

UNIVERSIDADE FEDERAL DO PAMPA

WESLEY SOARES DE SOUZA

**GESTÃO DO CONHECIMENTO EM
ARQUITETURAS DE SOFTWARE: UMA
ANÁLISE EXPERIMENTAL**

**Alegrete
2025**

WESLEY SOARES DE SOUZA

**GESTÃO DO CONHECIMENTO EM
ARQUITETURAS DE SOFTWARE: UMA
ANÁLISE EXPERIMENTAL**

Projeto de Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Software como requisito parcial para a obtenção do título de Mestre em Engenharia de Software.

Orientador: Dr. Gilleanes Thorwald Araujo
Guedes
Coorientador: Dr. Williamson Alison Freitas
Silva

**Alegrete
2025**

Ficha catalográfica elaborada automaticamente com os dados fornecidos
pelo(a) autor(a) através do Módulo de Biblioteca do
Sistema GURI (Gestão Unificada de Recursos Institucionais) .

S719g Souza, Wesley Soares de
GESTÃO DO CONHECIMENTO EM ARQUITETURAS DE SOFTWARE: UMA
ANÁLISE EXPERIMENTAL / Wesley Soares de Souza.
132 p.

Dissertação(Mestrado)-- Universidade Federal do Pampa,
MESTRADO EM ENGENHARIA DE SOFTWARE, 2025.

"Orientação: Gilleanes Thorwald Araujo Guedes".

1. Gestão do Conhecimento Arquitetural. 2. Engenharia de
software. 3. Arquitetura de software. I. Título.

WESLEY SOARES DE SOUZA

GESTÃO DO CONHECIMENTO EM ARQUITETURAS DE SOFTWARE: UMA ANÁLISE EXPERIMENTAL

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Software da Universidade Federal do Pampa, como requisito parcial para obtenção do Título de Mestre em Engenharia de Software

Dissertação defendida e aprovada em: 14/03/2025

Banca examinadora:

Prof. Dr. Gilleanes Thorwald Araujo Guedes

Orