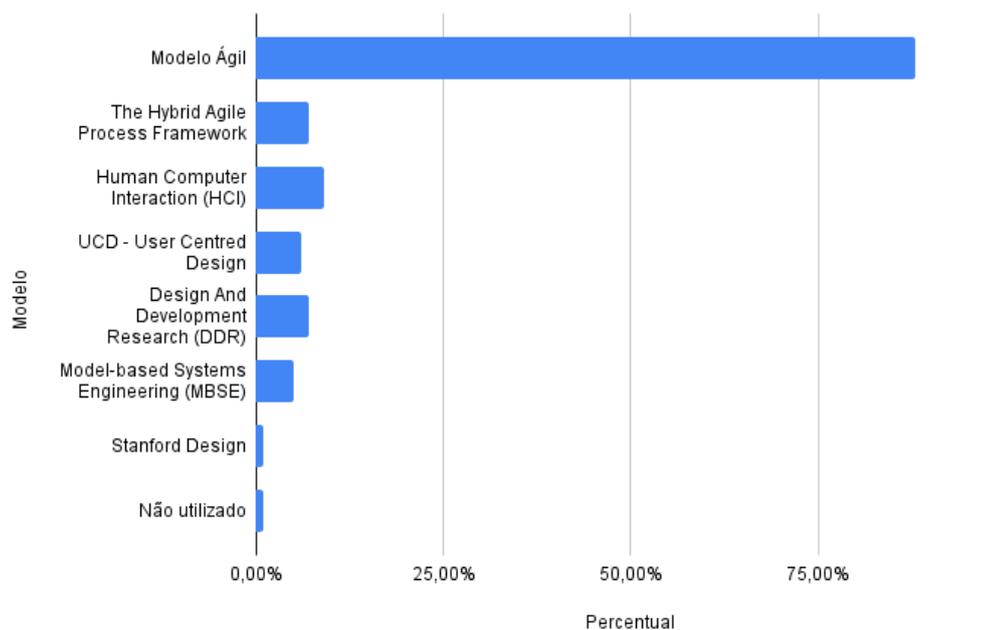
Em resposta à distribuição de metodologias e frameworks utilizados neste estudo, destacou-se quais modelos são mais frequentemente utilizados em conjunto com o Design

Figura 16 – Fases do Desenvolvimento de Software com a abordagem Design Thinking



Thinking em contextos de desenvolvimento de software, incluindo as seguintes respostas: Modelo Ágil (86,3%), The Hybrid Agile Process Framework (6,9%), Human Computer Interaction (HCI) (8,8%), UCD - User Centred Design (5,9%), Design And Development Research (DDR) (6,9%), Model-based Systems Engineering (MBSE) (4,9%), Stanford Design (1,0%) e Não Utilizado (1,0%). A figura 17 destaca a predominância do Modelo Ágil em conjunto com o Design Thinking.

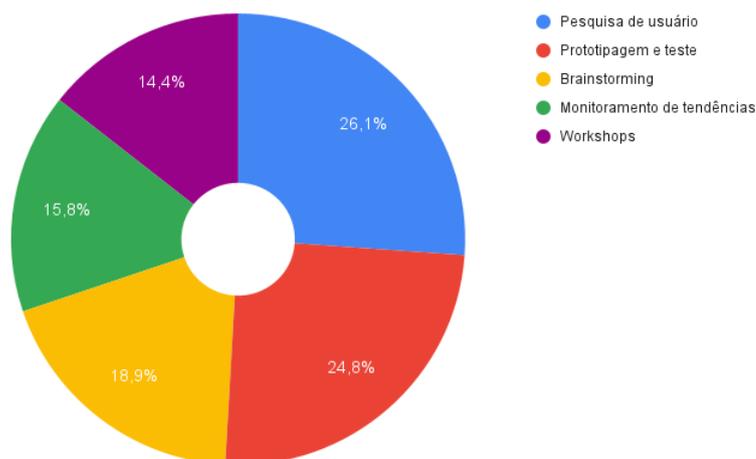
Figura 17 – Modelos e Metodologias Aplicados na Produção de Software



A questão “Como a empresa busca novas práticas, metodologias e processos para aprimorar seu processo de Design Thinking?” teve como objetivo explorar de que maneira as organizações procuram novos recursos para atender às necessidades dos usuários de

seus produtos. Os resultados indicam que a maioria das empresas se envolve em pesquisas com usuários, bem como em prototipagem e testes para aprimorar suas práticas de Design Thinking. Além disso, muitas organizações utilizam sessões de brainstorming, monitoram tendências e participam de eventos como parte de sua busca por metodologias inovadoras. Isso evidencia uma abordagem proativa por parte das empresas para refinar seus processos de Design Thinking em resposta às demandas dos usuários. A Figura 18 retrata esta abordagem.

Figura 18 – Abordagens Empresariais para a Melhoria do Processo de Design Thinking



A pergunta seguinte, “Como as metodologias e modelos associados ao Design Thinking atendem às necessidades do seu projeto?” teve como objetivo avaliar as percepções dos participantes em relação à eficácia do alinhamento entre as metodologias do Design Thinking e o sucesso de seus projetos. Esta investigação buscou elucidar o quão bem essas metodologias ressoam com os requisitos do projeto dos participantes. A investigação sobre os principais desafios encontrados pelas equipes ao utilizar o Design Thinking teve como objetivo identificar e corroborar questões que foram discutidas anteriormente na literatura. Os desafios mais frequentemente citados incluíram o senso de urgência e relevância, a necessidade de envolver os usuários no processo, a reconciliação do Design Thinking com as atividades de desenvolvimento de software e a participação das partes interessadas ao longo do processo. Essas descobertas destacam barreiras críticas que as equipes enfrentam na implementação eficaz das metodologias do Design Thinking. A Figura 19 apresenta os principais desafios enfrentados pelas equipes ao usar o Design Thinking.

A pergunta “Quais ferramentas alinhadas ao Design Thinking são utilizadas em seus projetos de desenvolvimento de software?” provocou uma variedade de respostas, com inúmeras ferramentas sendo mencionadas em diferentes fases, incluindo prototipagem, planejamento e execução de tarefas. A Figura 20 ilustra a gama diversificada de ferramentas empregadas em conjunto com o Design Thinking em contextos de tecnologia. Esta visualização destaca a natureza multifacetada da integração do Design Thinking nas ferramentas de desenvolvimento de software.

Figura 19 – Principais Desafios Enfrentados pelas Equipes na Utilização do Design Thinking

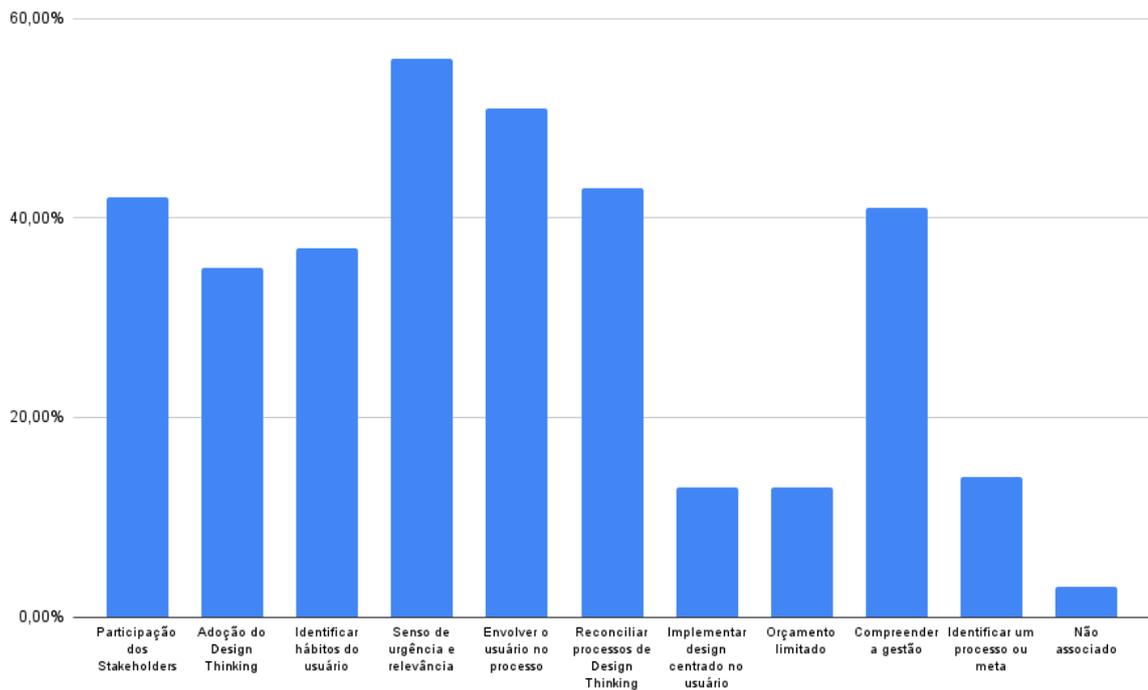
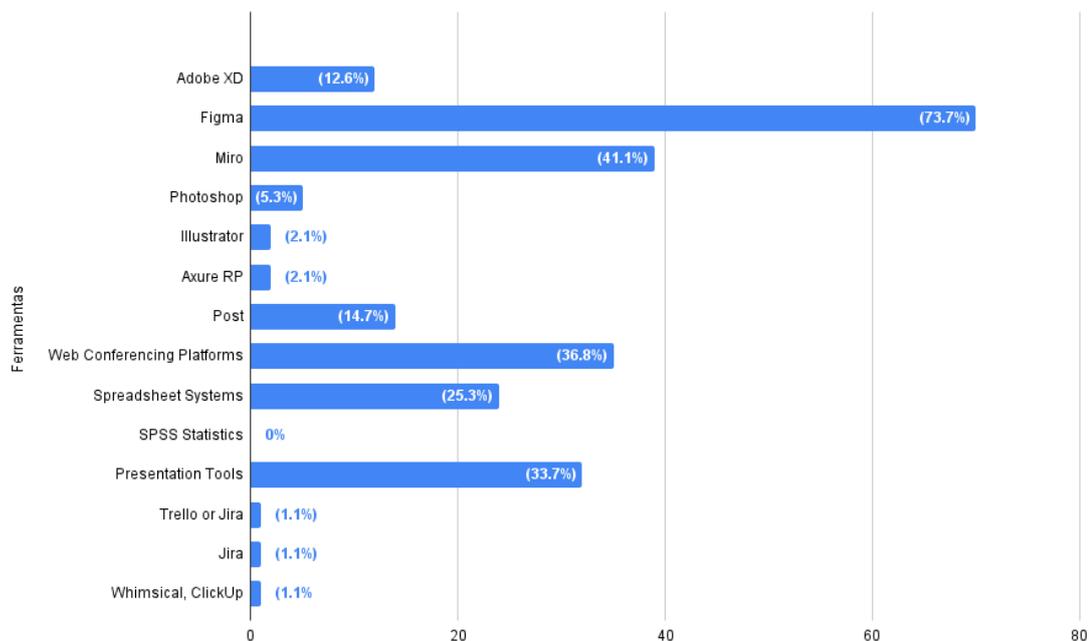


Figura 20 – Ferramentas Complementares ao Design Thinking



Na pergunta final, sobre “Quais tipos de projetos de software são desenvolvidos?”, os resultados indicaram que os cinco tipos de projetos mais frequentemente encontrados incluem aplicativos da web, aplicativos móveis, sistemas ERP, sistemas de inteligência artificial

e sistemas de gerenciamento de banco de dados. Essa distribuição fornece insights valiosos sobre as tendências predominantes no cenário de desenvolvimento de software e enfatiza a gama diversificada de projetos sendo realizados por organizações.

As descobertas deste estudo são baseadas nas respostas de profissionais de tecnologia ativamente envolvidos em ambientes de desenvolvimento de software. À medida que a demanda global por produtos que efetivamente atendam às necessidades do usuário continua a crescer, os insights coletados neste estudo posicionam o Design Thinking como uma abordagem viável que foi adotada em vários segmentos de empresas de tecnologia. Esta implementação do Design Thinking reflete um compromisso contínuo com o aprimoramento de soluções centradas no usuário em resposta aos requisitos de mercado em evolução.

4.2.2 Resposta a Questão de pesquisa N°1 - Survey

Em resposta à questão de pesquisa N°1: “Quais evidências apoiam o uso do Design Thinking no desenvolvimento de software?”, examinamos até que ponto a empresa incorporou o Design Thinking em várias fases do desenvolvimento de software, juntamente com os esforços das empresas para aprimorar seus processos de Design Thinking por meio da adoção de novas práticas e metodologias. Uma observação significativa é que o Design Thinking é altamente enfatizado na fase de Análise de Requisitos (72,9%), o que se alinha estreitamente com o foco na Pesquisa do Usuário (58,5%) como um método para melhorar as práticas do Design Thinking. Ambos refletem uma forte ênfase na compreensão das necessidades do usuário no início do processo de desenvolvimento. Em contraste, a aplicação do Design Thinking nas fases de Codificação (35,3%) e Teste (27,1%) é menos expressiva. No entanto, a prática relacionada de Prototipagem e Teste (55,3%) é altamente priorizada para melhorias, sugerindo que, embora o Design Thinking não esteja totalmente integrado aos estágios posteriores do desenvolvimento, as empresas estão ativamente buscando formas de aprimorá-lo por meio da prototipagem. Outra lacuna importante é observada nas fases de Implantação e Manutenção, onde o Design Thinking é minimamente aplicado. Essa limitação pode ser superada com um maior foco no monitoramento de tendências e na realização de brainstorming, visando identificar abordagens inovadoras que possam ser incorporadas após o desenvolvimento. A Tabela 12 apresenta essas informações.

A Tabela 13 apresenta os dados fornecidos pelos participantes da pesquisa quanto ao nível de conhecimento sobre Design Thinking. Pode-se observar que há uma limitação na compreensão do uso do DT em relação à produção de software, pois apenas 8% dos participantes relataram ter um nível excelente de conhecimento sobre Design Thinking. Esta informação torna-se relevante visto que a falta de entendimento dos processos de Design Thinking por permitir uma adoção do mesmo conturbada ou ineficiente, o que pode acarretar em fraco aproveitamento dos seus benefícios dentro da produção de software.

Tabela 12 – Fases e Aperfeiçoamento do Processo de Design Thinking

Fases de desenvolvimento de software em que as empresas utilizaram o Design Thinking	Como as empresas buscam novas práticas, metodologias para aprimorar seu processo de Design Thinking
Análise de Requisitos - (72.9%) Estudo de viabilidade - (49.4%) Design - (56.5%) Codificação - (35.3%) Teste - (27.1%) Manutenção - (20%)	Pesquisa do usuário - (58.5%) Prototipagem e Testes - (55.3%) Brainstorming - (40.4%) Monitoramento de tendências - (35.1%) Workshops - (31.9%) Não utiliza - (3.3%)

Tabela 13 – Nível de Conhecimento em Design Thinking no Desenvolvimento de Software

Conhecimento sobre Design Thinking
Excelente (8%) Muito bom (23%) Intermediário (30%) Um pouco bom (24%) Nenhum (15%)

As respostas ao formulário evidenciam a prevalência do modelo ágil nas metodologias utilizadas por empresas que atuam no desenvolvimento de software. Embora tenha sido disponibilizado um campo aberto para que os participantes adicionassem outros modelos utilizados em suas empresas, não foram recebidas respostas adicionais. Também buscou-se compreender como as empresas dos participantes adotam novas práticas, metodologias e processos para aprimorar seu processo de Design Thinking.

4.2.3 Resposta a Questão de pesquisa N°2 - Survey

Em resposta à questão de pesquisa n° 2: “Quais são os principais desafios encontrados na utilização de Design Thinking (DT) na produção de software?”, identificou-se que os principais desafios no desenvolvimento de software envolvem a definição de um senso de urgência e relevância nas atividades do projeto, bem como o engajamento do usuário no processo de criação do software.

A relação entre esses desafios e o uso do Design Thinking está diretamente conectada, uma vez que as dificuldades enfrentadas geralmente dizem respeito à compreensão dos interesses dos envolvidos no projeto e à superação de suas expectativas. No entanto, observa-se que, segundo os participantes da pesquisa, a adoção do Design Thinking apresenta um grau intermediário de eficácia na superação dos desafios mencionados. Essa perspectiva pode estar associada com a evidência da questão anterior, onde nem todos os participantes da pesquisa possuem um entendimento aprofundado sobre Design Thinking.

Como alternativa para visualizar os desafios identificados, a Tabela 14 apresenta

o relacionamento entre as práticas associadas a cada desafio do *Survey*. Por meio dessa abordagem, é possível observar como cada desafio se manifesta no desempenho das práticas associadas ao Design Thinking no contexto do desenvolvimento de software. Compreender a relação entre cada desafio permite definir estratégias mais eficazes para os usuários do Design Thinking, a fim de otimizar a aplicação dessas práticas nos processos de desenvolvimento de software.

Tabela 14 – Desafios associados a práticas - Survey

Desafios - Survey	Práticas Associadas - Survey
Participação dos Stakeholders no processo	Prototipagem, Estudos de usabilidade, Cultura de Feedbacks, Storyboards, Personas, Brainstorming, Foco nas necessidades dos usuários, Suporte Executivo, Definição dos objetivos de negócio, Requisitos básicos firmes, Planejamento Adequado
Adoção do Design Thinking com outros métodos	Prototipagem, Estudos de usabilidade, Cultura de Feedbacks, Gamificação, Storyboards, Personas, Brainstorming, Entrevistas Preliminares, Foco nas necessidades dos usuários, Focal Group, Suporte Executivo, Definição dos objetivos de negócio de forma clara, Entrega contínua de versões, Requisitos básicos firmes, Planejamento Adequado
Identificar os hábitos dos usuários	Prototipagem, Estudos de usabilidade, Cultura de Feedbacks, Gamificação, Storyboards, Personas, Brainstorming, Entrevistas Preliminares, Foco nas necessidades dos usuários, Focal Group, Suporte Executivo, Definição dos objetivos de negócio de forma clara, Entrega contínua de versões, Requisitos básicos firmes, Planejamento Adequado
Senso de urgência e relevância	Estudos de usabilidade, Cultura de Feedbacks, Storyboards, Personas, Brainstorming, Foco nas necessidades dos usuários, Suporte Executivo, Definição dos objetivos de negócio, Requisitos básicos firmes, Planejamento Adequado
Envolver o usuário no processo	Prototipagem, Estudos de usabilidade, Cultura de Feedbacks, Gamificação, Storyboards, Personas, Brainstorming, Entrevistas Preliminares, Foco nas necessidades dos usuários, Focal Group, Suporte Executivo, Definição dos objetivos de negócio de forma clara, Entrega contínua de versões, Requisitos básicos firmes, Planejamento Adequado
Conciliar DT com as tarefas de software	Prototipagem, Estudos de usabilidade, Cultura de Feedbacks, Gamificação, Storyboards, Personas, Brainstorming, Entrevistas Preliminares, Foco nas necessidades dos usuários, Focal Group, Suporte Executivo, Definição dos objetivos de negócio de forma clara, Entrega contínua de versões, Requisitos básicos firmes, Planejamento Adequado
Implementar design centrado no usuário com outras metodologias	Prototipagem, Estudos de usabilidade, Cultura de Feedbacks, Gamificação, Storyboards, Personas, Brainstorming, Entrevistas Preliminares, Foco nas necessidades dos usuários, Focal Group, Suporte Executivo, Definição dos objetivos de negócio de forma clara, Entrega contínua de versões, Requisitos básicos firmes, Planejamento Adequado
Orçamento limitado	Prototipagem, Estudos de usabilidade, Entrega contínua de versões
Compreensão da gestão (Prazos/Prioridades/Definição clara dos requisitos do projeto)	Prototipagem, Estudos de usabilidade, Cultura de Feedbacks, Gamificação, Storyboards, Personas, Brainstorming, Entrevistas Preliminares, Foco nas necessidades dos usuários, Focal Group, Suporte Executivo, Definição dos objetivos de negócio de forma clara, Entrega contínua de versões, Requisitos básicos firmes, Planejamento Adequado
Identificar um processo ou guia de referência para algumas fases de DT	Prototipagem, Estudos de usabilidade, Cultura de Feedbacks, Gamificação, Storyboards, Personas, Brainstorming, Entrevistas Preliminares, Foco nas necessidades dos usuários, Focal Group, Suporte Executivo, Definição dos objetivos de negócio de forma clara, Entrega contínua de versões, Requisitos básicos firmes, Planejamento Adequado

4.2.4 Resposta a Questão de pesquisa N°3 - Survey

Em relação à questão de pesquisa nº 3: “Quais recursos do Design Thinking costumam ser utilizados para otimizar a construção de software?”, observou-se que a etapa do desenvolvimento de software em que o Design Thinking é mais comumente aplicado é a de análise de requisitos. Essa aplicação está fortemente associada ao conceito de ideação no Design Thinking, uma vez que, na fase de ideação, conforme descrito na Seção 2, ocorre o processo de geração, desenvolvimento e teste de ideias — etapas fundamentais para a definição dos requisitos dos projetos.

Outro aspecto importante é a prototipação, que apresentou o maior percentual de respostas dos participantes ao serem questionados sobre as práticas mais utilizadas em

seus projetos. Esse fator é positivo, pois reflete os conceitos centrais do Design Thinking que podem ser incorporados ao processo de desenvolvimento de software. A Figura 21 apresenta o conjunto de práticas mencionadas nas respostas do formulário da pesquisa, relacionando-as com as fases dos projetos em que estão evidenciadas. As fases dos projetos foram definidas como: Brainstorming, Prototipagem, Desenvolvimento e Teste. Embora tenha sido disponibilizado um campo aberto para que os participantes sugerissem novas práticas associadas ao uso do Design Thinking, não foram registradas contribuições nesse sentido.

Figura 21 – Utilização das Práticas de Design Thinking em Diferentes Fases do Projeto

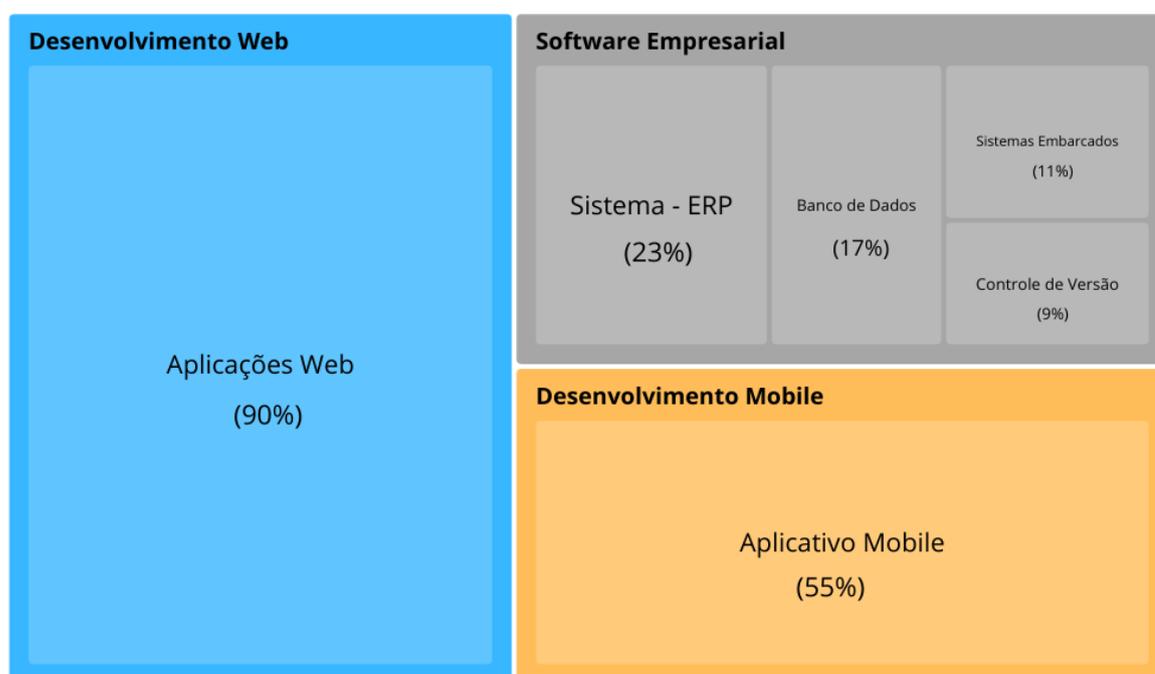
Práticas - Survey	Fase do projeto associada- Survey
Prototipagem	Prototipagem, Brainstorming
Estudos de usabilidade	Prototipagem, Brainstorming
Cultura de Feedbacks	Prototipagem, Brainstorming, Desenvolvimento, Teste
Gamificação	Brainstorming, Desenvolvimento
Storyboards	Prototipagem, Brainstorming
Personas	Prototipagem, Brainstorming
Brainstorming	Brainstorming, Desenvolvimento
Entrevistas Preliminares	Brainstorming, Desenvolvimento
Foco nas necessidades dos usuários	Prototipagem, Brainstorming, Desenvolvimento, Teste
Focal Group	Brainstorming
Suporte Executivo	Prototipagem, Brainstorming, Desenvolvimento, Teste
Definição dos objetivos de negócio de forma clara	Prototipagem, Brainstorming
Entrega contínua de versões	Desenvolvimento, Teste
Requisitos básicos firmes	Prototipagem, Brainstorming
Planejamento Adequado	Prototipagem, Brainstorming

4.2.5 Resposta a Questão de pesquisa N°4 - Survey

Em resposta à pergunta de pesquisa “Quais são os tipos de projeto em que DT é utilizado?”, constatou-se que 90% dos participantes atuam em projetos de desenvolvimento de aplicações web, seguidos por projetos de aplicações mobile. Alinhados aos dados contidos em nossa Seção 1.1 que menciona a necessidade constante das empresas em desenvolver produtos cada vez mais usuais e homogêneos, aplicações web e mobile são comumente associadas à conceitos de UX Design e práticas de prototipação para que possam promover maior interatividade entre os usuários. Observou-se uma quantidade relevante de respostas a sistemas ERP, que são sistemas de gerenciamento empresarial, isto porque o Design Thinking também está imerso no contexto empresarial, trazendo abordagens centradas no usuário para o ambiente corporativo. A Figura 22 apresenta o percentual de respostas relacionadas aos tipos de projetos de software nos quais os participantes atuam. Os projetos foram categorizados em três grupos principais: Desenvolvimento Web, Desenvolvimento

Mobile e Software Empresarial. Para cada grupo, foram definidos subgrupos com base nas respostas dos participantes, sendo eles: Aplicações Web, Aplicativo Mobile, Sistema ERP, Banco de Dados, Sistemas Embarcados e Controle de Versão. Houve também um único participante que indicou sua atuação no desenvolvimento de software SaaS (Software as a Service); no entanto, como sua representatividade foi muito baixa, esse resultado não está sendo apresentado na Figura em questão. Nesta questão de pesquisa, os participantes podiam selecionar mais de uma opção. Conforme ilustrado na Figura 22, entre os 105 participantes, 90% indicaram atuação em projetos de aplicações Web, 55% em projetos Mobile, e 23% em projetos relacionados a sistemas ERP, entre outras categorias mencionadas.

Figura 22 – Categorias de Projetos que Incorporam Design Thinking



4.2.6 Resposta a Questão de pesquisa N°5 - Survey

Em resposta à pergunta de pesquisa “Quais papéis são mais frequentemente envolvidos no Design Thinking dentro do desenvolvimento de software?”, observou-se neste estudo uma expressiva participação de desenvolvedores de software, seguidos por analistas de sistemas, conforme ilustrado na Figura 9. Esses resultados indicam que o Design Thinking pode ser aplicado em empresas, abrangendo tanto áreas técnicas quanto setores responsáveis pela gestão e governança de projetos.

Com base no corpo de evidências da aplicação deste estudo, observou-se como os papéis estão associados aos desafios encontrados na adoção do Design Thinking no processo de software. Estes papéis estão associados a várias frentes de atuação do processo de software, mostrando que o sucesso no uso de Design Thinking requer uma colaboração ampla entre equipes de gestão, desenvolvimento e design, além de uma coordenação estratégica entre as lideranças da empresa. Para a maioria dos desafios, os papéis cobrem tanto gestores estratégicos (CEO, CTO) quanto profissionais de desenvolvimento técnico e design. Isso sugere que a solução de problemas e implementação de métodos de design exigem uma interação forte entre as áreas. O envolvimento de diretores de tecnologia (CTO, CIO) em quase todos os desafios evidencia a necessidade de decisões estratégicas e bem informadas sobre a direção do projeto e a adoção de metodologias, como o Design Thinking. Conciliar o Design Thinking com métodos tradicionais de desenvolvimento de software, bem como envolver o usuário no processo, são desafios centrais. Isso demonstra a importância de uma integração forte entre a equipe técnica e a equipe de design. A Figura 23 apresenta os papéis relacionados a cada um desses desafios. Os papéis repetem-se em grande parte, sugerindo uma visão multifuncional e destacando a importância da colaboração em equipes com diferentes perfis.

Figura 23 – Papéis e Desafios na Adoção do Design Thinking

Desafios - Survey	Papéis Relacionados - Survey
A participação dos Stakeholders no processo	Gerente de Projetos, Product Owner (PO), Gerente de Produto, Chief Executive Officer (CEO), Chief Technology Officer(CTO), Chief Information Officer(CIO)
Adoção do Design Thinking com outros métodos	Gerente de Projetos, Desenvolvedor, Lider Técnico, Product Owner (PO), Gerente de Produto, UX/UI Designer, Arquiteto de Software, Analista de Sistemas
Identificar os hábitos dos usuários	Gerente de Projetos, Product Owner (PO), Gerente de Produto, Chief Executive Officer (CEO), Chief Technology Officer(CTO), Chief Information Officer(CIO)
Senso de urgência e relevância	Gerente de Projetos, Desenvolvedor, Lider Técnico, Product Owner (PO), Gerente de Produto, UX/UI Designer, Arquiteto de Software, Analista de Sistemas, Analista de Dados, Analista de Segurança da Informação, Analista de Infraestrutura
Envolver o usuário no processo	Gerente de Projetos, Product Owner (PO), Gerente de Produto, Chief Executive Officer (CEO), Chief Technology Officer(CTO), Chief Information Officer(CIO)
Conciliar os processos de Design com as tarefas de desenvolvimento de software	Gerente de Projetos, Desenvolvedor, Lider Técnico, Product Owner (PO), Gerente de Produto, UX/UI Designer, Arquiteto de Software, Analista de Sistemas, Analista de Dados, Analista de Segurança da Informação, Analista de Infraestrutura
Implementar design centrado no usuário com outras metodologias	Gerente de Projetos, Desenvolvedor, Lider Técnico, Product Owner (PO), Gerente de Produto, UX/UI Designer, Arquiteto de Software, Analista de Sistemas
Orçamento limitado	Gerente de Projetos, Product Owner (PO), Gerente de Produto, Chief Executive Officer (CEO), Chief Technology Officer(CTO), Chief Information Officer(CIO)
Compreensão da gestão (Prazos/Prioridades/Definição clara dos requisitos do projeto)	Gerente de Projetos, Product Owner (PO), Gerente de Produto, Chief Executive Officer (CEO), Chief Technology Officer(CTO), Chief Information Officer(CIO)
Identificar um processo ou guia de referência para algumas fases de DT	Gerente de Projetos, Desenvolvedor, Lider Técnico, Product Owner (PO), Gerente de Produto, UX/UI Designer, Arquiteto de Software, Analista de Sistemas, Analista de Dados, Analista de Segurança da Informação, Analista de Infraestrutura

5 ANÁLISE E DISCUSSÃO DOS PADRÕES DE EVIDÊNCIAS IDENTIFICADOS

O objetivo deste capítulo é apresentar as análises realizadas com base nos padrões de evidências obtidos ao longo da execução deste estudo. Os dados expostos consistem nas análises derivadas dos resultados do Mapeamento Sistemático e do Survey. Os temas abordados estão organizados nas seguintes seções: Análise Geral dos Resultados (Seção 5.1), Análise dos Resultados dos Estudos Experimentais (Seção 5.2), Padrões de Evidências (Seção 5.3), e por fim, Limitações Gerais desta Pesquisa (Seção 5.4).

5.1 ANÁLISE GERAL DOS RESULTADOS

Esta seção tem como objetivo analisar de forma geral os resultados obtidos neste estudo, à luz dos objetivos propostos e da literatura revisada. A análise permitirá identificar padrões e implicações dos achados para o campo de estudo. Os resultados indicaram que a adoção de Design Thinking no desenvolvimento de software impacta diretamente a colaboração entre equipes, a solução criativa de problemas e redução de custos a mitigação de problemas de usabilidade dos projetos. Além disso, identificou-se a necessidade de um conjunto ferramentas que auxiliem o processo de prototipação e ideação em ambientes de desenvolvimento de software.

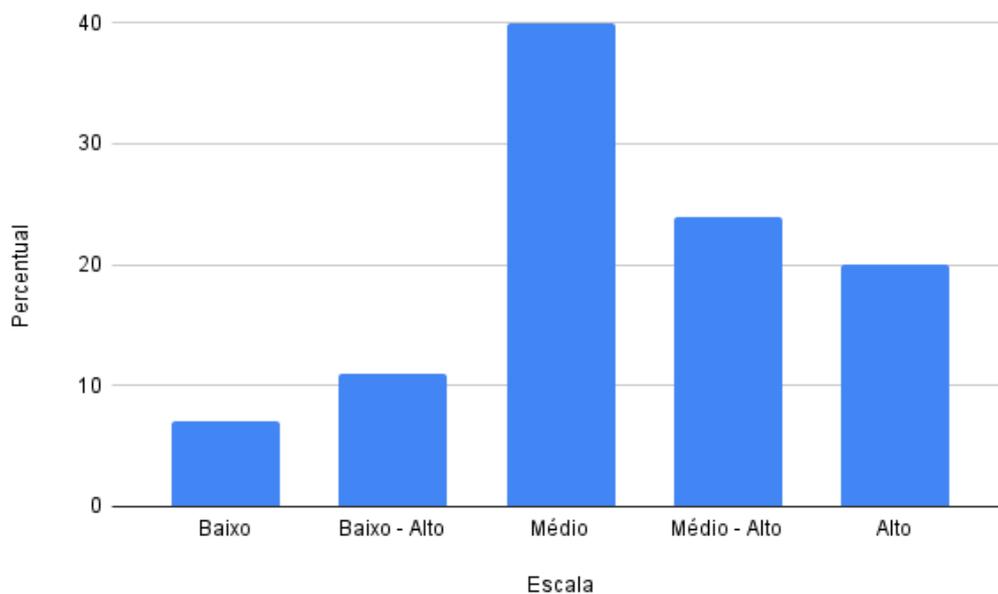
Em comparação com estudos de autores como DOBRIGKEIT (2019) e PEREIRA (2018) abordados no capítulo de Trabalhos Relacionados (Capítulo 6), nossos resultados reforçam a importância do Design Thinking para o engajamento das equipes de desenvolvimento. No entanto, diferentemente de ALMEIDA (2019), que aponta dificuldades em definir indicadores específicos para projetos de DT, nossos resultados indicam uma adaptação mais fluida quando os conceitos de DT são utilizados nas fases de análise de requisitos e estudos de usabilidade. Apesar dos resultados positivos, algumas limitações devem ser consideradas. Por exemplo, a atualização das bases científicas limitando o quantitativo dos achados associados aos estudos primários do Mapeamento Sistemático, a diversidade da composição dos times de tecnologia, especialmente em organizações com diferentes níveis de conhecimento sobre Design Thinking. Além disso, os dados qualitativos sugerem que a integração de DT com metodologias ágeis, porém a efetividade do uso de DT com outras metodologias merece uma investigação mais aprofundada visto que houve uma baixa quantidade de evidências neste sentido.

Esses conjuntos de evidências sugerem que a adoção dos conceitos de DT pode trazer benefícios consideráveis para o desenvolvimento de software, especialmente em cenários onde deseja-se aumentar a colaboração entre equipes de tecnologia e cultivar ideais

associadas as perspectivas dos usuários nos requisitos de software dos projetos. No campo teórico, os resultados abrem novas possibilidades para estudos futuros que investiguem a adaptação do Design Thinking com outros métodos de desenvolvimento de software em diferentes contextos organizacionais.

Diante disso, as questões de pesquisa dos experimentos serão explanadas a seguir. No Survey, a questão geral “Como utilizar os recursos de Design Thinking de forma satisfatória na indústria de software?” foi discutida e avaliada por meio das respostas às questões objetivas, e principalmente sobre o uso do Design Thinking nas fases do desenvolvimento de software. O objetivo foi compreender em quais etapas do processo o Design Thinking apresentava maior evidência de aplicação. Os resultados indicaram que o Design Thinking mostrou-se promissor, especialmente nas fases de definição e planejamento de projetos, uma vez que sua abordagem colaborativa permite o agrupamento de ideias e facilita a definição de requisitos com base em sua metodologia UCD (User Center Design), que coloca o usuário no centro do projeto. Embora haja uma tendência de utilização do Design Thinking nas fases iniciais, sua abordagem permite a absorção em outras etapas do desenvolvimento de software; contudo, os resultados nessas fases foram menos promissores. Com o intuito de avaliar a satisfação dos participantes quanto ao uso adequado do Design Thinking, a questão objetiva “Em seu contexto profissional, como metodologias/modelos associados ao Design Thinking atendem às necessidades do seu projeto?” obteve uma resposta intermediária da maior parte dos participantes, conforme ilustrado no gráfico da Figura 24.

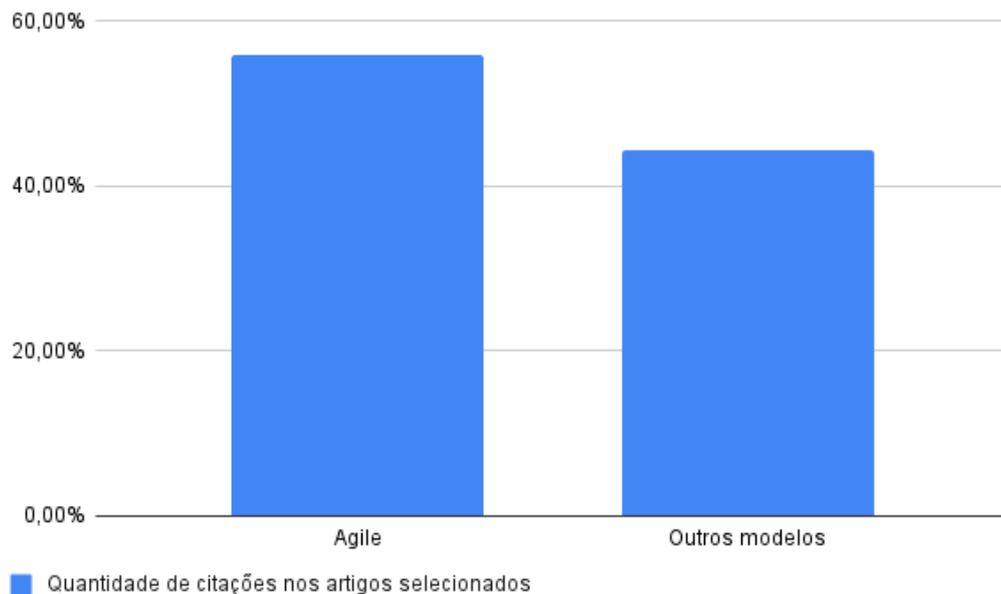
Figura 24 – Desafios Relacionadas a Outras Questões de Pesquisa



Na questão geral do Mapeamento Sistemático, “Como o Design Thinking afeta o processo de desenvolvimento de software?”, observou-se que o Design Thinking (DT) surge como uma alternativa para promover interações entre o usuário final e os demais membros

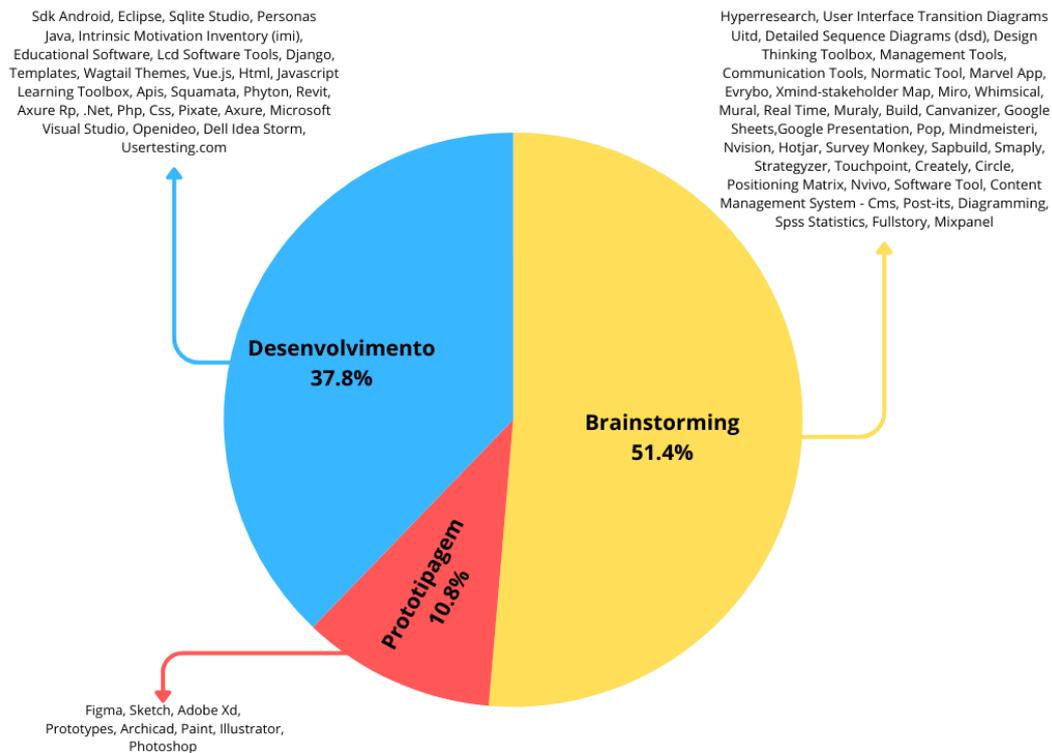
da equipe de desenvolvimento de software, visando a produção de resultados que atendam às necessidades dos usuários. Embora haja uma grande variação nos modelos em que os estudos primários aplicam o Design Thinking, o contexto ágil apresenta o maior percentual de citações e utilizações entre os estudos selecionados, conforme ilustrado na figura 25.

Figura 25 – Citação dos modelos nos estudos selecionados



Com base nesses resultados, constatou-se que o Design Thinking também contribui para o processo de desenvolvimento de software em diferentes fases do projeto. Por meio do uso de ferramentas, essas etapas podem ser melhor trabalhadas com o objetivo de trazer à tona as perspectivas do usuário, mitigando a produção de soluções inadequadas. A Figura 26 apresenta as ferramentas associadas ao Design Thinking que foram comumente utilizadas em cada fase do desenvolvimento de software, confirmando como o DT pode impactar positivamente todas as etapas, especialmente nas fases de planejamento e definição de soluções.

Figura 26 – Citação dos Modelos Nos Estudos Seleccionados



5.2 ANÁLISE DOS RESULTADOS DOS ESTUDOS EXPERIMENTAIS

Os resultados obtidos através da execução do mapeamento sistemático mostram o que vem sendo utilizado no cenário mundial de desenvolvimento de software, onde o usuário precisa cada vez mais estar em evidência no processo de desenvolvimento de software para que os produtos tenham boa aceitação por seus consumidores. Com isso, muitas práticas, modelos e ferramentas têm sido utilizadas para suprir essa necessidade de superar os desafios em projetar produtos que promovam melhores experiências para o usuário. No entanto, trazer o usuário para o centro do desenvolvimento de software não é uma tarefa simples, os processos e papéis que atuam neste cenário precisam ser bem definidos para que isso não impacte no resultado final do produto. Dentro da engenharia de software, que compõe a linha de pesquisa deste estudo, a DT tem se tornado uma alternativa para solução de problemas ao adicionar o usuário ao processo de desenvolvimento de software.

Este corpo de evidências é uma proposição de mapeamento e é construído a partir das evidências empíricas encontradas nos estudos primários e em respostas ao formulário Survey aplicado para profissionais da área de TI, o que fornece algum nível de confiança em sua validade. Além disso, as evidências declaradas são apenas exemplos de possíveis relações entre tópicos fundamentais no processo de desenvolvimento de software, neste caso, referindo-se às outras quatro questões de pesquisa. A lista final abrangendo os 95 estudos selecionados pode ser avaliada no site a seguir: (<https://abrir.link/QhnEF>).

Por meio dos resultados encontrados no Mapeamento Sistemático após as fases de seleção foi possível observar que a maioria dos autores adotaram a metodologia de estudo de caso para estruturação dos artigos, correspondendo a 56% dos estudos encontrados após a segunda fase de seleção do mapeamento. Métodos de experimentação corresponderam a 36% dos resultados e questionários corresponderam a 8%.

A Figura 27, mostra a relação entre as práticas encontradas neste estudo e os resultados encontrados em outras questões de pesquisa. A primeira coluna aborda conceitualmente as boas práticas associadas à DT no processo de desenvolvimento de software. As colunas seguintes apresentam os identificadores dos desafios listados no conjunto de evidências, bem como os identificadores das funções, modelos e ferramentas, os identificados mencionados podem ser visualizados através dos Apêndices B, C, D, E e F. Em geral, as práticas apresentadas podem ser facilmente implementadas em muitos cenários, incluindo desenvolvimento de software. Além disso, há a possibilidade de usar várias combinações entre os resultados encontrados. A adoção dessas práticas também pode ajudar a lidar com os desafios encontrados para abordagens centradas no usuário no processo de software. De forma semelhante, a Figura 28, aborda como os desafios encontrados no processo de desenvolvimento de software com Design Thinking estão associados às demais questões de pesquisa. Por meio das evidências listadas, é possível observar que existem ferramentas e práticas que auxiliam a minimizar os desafios enfrentados durante o processo de software com DT, além disso, existem modelos de DT que podem orientar a melhor execução deste processo e os papéis desempenhados tentando garantir mais qualidade no processo de software.

Com relação ao Survey, os dados obtidos com a realização da aplicação do formulário permitiu coletar uma ampla gama de informações sobre os efeitos da utilização do Design Thinking na indústria de software. Ao apresentar estes resultados buscou-se agrupar padrões do uso do DT que atendam as necessidades e expectativas dos usuários, minimizando custos e mantendo as empresas em competição de mercado com a produção de soluções inovadoras. Análises descritivas foram realizadas para reunir insights sobre métricas de utilidade e satisfação. Ao empregar essas metodologias, a pesquisa não apenas identifica a prevalência do Design Thinking no processo de desenvolvimento de software, mas também fornece uma compreensão aprofundada das etapas de desenvolvimento de software, ferramentas e papéis associados, contribuindo, em última análise, com conhecimento valioso para o campo. A execução do Survey promoveu uma confirmação em grande parte dos achados do Mapeamento Sistemático, com diferenças observadas apenas em resultados onde os participantes indicaram a não utilização dos conceitos de Design Thinking no desenvolvimento de software. No entanto, o percentual dessas respostas foi significativamente baixo em comparação com as demais opções da pesquisa. Apesar da pesquisa promover uma interação com os participantes, promovendo campos onde os mesmos poderiam indicar novas variações de respostas, não houve nenhum avanço neste aspecto, tendo uma variação muito pequena entre o que realmente acontece na prática da indústria de software que utiliza

Figura 27 – Práticas Relacionadas a Outras Questões de Pesquisa

Práticas	[ID] Desafio Associado	[ID] Papel Associado	[ID] Modelo Associado	[ID] Ferramenta Associada
P1. Identificação de Personas	[1,2,9,10,12,14,17,19,21,25,26,27,28]	[3,4,5,6,7,8,9,22,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,51,60]	[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41]	[4,16,43,35,50,51]
P2. Especificação dos Requisitos do Usuário	[1,2,9,10,12,14,17,19,21,25,26,27,28]	[2,14,15,16,17,24,25,31,41,48,57]	[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41]	[4,16,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,56,60,63,64,65,66,74,75]
P3. Metas de Usabilidade	[1,2,9,10,12,14,17,19,21,25,26,27,28]	[3,4,5,6,7,8,9,22,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,51,60]	[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41]	[4,16,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,56,60,63,64,65,66,74,75]
P4. Interação entre equipes	[1,2,3,4,5,6,8,9,10,11,12,13,14,15,17,18,19,20,21,22,23,24,25,26,27,28]	[2,14,15,16,17,24,25,31,41,48,57]	[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41]	[4,16,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,56,60,63,64,65,66,74,75]
P5. Design	[1,2,9,10,12,14,17,19,21,25,26,27,28]	[3,4,5,6,7,8,9,22,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,51,60]	[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41]	[4,16,43,35,50,51]
P6. Documentação de Processos	[1,2,3,4,5,6,8,9,10,11,12,13,14,15,17,18,19,20,21,22,23,24,25,26,27,28]	[2,14,15,16,17,24,25,31,41,48,57]	[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41]	[4,16,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,56,60,63,64,65,66,74,75]
P7. Documentação de Tarefas	[1,2,3,4,5,6,8,9,10,11,12,13,14,15,17,18,19,20,21,22,23,24,25,26,27,28]	[2,14,15,16,17,24,25,31,41,48,57]	[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41]	[4,16,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,56,60,63,64,65,66,74,75]
P8. Plano do Projeto	[1,2,3,4,5,6,8,9,10,11,12,13,14,15,17,18,19,20,21,22,23,24,25,26,27,28]	[2,14,15,16,17,24,25,31,41,48,57,49,50,56]	[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41]	[4,16,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,56,60,63,64,65,66,74,75]
P9. Protótipo	[1,2,3,4,5,6,8,9,10,11,12,13,14,15,17,18,19,20,21,22,23,24,25,26,27,28]	[3,4,5,6,7,8,9,22,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,51,60]	[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41]	[4,16,43,35,50,51]
P10. Teste	[1,3,7,8,11,20,23,24,27]	[1,12,13,21,44,53,54,55]	[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41]	[1,2,3,5,11,12,13,21,24,61,62,67,70]

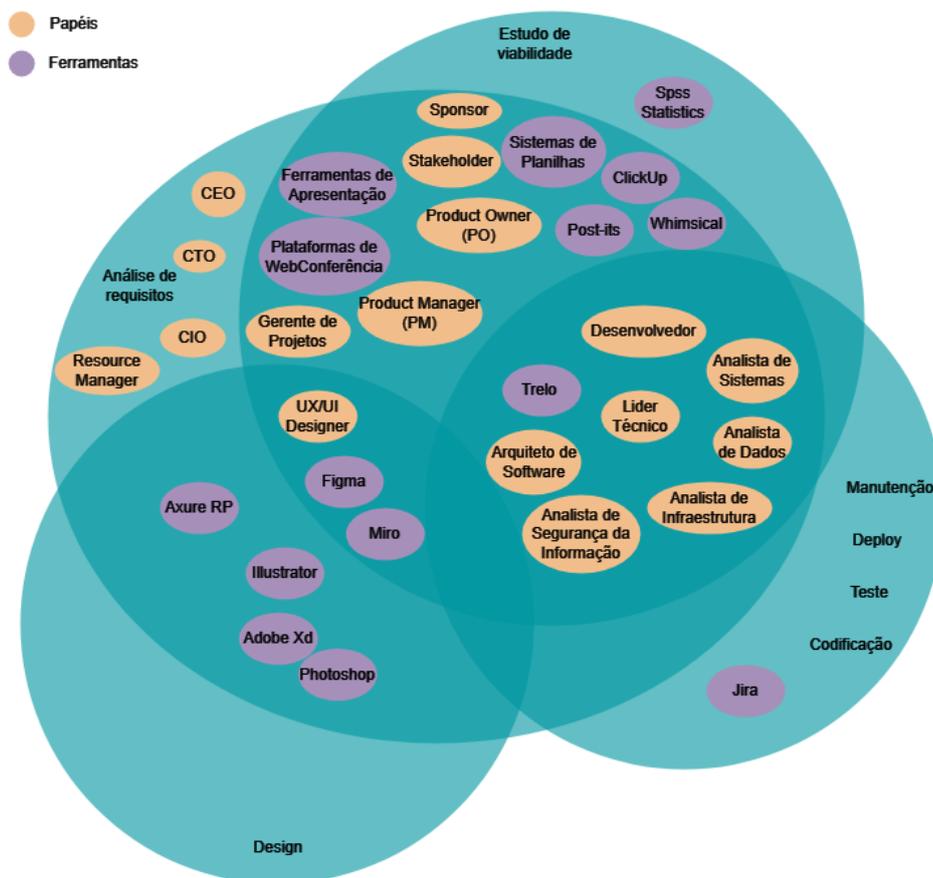
Figura 28 – Desafios Relacionadas a Outras Questões de Pesquisa

Desafio	[ID] Prática Associada	[ID] Papel Associado	[ID] Modelo Associado	[ID] Ferramenta Associada
C1. Engajamento do cliente e valor do produto	[6,7,8,10,11,12,14,15,41,46,66,69,105,110]	[2,3,4,5,6,7,8,11,12,22,30,32,33,34,36,38,39,42,50,57,60]	[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41]	[4,9,10,14,15,17,25,26,27,28]
C2. Desafios geográficos	[10,22,29,30,34,48,51,60,85]	[10,12,19,31,48,49,50,56,57]	[1,2,4]	[1,2,3,11,12,13,35,36,37,38,39,40,41,42,44,45,46,47,48,49,52,56,60,64,65,66,74,75]
C3. Desafios econômicos	[1,68, 91]	[1,14,15,16,31,48,57]	[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41]	[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75]
C4. Desafios socioculturais	[6,7,15,19,36,60,78,112,114]	[10,12,19,31,48,49,50,56,57]	[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41]	[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75]
C5. Controle de falhas e defeitos	[11,13,16,21,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,55,57,68,69,81,82]	[1,2,3,4,5,6,15,16,21,44, 53,54,55,60]	[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41]	[1,2,3,11,12,13,35,36,37,38,39,40,41,42,44,45,46,47,48,49,52,56,60,64,65,66,74,75]
C6. Monitorar falhas e defeitos	[11,13,16,21,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,55,57,68,69,81,82]	[1,2,3,4,5,6,15,16,21,44, 53,54,55,60]	[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41]	[1,2,3,11,12,13,35,36,37,38,39,40,41,42,44,45,46,47,48,49,52,56,60,64,65,66,74,75]
C7. Implementação de UCD no processo do produto.	[1,2,3,4,5,6,7,8,16,17,18,22,23,30,31,32,33,34,35,36,37,38,39,40,41,42,43,46,47,48,49,59]	[3,4,5,6,7,8,11,15,19,20,21,22,23,30,34,38,39,40,41,42,43,45,60]	[2,3,7,11,22,30,31]	[1,2,3,11,12,13,35,36,37,38,39,40,41,42,44,45,46,47,48,49,52,56,60,64,65,66,74,75]
C8. Identificar hábitos/interesses de consumo do usuário	[1,6,7,10,12,25,26,36,37,38,39,40,41,54,55,66,74,75,92,93,108,110,111]	[3,4,5,6,7,8,11,15,19,20,21,22,23,30,34,38,39,40,41,42,43,45,60]	[2,3,7,11,22,30,31]	[1,2,3,4,9,10,14,15,17,25,26,27,28,11,12,13,35,36,37,38,39,40,41,42,44,45,46,47,48,49,52,56,60,64,65,66,74,75]
C9. Incerteza nos processos	[8,10,30,31,32,33,34,35,36,37,48,112,113,114]	[2,8,12,14,15,16,24,31,42,48,49,50]	[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41]	[1,2,3,4,9,10,14,15,17,25,26,27,28,11,12,13,35,36,37,38,39,40,41,42,44,45,46,47,48,49,52,56,60,64,65,66,74,75]
C10. Implementar UCD com outros métodos.	[6,7,14,15,27,28,29,30,31,32,33,34,35,37,38,39,46,47,48,53,54,55,58,59,60,61,62,63,64,65,66,74,75,76,85,110,111,112,113]	[3,4,5,6,7,8,11,15,19,20,21,22,23,30,34,38,39,40,41,42,43,45,60]	[2,3,7,11,22,30,31]	[1,2,3,4,9,10,14,15,17,25,26,27,28,11,12,13,35,36,37,38,39,40,41,42,44,45,46,47,48,49,52,56,60,64,65,66,74,75]

o Design Thinking em seus processos, em relação ao que foi disponibilizado como opções de resposta no formulário da pesquisa.

A análise das respostas dos participantes do Survey revelou como os diferentes papéis utilizam as ferramentas associadas ao Design Thinking e em quais fases do processo de desenvolvimento de software essas ferramentas são mais comumente aplicadas. Essa abordagem permite compreender a adoção do Design Thinking em várias etapas do processo de desenvolvimento e destaca inovações no uso das ferramentas em diferentes cenários da indústria de software. A Figura 29 apresenta uma classificação das fases do desenvolvimento de software, que incluem: Análise de Requisitos, Estudo de Viabilidade, Design, Codificação, Teste, Deploy e Manutenção. Para cada fase, foram identificadas as ferramentas comumente utilizadas, bem como os papéis responsáveis por seu uso ao longo do ciclo de produção de software.

Figura 29 – Ferramentas Utilizadas por Diferentes Papéis no Desenvolvimento de Software com Design Thinking



Em resumo, os papéis associados ao desenvolvimento de software, conforme destacados na Figura 29, cobrem uma ampla gama de responsabilidades que vão desde a liderança estratégica até a execução técnica. A utilização de ferramentas eficientes pelos profissionais de TI é fundamental para superar os desafios enfrentados em cada fase do ciclo de vida

do software. A abordagem centrada no usuário, combinada com o gerenciamento eficaz de recursos, ferramentas e atuação dos papéis garante que o software atenda às necessidades de negócio e dos usuários, ao mesmo tempo em que mantém uma base técnica sólida e escalável. Isso evidencia a importância de uma equipe multidisciplinar utilizando um conjunto de ferramentas para cada etapa do processo de software e integrando metodologias como o Design Thinking com o desenvolvimento de software para viabilizar o sucesso do projeto.

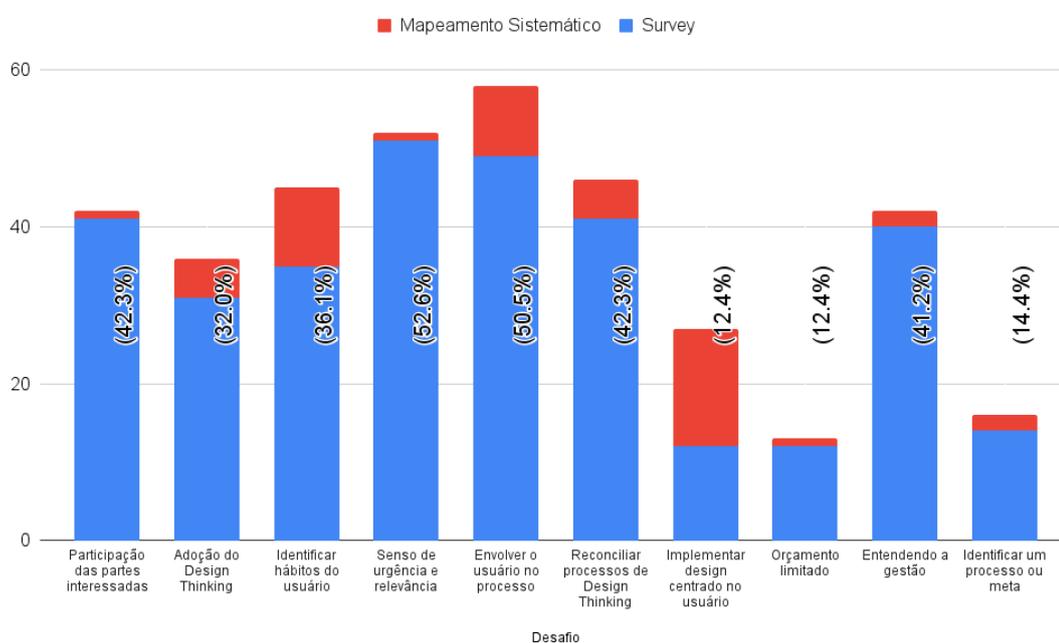
5.3 PADRÕES DE EVIDÊNCIAS

Com a execução dos estudos experimentais, podê-se obter um conjunto de informações que aparecem de forma recorrentes nos resultados obtidos associados ao Mapeamento Sistemático e ao Survey. A análise destas informações permite identificar tendências na utilização do Design Thinking no desenvolvimento de software. Essas evidências podem fornecer insights para tomada de decisão entre profissionais ou pesquisadores interessados na área desta dissertação, onde é possível avaliar as principais características associadas a adoção do DT no contexto da produção de software. O processo de desenvolvimento de software é complexo e envolve uma série de desafios técnicos e organizacionais, abrangendo desde a fase de estruturação e planejamento do projeto até sua conclusão. Para obter os resultados esperados dentro do projeto e trazer as perspectivas dos usuários para dentro do processo é necessário o auxílio de ferramentas, modelos e práticas para que possam cumprir este objetivo de maneira eficiente, angariando assim características de inovação para que possam romper os seus desafios, e conseqüentemente realizar entregas que atendam as necessidades dos usuários na utilização do software.

Neste trabalho, os desafios encontrados tiveram uma forte relação com o processo de desenvolvimento de software. De maneira geral, os desafios encontrados no Mapeamento Sistemático e no Survey apontam dificuldades em envolver as partes interessadas (Stakeholders), além da implantação da abordagem UCD (User Center Design) que consiste em trazer o usuário para o centro do processo de software, isto pode estar fortemente associado a um dos principais desafios informados pelos participantes do survey que é definir o senso de urgência e relevância das atividades do projeto, isto porque ter uma etapa de levantamento de requisitos mal direcionada pode trazer sérios danos ao produto, entretanto, a utilização dos princípios do Design Thinking surge como alternativa para esta necessidade (CARROLL, 2016). A utilização dos conceitos de Design Thinking para a estruturação dos processos associados a etapa de definição de requisitos e priorização de tarefas pode se tornar uma alternativa para definir modelos automatizados que auxiliem as inconsistências da priorização das tarefas em projetos de desenvolvimento de software. A Figura 30 apresenta os principais desafios encontrados durante a execução dos experimentos. Observa-se que há uma maior concentração de dados do Survey na referida figura, o que se deve ao fato de o percentual de citações associadas ao Mapeamento Sistemático ter sido calculado com base no número de

artigos selecionados que abordam os desafios apresentados.

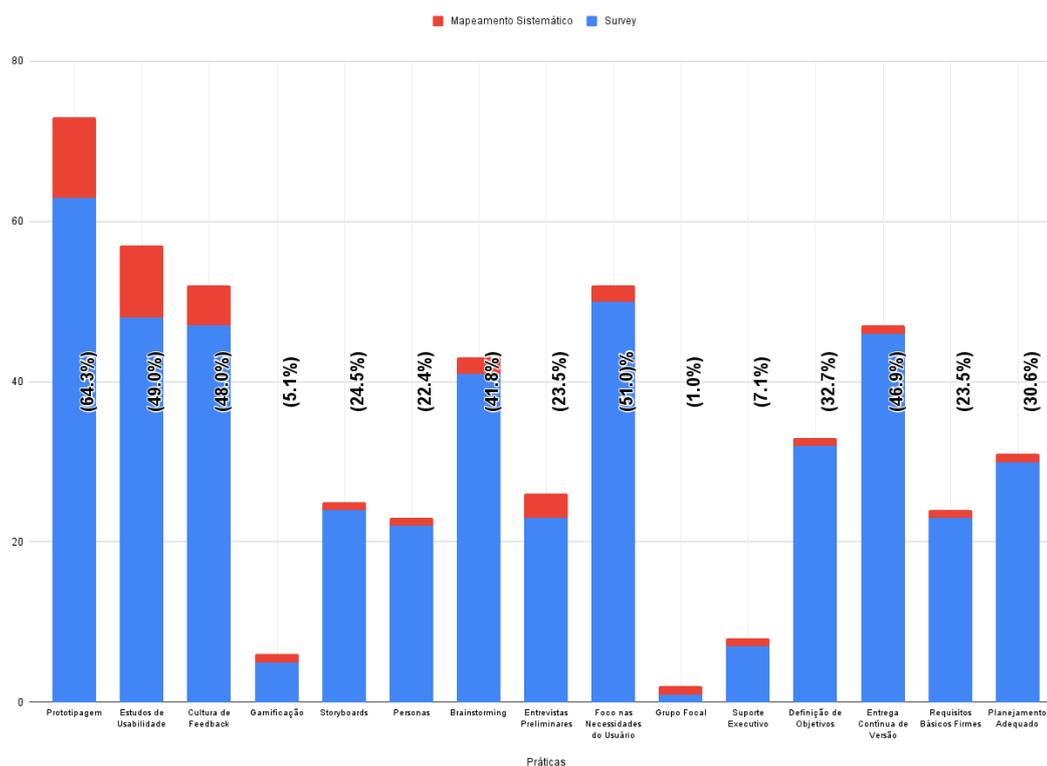
Figura 30 – Desafios - Citações no Survey e Mapeamento Sistemático



Em relação às práticas observadas nos estudos experimentais, a prototipação e os estudos de usabilidade se destacaram de forma significativa em ambos os experimentos. Estudos apontam que, na maior parte dos projetos de software, problemas como usabilidade não levam o software ao sucesso. Em contrapartida, abordagem de design centrada no usuário podem oferecer suporte ao processo de desenvolvimento de software para lidar com problemas de usabilidade (INAL, 2020; ANWAR, 2014). A indústria de software tem buscado aplicar práticas que forneçam uma experiência do usuário (UX) mais significativa e ajudem a manter um negócio sustentável (MARTINELLI, 2022). Os resultados desta dissertação confirmam que há uma crescente busca para encontrar novas práticas e processos associadas ao Design Thinking dentro das empresas, sendo a pesquisa de usuário o caminho mais evidente entre os resultados dos experimentos. A Figura 31 apresenta um comparativo da utilização das práticas associadas ao Design Thinking no desenvolvimento de software, conforme os achados dos experimentos. As informações referentes ao Survey correspondem às respostas do questionário de pesquisa, enquanto os dados relativos ao Mapeamento Sistemático foram gerados com base nos artigos que mencionam essas práticas.

Os resultados do Survey indicam que o modelo Ágil é o mais destacado em cenários de desenvolvimento de software que integram o Design Thinking. No entanto, os achados do Mapeamento Sistemático revelaram uma diversidade de modelos utilizados em conjunto com os mecanismos do Design Thinking. Essa variação abrange desde modelos tradicionais, como o modelo Cascata e o Stanford Design, até modelos híbridos, como o Scrumban, que combina abordagens Scrum com conceitos do Kanban. A Figura 32 compara as respostas

Figura 31 – Práticas - Citações no Survey e Mapeamento Sistemático



associadas aos modelos utilizados no processo de desenvolvimento de software pelos participantes do Survey com os achados do Mapeamento Sistemático. Esta abordagem confirma a predominância do modelo Ágil em ambos os experimentos. Cabe destacar que o gráfico não apresenta o modelo Design Thinking, uma vez que o objetivo, na estruturação do questionário Survey, foi identificar quais modelos podem ser utilizados em conjunto com o Design Thinking. Portanto, essa perspectiva do gráfico se baseia nessa abordagem.

Com o aumento exponencial no uso de serviços e aplicações baseados na internet, há também uma crescente demanda por web designers e desenvolvedores (MACHADO, 2021). Os dados relativos aos papéis que compõem as equipes de tecnologia em ambos os experimentos reforçam essa afirmação. Nesse contexto, a Figura 33 apresenta o nível de adoção do Design Thinking entre os diferentes papéis envolvidos no desenvolvimento de software. Essa abordagem visa identificar o grau de conhecimento e aplicação do Design Thinking em diversos segmentos do processo de software, levando em consideração os papéis que podem ser integrados nas fases de planejamento estratégico, abrangendo desde os papéis gerenciais (Gerente de Projetos, Líder Técnico, CEO) até os papéis de execução (Desenvolvedores, Analistas de Sistemas, Arquitetos de Software, entre outros). A Figura 33 foi gerada com base nas respostas dos participantes do Survey, e as definições dos papéis foram elaboradas conforme a maior frequência de citações entre os artigos selecionados no Mapeamento Sistemático.

Figura 32 – Modelos Integrados ao Design Thinking no Desenvolvimento de Software

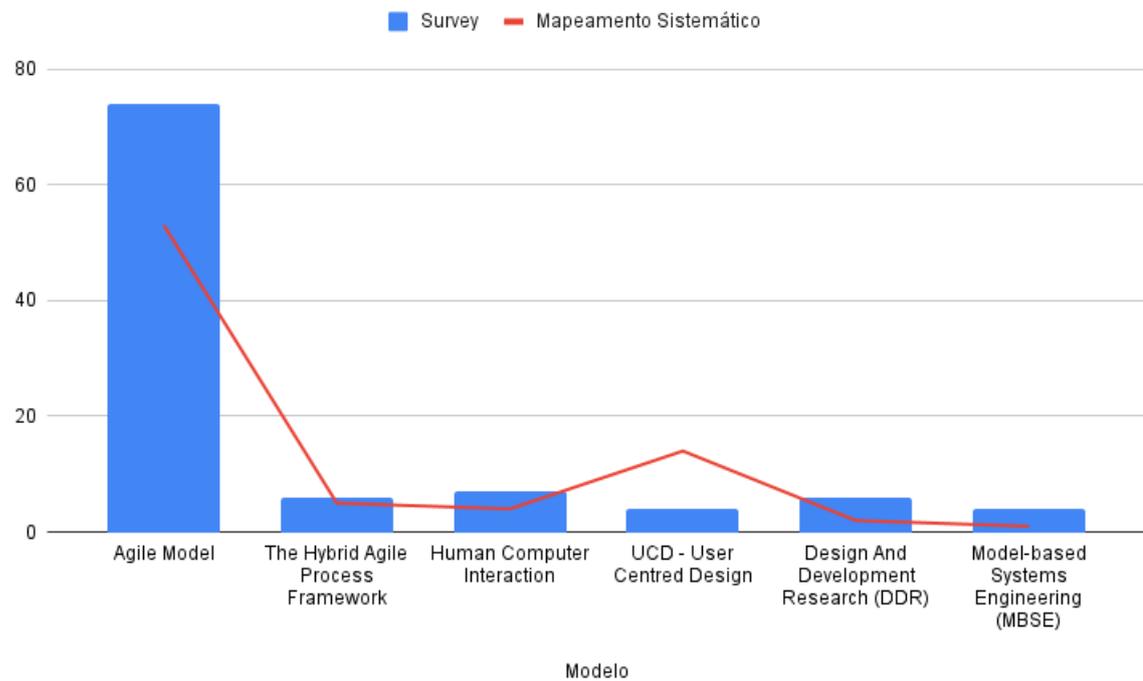
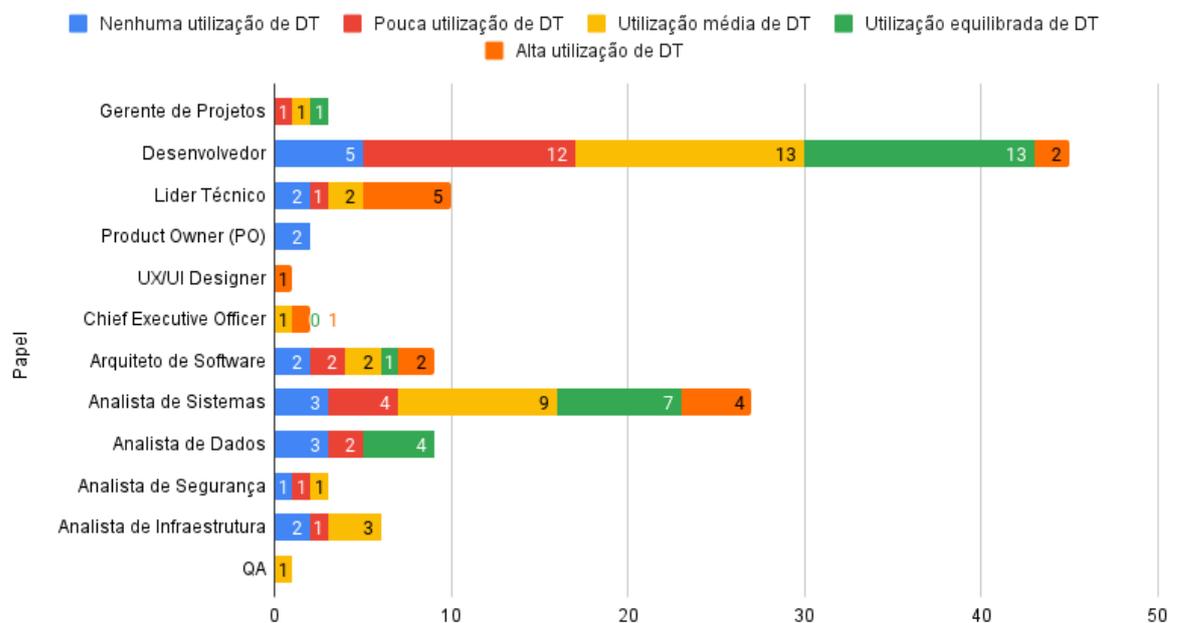


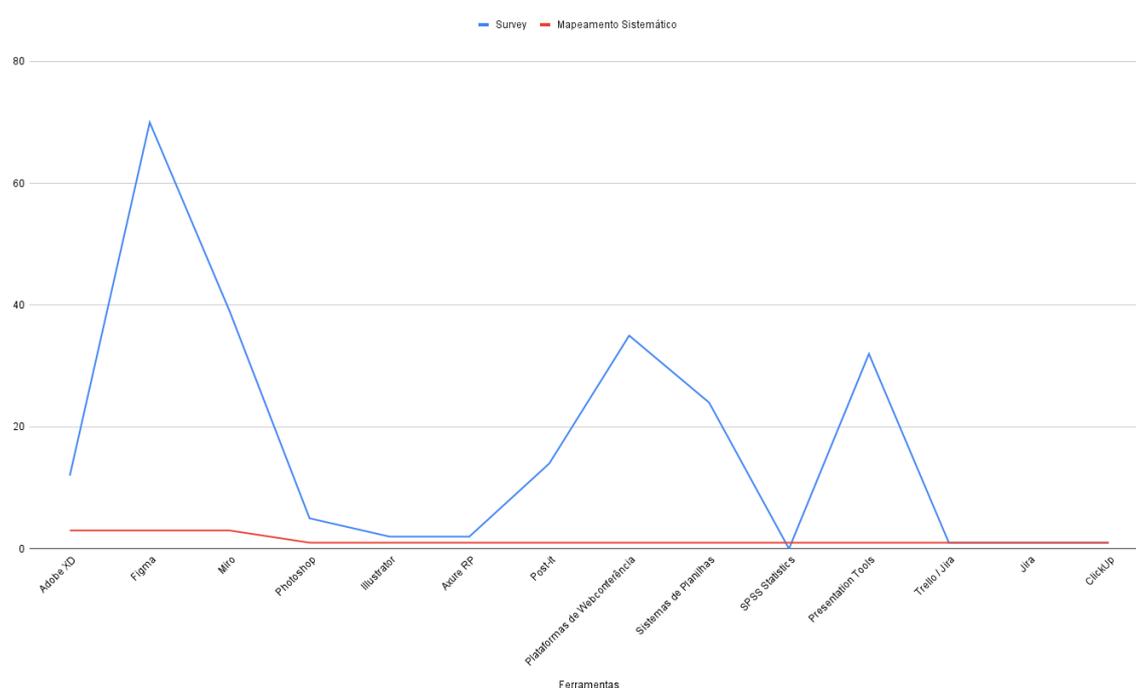
Figura 33 – Design Thinking e Sua Adoção em Diversos Papéis



As ferramentas derivadas destes estudos apresentam grande similaridade nos resultados, evidenciando uma forte utilização de ferramentas de prototipagem e ideação, como Figma, Miro e Adobe XD. A Figura 34 apresenta um comparativo da utilização dessas

ferramentas entre os achados dos experimentos. Essa análise foi baseada nas respostas dos participantes do Survey e nos estudos primários que apontam as ferramentas mencionadas no gráfico. No Mapeamento Sistemático, foram identificadas 75 ferramentas a partir dos artigos selecionados, que estão disponíveis no Apêndice F deste documento. No entanto, o gráfico da Figura 34 foi elaborado exclusivamente com base nas ferramentas mencionadas no Survey, sendo realizado o comparativo entre os estudos.

Figura 34 – Ferramentas Integradas ao Design Thinking no Desenvolvimento de Software



A realização conjunta do Mapeamento Sistemático e do *Survey* permitiu uma análise abrangente sobre o uso do Design Thinking no desenvolvimento de software. Enquanto o mapeamento sistemático ofereceu uma visão consolidada da literatura, identificando recursos de DT no processo de desenvolvimento de software, o *Survey* complementou essa perspectiva ao captar insights práticos e percepções diretamente das empresas. Os resultados convergiram ao destacar a crescente adoção do Design Thinking como um recurso eficaz para promover inovação, colaboração e foco nas necessidades dos usuários no contexto do desenvolvimento de software. Essa convergência reforça a relevância da prática, tanto em termos teóricos quanto práticos, como um diferencial estratégico para adoção do DT em cenários reais de desenvolvimento de software.

5.4 LIMITAÇÕES GERAIS DESTA PESQUISA

A execução do Mapeamento Sistemático enfrentou algumas limitações, por exemplo, considerou apenas os artigos publicados a partir do ano 2000, mas o conceito de DT surgiu nos anos 90, muitos artigos não responderam a todas as questões de pesquisa, alguns artigos duplicados entre as bases de dados utilizadas e, finalmente, a maioria dos dados estavam associados apenas ao desenvolvimento de software. Vários trabalhos devem ser desenvolvidos para refinar e testar este corpo de evidências e outros que podem ser derivados como um novo tipo de modelo. Finalmente, os resultados deste estudo são limitados a uma abordagem teórica por meio de evidências encontradas nos estudos primários selecionados durante a execução deste mapeamento sistemático. Em geral, os dados aqui apresentados podem promover direções de pesquisa com base em resultados encontrados por outros autores ao adotar o Design Thinking em projetos de desenvolvimento de software.

Referente a execução do Survey, observou-se que muitos profissionais de tecnologia da informação ainda possuem familiaridade limitada com técnicas de Design Thinking. Em segundo lugar, embora a pesquisa tenha um escopo internacional, a maioria dos participantes atua em empresas brasileiras, o que pode restringir a generalização dos resultados para outros contextos organizacionais.

6 TRABALHOS RELACIONADOS

Esta seção apresenta quatro estudos que foram publicados nos últimos 5 anos, que apresentam dados importantes sobre a utilização do Design Thinking dentro dos processos de desenvolvimento de software. A pesquisa “Design thinking in practice: understanding manifestations of design thinking in software engineering” (DOBRIGKEIT, 2019), destaca a importância do Design Thinking (DT) como ferramenta de apoio à equipe de desenvolvimento, pois facilita a assimilação das necessidades e requisitos do usuário. Como estudo de caso, os autores trabalharam com três equipes de uma empresa global de software, fazendo uso de extensas observações e conduzindo entrevistas semiestruturadas com cada equipe. Os autores afirmam que novos conhecimentos significativos chegam à literatura emergente sobre como harmonizar o processo de DT e o desenvolvimento de software. O estudo também discute que a DT pode ser descrita em três níveis de manifestação, a saber, como uma mentalidade, como um processo ou como uma caixa de ferramentas. Este trabalho tenta responder a duas questões de pesquisa: (1) como as equipes de software da indústria aplicam DT em seus projetos e (2) como os membros da equipe estão envolvidos com o uso de DT. Esta pesquisa de mestrado possibilitou identificar as principais evidências do uso do DT em cenários reais de desenvolvimento de software, sendo prioritariamente encontradas por meio de evidências no estado da arte, por meio do mapeamento sistemático descrito nas seções anteriores, e da aplicação do survey focado em respostas de profissionais de TI de diferentes empresas. Isto difere-se do estudo de DOBRIGKEIT (2019) que limitou-se a aplicação formulários dentro de um contexto corporativo.

O estudo “Integrating design thinking in extreme Programming” (SOHAIB, 2019) aborda a integração de práticas de DT em Extreme Programming a fim de obter a qualificação do produto de software para usuários e o desenvolvimento de inovações em processos de metodologia ágil. Os dados coletados neste trabalho foram coletados em entrevistas semiestruturadas e abertas com dez participantes em duas empresas de software em Sydney. Por fim, o estudo apresentou uma estrutura integrada baseada nos conceitos de Design Thinking e Extreme Programming, contribuindo para esse conhecimento da pesquisa de design thinking em software, porém restrita a esse método de desenvolvimento de software.

O estudo “Alignment of stakeholder expectations about user involvement in agile software development” (BUCHAN, 2017) direcionou o estudo para o papel do envolvimento do usuário no desenvolvimento de software. Os autores realizaram um estudo de caso exploratório de expectativas sobre o envolvimento do usuário no desenvolvimento ágil. Dados qualitativos foram coletados por meio de entrevistas para projetar um novo método para avaliar o alinhamento das expectativas sobre o envolvimento do usuário por meio da aplicação de *Repertory Grids*. O estudo em questão avalia se a interação com o usuário durante

a estruturação do projeto é benéfica para os processos. No entanto, este artigo envolve apenas evidências de envolvimento do usuário, não incluindo evidências envolvendo diretamente a equipe de desenvolvimento.

O estudo “Design Thinking Integrated in Agile Software Development: A Systematic Literature Review” (PEREIRA, 2018) apresenta uma revisão sobre como a abordagem do Design Thinking é usada integrada às metodologias do Desenvolvimento Ágil de Software. Os autores realizaram uma Revisão Sistemática da Literatura que comprovou resultados benéficos do envolvimento do usuário no processo de desenvolvimento de software. O estudo teve 29 artigos relacionados a este tópico. Os resultados mostraram que a maioria dos modelos integrados ao processo de desenvolvimento são aplicados ao longo do ciclo de vida do software, e o Scrum como metodologia utilizada com mais frequência. Em geral, este trabalho trata de como o design thinking e o processo de desenvolvimento de software são abordados na literatura, apresentando um conjunto de desafios e modelos neste contexto. Assim, esse conjunto de dados pode facilitar a adoção prática de DT no cenário de desenvolvimento de software.

Este estudo “Definition of indicators in the execution of educational projects with design thinking using the systematic literature review” (ALMEIDA, 2019), investiga o uso de métricas e indicadores para avaliar a eficácia do Design Thinking (DT) em projetos educacionais. Os autores conduziram uma revisão sistemática da literatura e a execução de um survey para determinar a compreensão conceitual do DT e seus indicadores associados entre profissionais de TI e estudantes. Os resultados destacam a dificuldade em definir indicadores específicos para projetos de DT devido à falta de conhecimento e experiência com a metodologia. O artigo conclui que mais pesquisas são necessárias para desenvolver uma compreensão mais profunda do DT e suas aplicações, particularmente em ambientes acadêmicos e industriais, para estabelecer indicadores robustos e aplicáveis para medir o desempenho. No entanto, suas descobertas são afetadas por um pequeno tamanho de amostra, potencial viés no design do questionário e uma interpretação subjetiva dos indicadores. A falta de uma estrutura abrangente para definir e aplicar indicadores, juntamente com um foco na compreensão geral em vez de aplicações específicas, restringe o valor prático do artigo. Mais pesquisas são necessárias para abordar essas limitações e desenvolver uma metodologia mais robusta para usar indicadores em projetos de DT.

A principal distinção entre este artigo e trabalhos relacionados está em sua ênfase e abordagem metodológica. Os quatro primeiros estudos focam em estruturas teóricas e análises de caso que investigam aspectos específicos do Design Thinking (DT), incluindo estratégias de integração, métricas de eficácia, manifestações em contextos da indústria e o refinamento de processos de elicitação de requisitos. Esses estudos utilizam predominantemente revisões sistemáticas de literatura, estudos de caso e análises qualitativas para destacar os desafios e limitações de suas descobertas. Em contraste, esta pesquisa

visa identificar padrões de evidências da utilização do Design Thinking em cenários de desenvolvimento de software do mundo real, enfatizando os desafios encontrados, as ferramentas utilizadas e as melhores práticas implementadas pelos profissionais. Enquanto os estudos anteriores exploram principalmente construções teóricas e propõem estruturas para aprimorar a compreensão do DT, este estudo fornece insights acionáveis enraizados nas experiências de profissionais da indústria. Esses insights reforçam a aplicação prática do Design Thinking dentro da indústria de software. Além disso, este estudo oferece evidências convincentes que apoiam a adoção de conceitos projetados para atender às necessidades do usuário em contextos do mundo real, como por exemplo um conjunto de práticas e ferramentas que podem ser utilizadas ao adotar o Design Thinking no desenvolvimento de software. Alinhando-se com as descobertas apresentadas na literatura existente, as evidências geradas por esta pesquisa permitem que acadêmicos e profissionais avaliem descobertas relevantes em ambientes reais de desenvolvimento de software e adaptem esses insights às suas circunstâncias específicas.

A Tabela 15 apresenta uma comparação entre a quantidade de evidências encontradas em trabalhos relacionados e as propostas deste estudo. Cada trabalho relacionado é classificado de acordo com a quantidade de evidências do uso do Design Thinking no desenvolvimento de software, considerando modelos, práticas, papéis, ferramentas e desafios. A Tabela 16 apresenta os resultados detalhados de cada trabalho relacionado. De modo geral, nota-se uma escassez de evidências nos trabalhos mencionados sobre como o Design Thinking pode ser utilizado no processo de desenvolvimento de software, bem como os benefícios de sua adoção. Este trabalho de mestrado busca preencher essa lacuna, fornecendo um conjunto de padrões de evidências que auxiliam no processo de adoção do Design Thinking em cenários de produção de software. Em virtude da limitação de espaço na Tabela 16, as informações associadas aos experimentos desta dissertação não foram incorporadas, uma vez que os dados referentes a 116 práticas, 75 ferramentas, 60 papéis, 47 modelos e 28 desafios tornariam a tabela ilegível nesse formato. Como alternativa, os dados relacionados a este estudo podem ser consultados nos apêndices deste documento.

Tabela 15 – Evidências do Design Thinking nos Trabalhos Relacionados

Quantitativo de evidências						
Autor	Modelos	Práticas	Papéis	Ferramentas	Desafios	Tipo
Dobrigkeit and Paula	8	0	0	2	0	CS
Sohaib et al.	3	23	7	1	1	CS
Pereira and Russo	12	10	2	0	3	SLR
Almeida et al.	1	0	10	0	0	SLR
Este estudo	47	116	60	75	28	SM+SV

Legenda: SLR - Systematic Literature Review / SM - Systematic Mapping / CS - Case Study / FG - Focus Group / SV - Survey

Autor	Modelos	Práticas	Papéis	Ferramentas	Desafios
Dobrigkeit and Paula	Brown, d.School Stanford, Wolbling, Brenner et al., Split project model, Overlapping teams model, Unified project model, Toolbox model	-	-	Workshop, Design Thinking	-
Sohaib et al.	Extreme Programming, Design Thinking, Agile	Planning game, Small releases, Metaphor, Simple design, Tests, Refactoring, Pair programming, Continuous integration, Collective ownership, On-site customer, 40 hours' week, Open workspace and just rules, User profiling, Collaborative ideation, Prototyping, User feedback, Usability, User stories, Spike solution, Incremental design, Acceptance test, Working software	UX/UI - Designer, Developer, Software Engineers, Agile Coaches, Business Analysts, Software Tester	NVIVO	Lack of usable project
Pereira and Russo	IDEO, Stanford, ISO, Dschool, Scaled Agile Framework (SAFe), Design Thinking, Agile, Extreme programming (XP), Scrum, Feature driven development (FDD), Crystal methods, Dynamic systems development methods (DSDM), Adaptive software development (ASD), Lean development, Model-driven development (AMDD)	Empathize, define, ideate, prototype, test, Define strategy (understand; define), execute solution, Inspiration, specify, sketch	Developer, Customers	-	Geographic constraints, Narrow project schedule, Short budget
Almeida et al.	IDEO	-	Project Manager (PM), Project Administrator (PA), Software Engineer (SE), Systems Analyst (SA), Software Tester (ST), Programmer/Developer (PD), Specialist in Human-Computer Interaction (HCIS), Designer (interaction designer, UX designer)(D), Researcher/teacher (RT), User (informant, tester, design partner)	-	-

Tabela 16 – Evidências do Uso de Design Thinking nos Trabalhos Relacionados

7 CONSIDERAÇÕES FINAIS

A indústria de software precisa de produtos inovadores que atendam cada vez mais as perspectivas dos usuários, com boa usabilidade e funcionalidades que atendam às suas reais demandas. Nesse sentido, o conceito de Design Thinking tem se consolidado como um poderoso recurso para desenvolvimento de software, trazendo vantagens competitivas e impactando diretamente na qualidade do produto. O Design Thinking se destaca como uma metodologia que aborda esses desafios priorizando usuários e clientes. Este conceito é definido por sua ênfase na colaboração multidisciplinar e processos tangíveis, orientando, em última análise, as organizações em direção a soluções de negócios inovadoras e viáveis. Essa abordagem estratégica ao design reflete uma perspectiva holística voltada para a inovação, alavancando princípios de design para atender às necessidades do usuário por meios tecnológicos. O objetivo é criar soluções que não sejam apenas tecnicamente viáveis, mas também estrategicamente alinhadas com os objetivos de negócios, entregando valor aos clientes. Conforme mencionado anteriormente, a adoção do Design Thinking no desenvolvimento de software produz benefícios significativos, incluindo uma compreensão mais clara dos requisitos, identificação precoce de erros por meio de prototipagem e implementação mais suave facilitada pela contribuição do cliente em protótipos.

A realização deste estudo teve como principal objetivo identificar padrões de evidências relacionados à adoção do Design Thinking em cenários de desenvolvimento de software, por meio da análise de práticas, modelos, papéis, desafios e ferramentas nesses contextos. Esses conjuntos de evidências podem auxiliar profissionais e pesquisadores da área de desenvolvimento de software a compreender melhor os desafios enfrentados e a implementar soluções mais eficazes, utilizando abordagens de Design Thinking, resultando em benefícios diretos e relevantes.

A metodologia deste estudo iniciou-se com a busca por padrões de evidências presentes na literatura e no setor industrial de produção de software, que enfatizam o uso de DT no processo de desenvolvimento de software por meio de estudos experimentais, como Mapeamento Sistemático e Survey. Durante a execução dos experimentos, foram seguidas as diretrizes de KITCHENHAM (2010), que oferecem subsídios para avaliar criticamente e agregar todos os artigos de pesquisa relevantes (denominados estudos primários) sobre uma questão ou tópico específico de pesquisa. Para o Survey, foram adotadas as diretrizes de pesquisa propostas por WAGNER et al. (2020), contemplando as seguintes etapas: desenho inicial da pesquisa, revisão do desenho da pesquisa, avaliação preliminar da estruturação do estudo e, por fim, o refinamento do conteúdo da pesquisa.

As perguntas do mapeamento sistemático buscavam identificar e analisar quais desafios, funções, práticas, modelos e ferramentas do Design Thinking são usados no desenvolvimento de software. Este mapeamento sistemático utilizou 3 bases científicas (IEEE, ACM e SCOPUS) para buscas compreendendo pesquisas publicadas no período de 2000 a 2023, resultando em um total de 95 artigos selecionados após a sua execução. Esses resultados foram identificados, compilados e analisados, e então algumas discussões e análises mais profundas foram realizadas. De forma sequencial, a partir da execução do survey foi examinado a adoção do Design Thinking em contextos de desenvolvimento de software em indústrias de software. Consistindo em 18 perguntas direcionadas, a pesquisa foi distribuída para vários locais específicos frequentados por profissionais de TI. Para garantir amplo alcance, a pesquisa foi disseminada por meio de listas de e-mail em universidades de tecnologia, bem como, por meio de grupos de desenvolvimento e design de software em plataformas de mídia social e diretamente para profissionais conhecidos do setor. A pesquisa permaneceu aberta para respostas até agosto de 2024, atraindo a participação de 105 profissionais de TI. Por meio deste estudo, pretendemos aprofundar o entendimento de como o Design Thinking é integrado às práticas de desenvolvimento de software e identificar os benefícios e desafios associados enfrentados pelos profissionais da área.

Com base nas respostas recebidas, diversas observações e análises foram realizadas. Inicialmente, foi conduzida uma análise descritiva, com foco em fatores como o porte da empresa, o nível educacional e a experiência profissional dos participantes. Em seguida, foram realizadas análises de tabulação cruzada e correlações, investigando, por exemplo, a relação entre as funções dos participantes e seu nível de conhecimento, bem como a conexão entre o porte da empresa e o grau de adoção do Design Thinking. Além disso, as análises exploraram a correlação entre conhecimento e adoção, bem como o impacto da experiência profissional na implementação de práticas. Adicionalmente, análises qualitativas e comparativas examinaram os desafios enfrentados e a busca por novas práticas, além de avaliar metodologias combinadas e o uso de ferramentas em diferentes tipos de projetos.

Este corpo de evidências cobre uma literatura atual sobre DT e esses resultados podem ajudar profissionais e pesquisadores da área de desenvolvimento de software para entender melhor os desafios e implementar soluções mais eficazes para melhorar suas tarefas usando abordagens de design thinking. Esses resultados também fornecem um mapeamento elementar sobre padrões de design thinking em desenvolvimento de software, identificando áreas onde mais pesquisas precisam ser realizadas. Em virtude disto, podem ajudar profissionais e pesquisadores da área a entender melhor os desafios e implementar soluções mais eficazes para melhorar suas tarefas no desenvolvimento de software usando abordagens de design thinking. Esses resultados também fornecem um mapeamento simples da pesquisa sobre design thinking no desenvolvimento de software, identificando áreas onde mais pesquisas são necessárias.

Como pesquisa futura, os próximos trabalhos podem explorar várias vias para aprimorar ainda mais o entendimento neste contexto. Uma área potencial de foco pode ser o impacto longitudinal do Design Thinking nos resultados do projeto, avaliando como sua implementação influencia o sucesso do produto e a satisfação do usuário ao longo do tempo. Além disso, a pesquisa pode examinar a integração do Design Thinking com outras metodologias contemporâneas, como Agile ou DevOps, para identificar as melhores práticas que fomentam a inovação e a eficiência.

8 REFERÊNCIAS BIBLIOGRÁFICAS

PARIZI, Rafael et al. How has design thinking being used and integrated into software development activities? A systematic mapping. *Journal of Systems and Software*, p. 111217, 2022.

KITCHENHAM, Barbara et al. Guidelines for performing systematic literature reviews in software engineering. Technical report, ver. 2.3 ebse technical report. ebse, 2007.

PEREIRA, Julio Cesar; DE FSM RUSSO, Rosaria. Design thinking integrated in agile software development: A systematic literature review. *Procedia computer science*, v. 138, p. 775-782, 2018.

SENF, Björn et al. A value-centered approach for unique and novel software applications. In: *Design, User Experience, and Usability. Practice and Case Studies: 8th International Conference, DUXU 2019, Held as Part of the 21st HCI International Conference, HCII 2019, Orlando, FL, USA, July 26–31, 2019, Proceedings, Part IV 21*. Springer International Publishing, 2019. p. 366-384.

KIMBELL, Lucy. Rethinking design thinking: Part I. *Design and culture*, v. 3, n. 3, p. 285-306, 2011.

LUPTON, E. D. T.; LUPTON, Ellen. *Graphic design thinking: Beyond brainstorming*. Princeton Architectural Press, 2011.

TSCHIMMEL, Katja. Design Thinking as an effective Toolkit for Innovation. In: *ISPIM Conference Proceedings. The International Society for Professional Innovation Management (ISPIM)*, 2012. p. 1.

SOHAIB, Osama et al. Integrating design thinking into extreme programming. *Journal of Ambient Intelligence and Humanized Computing*, v. 10, p. 2485-2492, 2019.

PRESTES, Matheus et al. On the use of design thinking: A survey of the Brazilian agile software development community. In: *Agile Processes in Software Engineering and Extreme Programming: 21st International Conference on Agile Software Development, XP 2020, Copenhagen, Denmark, June 8–12, 2020, Proceedings 21*. Springer International Publishing, 2020. p. 73-86.

DOBRIGKEIT, Franziska; DE PAULA, Danielly. Design thinking in practice: understanding manifestations of design thinking in software engineering. In: *Proceedings of the 2019 27th ACM joint meeting on European software engineering conference and symposium on the foundations of software engineering*. 2019. p. 1059-1069.

FISHER, Karen E.; FAWCETT, Phil. Using design thinking to empower ethnic minority

immigrant youth in their roles as information and technology mediaries. 2013.

FARRAR, Emily J. Implementing a design thinking project in a biomedical instrumentation course. *IEEE Transactions on Education*, v. 63, n. 4, p. 240-245, 2020.

UEBERNICKEL, Falk et al. *Design thinking: The handbook*. World Scientific, 2020.
PRZYBILLA, Leonard et al. Design thinking in digital innovation projects—Exploring the effects of intangibility. *IEEE Transactions on Engineering Management*, v. 69, n. 4, p. 1635-1649, 2020.

THORING, Katja; MÜLLER, Roland M. Understanding the creative mechanisms of design thinking: an evolutionary approach. In: *Proceedings of the Second Conference on Creativity and Innovation in Design*. 2011. p. 137-147.

VIEIRA, Renato Correa; DE JESUS RAFAEL, Guilherme; DA COSTA, Juliana AR. Design thinking experience. In: *Companion Proceedings of the 13th Brazilian Symposium on Human Factors in Computing Systems*. 2014. p. 1-4.

PETERSEN; et al. (2017). Development and Testing of a Mobile Application to Support Diabetes Self-Management for People with Newly Diagnosed Type 2 Diabetes: a Design Thinking Case Study. *BMC Medical Informatics and Decision Making*, 1-10.

JIMÉNEZ, Francys Marcel Rodríguez et al. Implementação da metodologia de design thinking no desenvolvimento de uma prótese robótica de membro superior. Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Mecânica da Universidade Federal de Minas Gerais, 2021.

S. DHANDAPANI, "Integration of User Centered Design and Software Development Process," 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 2016, pp. 1-5, doi: 10.1109/IEMCON.2016.7746075.

TRAVASSOS, G. H.; GUROV, D.; AMARAL, E. A. G. D. *Introdução à Engenharia de Software Experimental: Relatório Técnico*. COPPE/UFRJ, Rio de Janeiro. 2002.

D. TEKA, Y. DITTROCH and M. KIFLE, "Adapting Lightweight User-Centered Design with the Scrum-Based Development Process," 2018 IEEE/ACM Symposium on Software Engineering in Africa (SEiA), Gothenburg, Sweden, 2018, pp. 35-42.

GURAN, Adriana-Mihaela; COJOCAR, Grigoreta-Sofia; MOLDOVAN, Anamaria. Designing edutainment software for digital skills nurturing of preschoolers: A method proposal. In: *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering in Society*. 2020. p. 63-70.

BANO, Muneera; ZOWGHI, Didar. User involvement in software development and system success: a systematic literature review. In: *Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering*. 2013. p. 125-130.

BUCHAN, Jim et al. Alignment of stakeholder expectations about user involvement in agile software development. In: Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering. 2017. p. 334-343.

SFETSOS, Panagiotis et al. Integrating user-centered design practices into agile web development: A case study. In: 2016 7th International Conference on Information, Intelligence, Systems Applications (IISA). IEEE, 2016. p. 1-6.

PÉRAIRE, Cécile. Dual-track agile in software engineering education. In: 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET). IEEE, 2019. p. 38-49.

FOX, David; SILLITO, Jonathan; MAURER, Frank. Agile methods and user-centered design: How these two methodologies are being successfully integrated in industry. In: Agile 2008 Conference. IEEE, 2008. p. 63-72.

VAN DE VEN, Andrew H. et al. Engaged scholarship: A guide for organizational and social research. Oxford University Press on Demand, 2007.

PETERSEN, K. et al. Systematic mapping studies in software engineering. In: Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering. Swindon, UK: BCS Learning Development Ltd., 2008. (EASE'08), p. 68-77.

WOHLIN, Claes et al. Experimentation in software engineering. Springer Science Business Media, 2012.

PEREIRA, Lauriane et al. Towards an understanding of benefits and challenges in the use of design thinking in requirements engineering. In: Proceedings of the 36th Annual ACM Symposium on Applied Computing. 2021. p. 1338-1345.

CARROLL, Noel; RICHARDSON, Ita. Aligning healthcare innovation and software requirements through design thinking. In: Proceedings of the international workshop on software engineering in healthcare systems. 2016. p. 1-7.

MARRU, Suresh et al. User-Centric Design and Evolvable Architecture for Science Gateways: A Case Study. In: 2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid). IEEE, 2021. p. 267-276.

De, SUMAN. Minimal Satisfiable Product: A Novel Concept in Agile Project Management for better adherence to Customer Requirements. In: 2021 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON). IEEE, 2021. p. 1-6.

TEKAAT, Julian L.; ANACKER, Harald; DUMITRESCU, Roman. The paradigm of design thinking and systems engineering in the design of cyber-physical systems: A systematic literature review. In: 2021 IEEE International Symposium on Systems Engineering (ISSE). IEEE, 2021. p. 1-8.

DA SILVA, Tiago Silva et al. User-centered design and agile methods: a systematic review. In: 2011 AGILE conference. IEEE, 2011. p. 77-86.

BEKELE, Rahel et al. User-centered design in developing countries: a case study of a sustainable intercultural healthcare platform in Ethiopia. In: 2019 IEEE/ACM symposium on software engineering in Africa (SEiA). IEEE, 2019. p. 11-15.

MARRU, Suresh et al. User-Centric Design and Evolvable Architecture for Science Gateways: A Case Study. In: 2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid). IEEE, 2021. p. 267-276.

GURAN, Adriana-Mihaela; COJOCAR, Grigoreta-Sofia; MOLDOVAN, Anamaria. Designing edutainment software for digital skills nurturing of preschoolers: A method proposal. In: Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering in Society. 2020. p. 63-70.

MIRZA, Mahrukh Sameen; DATTA, Soma. Developing Software Using Agile and Design Thinking Framework. In: 2020 International Conference on Computational Science and Computational Intelligence (CSCI). IEEE, 2020. p. 1819-1823.

LIU, Simon. Engineering the success of software development. *IT Professional*, v. 15, n. 5, p. 4-5, 2013.

MAHÉ, Nolwen et al. Migrating a software factory to design thinking: Paying attention to people and mind-sets. *IEEE Software*, v. 37, n. 2, p. 32-40, 2019.

PETERS, Lawrence; MORENO, Ana M. Evaluating software project managers: A multidimensional perspective. *IEEE Software*, v. 34, n. 6, p. 104-108, 2017.

BROWN, T. Design thinking. *Harvard business review*, v. 86, n. 6, 2008, p. 84-92, 2008.

IBRAIGHEETH, Mohammad Ahmad; FADZLI, Syed Abdullah. Software project failures prediction using logistic regression modeling. In: 2020 2nd International Conference on Computer and Information Sciences (ICCIS). IEEE, 2020. p. 1-5.

DOBRIGKEIT, Franziska et al. Cherry picking-Agile software development teams applying design thinking tools. In: *Agile Processes in Software Engineering and Extreme Programming-Workshops: XP 2021 Workshops, Virtual Event, June 14-18, 2021, Revised Selected Papers 22*. Springer International Publishing, 2021. p. 201-206.

PALACIN-SILVA, Maria et al. Infusing design thinking into a software engineering capstone course. In: 2017 IEEE 30th Conference on Software Engineering Education and Training (CSEET). IEEE, 2017. p. 212-221.

ALMEIDA, Frederico Viana; CANEDO, Edna Dias; DA COSTA, Ruyther Parente. Definition of indicators in the execution of educational projects with design thinking using the systematic literature review. In: 2019 IEEE Frontiers in Education Conference (FIE). IEEE,

2019. p. 1-9.

WAGNER, Stefan et al. Challenges in survey research. *Contemporary Empirical Methods in Software Engineering*, p. 93-125, 2020.

JESUS, Felipe et al. Design Thinking Evidence in Software Development: Results from a Systematic Mapping. In: *2023 IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*. IEEE, 2023. p. 1-6.

ROWE, Peter G. *Design Thinking*. Cambridge, Massachusetts. 1987.

CARROLL, Noel; RICHARDSON, Ita. Aligning healthcare innovation and software requirements through design thinking. In: *Proceedings of the international workshop on software engineering in healthcare systems*. 2016. p. 1-7.

BUGAYENKO, Yegor et al. Prioritizing tasks in software development: A systematic literature review. *Plos one*, v. 18, n. 4, p. e0283838, 2023.

ANWAR, Sadaf et al. User-centered design practices in scrum development process: A distinctive advantage?. In: *17th IEEE International Multi Topic Conference 2014*. IEEE, 2014. p. 161-166.

INAL, Yavuz et al. Positive developments but challenges still ahead: a survey study on UX professionals' work practices. 2020.

MARTINELLI, Suéllen; LOPES, Larissa; ZAINA, Luciana. UX research in the software industry: an investigation of long-term UX practices. In: *Proceedings of the 21st Brazilian Symposium on Human Factors in Computing Systems*. 2022. p. 1-13

MACHADO, Catarina; CAMPOS, José Creissac. Towards the integration of user interface prototyping and model-based development. In: *2021 International Conference on Graphics and Interaction (ICGI)*. IEEE, 2021. p. 1-8.

KITCHENHAM, Barbara A.; BUDGEN, David; BRERETON, O. Pearl. The value of mapping studies—A participant-observer case study. In: *14th international conference on evaluation and assessment in software engineering (ease)*. BCS Learning Development, 2010.

BRENNER, Walter; UEBERNICKEL, Falk; ABRELL, Thomas. Design thinking as mind-set, process, and toolbox: Experiences from research and teaching at the University of St. Gallen. *Design thinking for innovation: Research and practice*, p. 3-21, 2016.

KUULA, Seppo; HAAPASALO, Harri; KOSONEN, Juha-Matti. Three phases of transforming a project-based IT company into a lean and design-led digital service provider. *IEEE software*, v. 37, n. 2, p. 41-48, 2019.

LEVY, Meira; HULLI, Chen. Design thinking in a nutshell for eliciting requirements of a business process: A case study of a design thinking workshop. In: *2019 IEEE 27th international requirements engineering conference (RE)*. IEEE, 2019. p. 351-356.

HEHN, Jennifer et al. On integrating design thinking for human-centered requirements engineering. *IEEE Software*, v. 37, n. 2, p. 25-31, 2019.

MAHÉ, Nolwen et al. Migrating a software factory to design thinking: Paying attention to people and mind-sets. *IEEE Software*, v. 37, n. 2, p. 32-40, 2019.

STANFORD, H. P. I. of D at: An introduction to design thinking: process guide. 2010.

GURUSAMY, Kavitha; SRINIVASARAGHAVAN, Narayanan; ADIKARI, Sisira. An integrated framework for design thinking and agile methods for digital transformation. In: *Design, User Experience, and Usability: Design Thinking and Methods: 5th International Conference, DUXU 2016, Held as Part of HCI International 2016, Toronto, Canada, July 17–22, 2016, Proceedings, Part I 5*. Springer International Publishing, 2016. p. 34-42.

Apêndices

APÊNDICE A – FORMULÁRIO PARA EXTRAÇÃO E AVALIAÇÃO DA QUALIDADE DOS TRABALHOS

Avaliação da qualidade do trabalho	
*Obs: Aqui será avaliado o critério de qualidade em cada um dos EPs, ou seja, como o trabalho responde a cada pergunta apresentada abaixo.	
Avaliação/Notas:	
<ul style="list-style-type: none"> ● 1.0 - Significa que o estudo cumpriu o critério COMPLETAMENTE. (No caso das perguntas 5,6,7,8 significa que o estudo endereça bem às questões) 	
<ul style="list-style-type: none"> ● 0.5 - Significa que o estudo cumpriu PARCIALMENTE o critério (No caso das perguntas 5,6,7,8 significa que o estudo endereça parcialmente às questões) 	
<ul style="list-style-type: none"> ● 0.0 - Significa que o estudo NÃO cumpriu o critério. (No caso das perguntas 4, 5,6,7,8 significa que o estudo não endereça as questões). 	
CRITÉRIO	AVALIAÇÃO
1. O estudo define os objetivos ou questões de pesquisa de forma clara?	
2. O estudo faz uma clara descrição do contexto na qual a pesquisa foi realizada?	
3. O estudo é coerente e bem estruturado, permitindo avaliação?	
4. O estudo apresenta os papéis de DT utilizados no desenvolvimento de software?	
5. O estudo apresenta modelos de processos?	
6. O estudo apresenta os desafios de se utilizar processos de software?	
7. O estudo apresenta práticas de DT utilizadas no processo de desenvolvimento de software?	
8. O estudo apresenta ferramentas que auxiliam no processo de software?	
TOTAL DE PONTOS	

APÊNDICE B – QUESTÃO DE PESQUISA Nº 1 / DESAFIOS

ID	Desafio	Contexto
1	Tipo de aprendizado entre os agentes	Existem agentes que possuem diferentes tipos de aprendizados, sendo classificados em aprendizados formais e informais.
2	Envolvimento do cliente e valor do produto	Envolver o cliente no processo de desenvolvimento, visando compreender as suas necessidades.
3	Desafios Geográficos	Interação entre pessoas geograficamente distantes.
4	Desafios Ambientais	"O design precisa considerar os desafios geográficos, ambientais, econômicos e socioculturais que podem afetar o acesso físico às Tecnologias de Informação e Comunicação (TIC). Por exemplo, infraestrutura básica, como eletricidade, redes de telecomunicações e serviços de internet devem ser contabilizados . Da mesma forma, a heterogeneidade na cultura e na situação sócio-econômica precisa ser considerada."
5	Desafios Econômicos	"O design precisa considerar os desafios geográficos, ambientais, econômicos e socioculturais que podem afetar o acesso físico às Tecnologias de Informação e Comunicação (TIC). Por exemplo, infraestrutura básica, como eletricidade, redes de telecomunicações e serviços de internet devem ser contabilizados . Da mesma forma, a heterogeneidade na cultura e na situação sócio-econômica precisa ser considerada."
6	Desafios Socioculturais	"O design precisa considerar os desafios geográficos, ambientais, econômicos e socioculturais que podem afetar o acesso físico às Tecnologias de Informação e Comunicação (TIC). Por exemplo, infraestrutura básica, como eletricidade, redes de telecomunicações e serviços de internet devem ser contabilizados . Da mesma forma, a heterogeneidade na cultura e na situação sócio-econômica precisa ser considerada."
7	Controle de Falhas e Defeitos	Gestão de falhas e defeitos, registros dos fatos para que sejam corrigidos e evitados.
8	Monitorar Falhas e Defeitos	Gestão de falhas e defeitos, registros dos fatos para que sejam corrigidos e evitados.
9	Implementação de UCD no processo do produto.	Implantação do design centrado no usuário no processo de desenvolvimento do produto.

10	Identificar hábitos/interesses de consumo do usuário	Identificação dos hábitos e interesses do usuário.
11	Incerteza nos Processos	A falta de entendimento específico do setor ou entre os agentes envolvidos no processo de software, sobre como enfrentar e gerenciar mudanças, pode se tornar um desafio.
12	Implemente o UCD com outros métodos.	Implantação do design centrado no usuário com outros métodos.
13	Ambiente de Desenvolvimento e Infraestrutura	Ambiente de desenvolvimento desfavorável e falta de recursos de infraestrutura podem se tornar desafios iminentes no processo de desenvolvimento de software.
14	Implementação de Design Thinking com outros métodos.	Implantação do Design Thinking com outros métodos.
15	Orçamento limitado	Orçamentos limitados podem se tornar desafios iminentes no processo de desenvolvimento de software.
16	transformação digital	A digitalização e a transformação digital podem se tornar um desafio quando implementadas em instituições tradicionais.
17	Interação entre as partes interessadas e a equipe de design	A transparência na comunicação entre as partes interessadas (Stakeholders) e o time de design podem se tornar um desafio quando não abordadas de forma eficiente.
18	Valor agregado	Envolver o cliente no processo de desenvolvimento, visando compreender as suas necessidades.
19	Integrando a perspectiva do usuário no Agile	Assimilação e aceitação do usuário em relação a metodologia ágil.
20	Gerenciamento de Atividades	Controle e gestão de atividades
21	Encontre o significado dos dados quando eles não explicam o porquê	Exposição de dados de forma errônea, podem comprometer a execução do projeto, tornando-se assim em um desafio.
22	Participação das Partes Interessadas	A falta de interação dos stakeholders no processo de desenvolvimento pode se tornar um desafio.
23	Senso de Urgência e Relevância	A falta de senso de urgência e priorização de requisitos pode se tornar um desafio.

24	Participação Remota em Processos	A participação remota dos agentes envolvidos no processo de desenvolvimento de software pode se tornar um desafio, estando atrelada a falta de comunicação efetiva e limitação de recursos para o projeto.
25	Separação dos processos de trabalho de UX das tarefas de desenvolvimento de software	Segmentar os trabalhos associados aos designers e os trabalhos referente aos desenvolvedores.
26	Mesclar análises de vários profissionais	Unificar a análise / interpretação de dados de diferentes profissionais associados ao processo de desenvolvimento de software pode se tornar um desafio em virtude da variabilidade de opiniões que estarão associadas a esta ação.
27	Esforço colocado	Centralizar o esforço em pontos corretos do projeto, requer habilidade e eficiência, podendo assim, se tornar um desafio para alguns.
28	Envolvimento do Cliente	Envolver o cliente e chamar a sua atenção para algo desejado é um processo desafiador.

APÊNDICE C – QUESTÃO DE PESQUISA Nº 2 / PAPÉIS ASSOCIADOS AO DESIGN THINKING

ID	Papel	ID	Papel
1	Software Engineer	31	User experience Designer
2	Product Manager	32	CEO
3	Product Designer	33	Innovation Facilitator
4	User experience Engineer	34	Lead Ux Researcher
5	User experience Developer	35	User experience Researcher
6	User Interface Developer	36	Service Designer
7	The Usability Specialists	37	User experience Consultant
8	User Study Group	38	Sales Representative
9	End-user	39	Innovation Director
10	Instructor	40	User experience Lead
11	Designer	41	Designer Evangelist
12	Development Team	42	Technology Director
13	Software Architect	43	Designer Thinker
14	Director	44	Service Designer
15	Managing Director	45	Software Developer
16	Director Of Engineering	46	Researcher Designer
17	Consultant	47	Marketing Analyst
18	Designer	48	Experience Designer
19	Electronics Engineer	49	Manager
20	Iteration Manager	50	Scrum Master (SM)
21	Analyst	51	Product Owner (PO)
22	Qa Specialist	52	Transformation Lead
23	User-centered Designer	53	Partners (PA)
24	Operations Lead	54	Back End Developer
25	Coordination Tasks	55	Team Leader
26	Stakeholder	56	Front End Developer
27	Systems Designers	57	Agile Coach
28	Entrepreneur	58	Project manager
29	Business Analyst	59	Business Analyst
30	Technology Consultant	60	Quality Assurance Team Lead

APÊNDICE D – QUESTÃO DE PESQUISA Nº 3 / PRÁTICAS

ID	Práticas	ID	Práticas
1	Workshops,	58	Stand-up Meetings
2	Cookbooks,	59	Requirements Definition
3	Learn-on-the-job Hands-on Immersion Training	60	Participation Of Users/stakeholders
4	Pilot Teams Acting As Enablers	61	Requirement Validation
5	Health-check Assessment	62	Project Schedule Estimation
6	Creation Of Personas	63	Details Of The Requirements
7	Identification Of Personas	64	Prioritization Of Requirements
8	Action Research (ar)	65	Interdependence Between Requirements
9	Cooperative Method Development (cmd)	66	Assign People To Lead The Transformation
10	Understanding The Context Of Use	67	Provide Material
11	Specifying User Requirements	68	Define Strategies
12	Producing Design Solutions To Meet User Requirements	69	Bring Support From Experts 
13	Evaluating The Design Against The Requirements	70	Use Recently Trained People To Train Others
14	Usability Goals	71	Use Experimentation As A Support Strategy
15	User Experience Goals	72	Introduce Agile Software Engineering Practices
16	Contextual Inquiry,	73	Identify Healthcare Problem
17	Paper Prototypes,	74	Identify Software Capabilities
18	Design Cards	75	Aligning Requirements
19	User Stories With Acceptance Criteria	76	Management Of Healthcare And Software Solution
20	Determinism	77	Design Thinking
21	Design History	78	Lean Startup
22	Design Languages	79	Agile
23	Artifact Design	80	Sprints
24	Artifact Transfer	81	Continuous Integration
25	Divergent Thinking Strategies	82	Each Iterations Produces Software

26	Convergent Thinking Strategies	84	Welcome Changes
27	Design	85	Motivation For Developers
28	Interaction between teams	86	Quality Of The Software
29	Flexible Development	87	Analysis And Design Carried
30	Documentation Of Processes	88	Heuristic Evaluation
31	Documentation Of Tasks	89	Focusing On User Real Needs
32	Monitoring Of Processes	90	Prototyping
33	Monitoring Of Tasks	91	User Driven Development
34	Project Plan	92	Usability
35	Sprint Plan	93	User Stories
36	Team Discussion	94	User Journey
37	Brainstorming	95	Wireframe
38	Mind Map	96	Object-oriented Programming (oop)
39	Prototype	97	Dependency Inversion Principle
40	Storyboards	98	Interface Segregation
41	Feedback Capture Grid	99	Liskov Substitution
42	Test	100	Open-closed
43	Review	101	Single-responsibility
44	Unified Modeling Language - Uml	102	Card Sorting,
45	Work Domain Analysis	103	Benchmarking,
46	Conceptual Analysis	104	Mock-up,
47	Usability Studies	105	User Research,
48	Implementation Of Projects	106	Eye-tracking,
49	Debugging Code	107	Psychophysiological Studies
50	Creativity	108	Gamification
51	Group Synergy	109	Research Planning
52	Presenting New Ideas	110	Collecting Data With Users
53	Separate Idea Generation And Evaluation.	111	Data Analysis
54	Preliminary Interviews	112	Communications About Users
55	Feedbacks	113	Research Training
56	Big Design Upfront	114	Separation Of Concerns (soc)
57	Retrospectives	115	Parsons Programming puzzles (ppps)
58	Focal Group	116	Hackathons

APÊNDICE E – QUESTÃO DE PESQUISA Nº 4 / PRINCIPAIS MODELOS

Id	Modelo
1	Waterfall Model
2	Iterative Model
3	Agile Model
4	Human Computer Interaction (HCI)
5	Convergence Model
6	The Hybrid Agile Process Framework
7	Ucd - User Centred Design
8	Fbm - Fogg Behavior Model
9	Model-based Systems Engineering (MBSE)
10	Extended Resources Model (ERM)
11	Design Thinking Model
12	Model For The Innovation Teaching (MOIT)
13	Stage-gate Governance Model
14	Ims-Id (instructional Management System - Learning Design)
15	Model-based Development
16	Flow Model
17	Physical Model
18	Sequence Model
19	Artifact Model
20	Cultural Model
21	Link Model Analysis
22	Scrumban
23	Aidaf Framework
24	Research Model
25	Research Study
26	Design And Development Research (DDR)
27	Spotify Model
28	One Stop Service System Model
29	Analysis, Design, Development, Implementation And Evaluation (ADDIE) Model

30	Innodev Models
31	Nordstrom Model
32	Agile Model Driven Development (AMDD)
33	Model-driven Engineering (MDE)
34	Business Model Canvas
35	Double Diamond Design
36	IBM Model
37	IDEO
38	Stanford Design
39	Mental Models
40	Spotify Model
41	Csforall
42	Model Of Pragmatic Design
43	Grounded Theory
44	SAP LAP
45	Human-centered Design for Interactive Systems Google Design Sprints
46	Dschool
47	Scaled Agile Framework (SAFe)

APÊNDICE F – QUESTÃO DE PESQUISA Nº 5 / FERRAMENTAS

ID	Ferramenta	Tipo
1	Sdk Android	IDE
2	Eclipse	IDE
3	Sqlite Studio	IDE
4	Personas	Design / Edição de Mídias
5	Java	Linguagem de programação
6	Intrinsic Motivation Inventory (imi)	Outros
7	Educational Software	Outros
8	Lcd Software Tools	Outros
9	Django Templates	Prototipagem
10	Wagtail Themes	Prototipagem
11	Vue.js	Linguagem de programação
12	Html	Linguagem de programação
13	Javascript	Linguagem de programação
14	Figma	Prototipagem
15	Sketch	Prototipagem
16	Adobe Xd	Design / Edição de Mídias
17	Hyperresearch	Prototipagem
18	User Interface Transition Diagrams	Prototipagem
19	Detailed Sequence Diagrams	Outros
20	Learning Toolbox	Outros
21	Apis	Linguagem de programação
22	Prototypes	Prototipagem
23	Squamata	Outros
24	Phyton	Linguagem de programação
25	Revit	Prototipagem
26	Archicad	Prototipagem
27	Design thinking Toolbox	Prototipagem
28	Management Tools	Prototipagem
29	Communication Tools	Outros
30	Normatic Tool	Outros
31	Marvel App	Prototipagem
32	Paint	Prototipagem

33	Evrybo	Outros
34	Xmind-stakeholder Map	Outros
35	Miro	Gestão de tarefas
36	Whimsical	Gestão de tarefas
37	Mural	Gestão de tarefas
38	Real Time	Gestão de tarefas
39	Muraly	Gestão de tarefas
40	Build	Gestão de tarefas
41	Canvanizer	Gestão de tarefas
42	Google Sheets	Gestão de tarefas
43	Adobe Xd	Design / Edição de Mídias
44	Google Presentation	Gestão de tarefas
45	Pop	Gestão de tarefas
46	Mindmeister	Gestão de tarefas
47	Nvision	Gestão de tarefas
48	Hotjar	Gestão de tarefas
49	Survey Monkey	Gestão de tarefas
50	Illustrator	Design / Edição de Mídias
51	Photoshop	Design / Edição de Mídias
52	Sap Build	Gestão de tarefas
53	Samplly	Gestão de tarefas
54	Strategyzer	Gestão de tarefas
55	Axure Rp	Prototipagem
56	Touchpoint	Gestão de tarefas
57	Creately	Creately
58	Circle	Creately
59	Positioning Matrix	Positioning Matrix
60	Nvivo Software Tool	Gestão de tarefas
61	.net	Linguagem de programação
62	Php	Linguagem de programação
63	Content Management System - Cms	Gestão de tarefas
64	Post-its	Gestão de tarefas

65	Diagramming	Gestão de tarefas
66	Spss Statistics	Gestão de tarefas
67	Css	Linguagem de programação
68	Pixate	Prototipagem
69	Axure	Prototipagem
70	Microsoft Visual Studio	IDE
71	Openideo	Prototipagem
72	Dell Idea Storm	Prototipagem
73	Ustesting.com	Prototipagem
74	Fullstory	Gestão de tarefas
75	Mixpanel	Gestão de tarefas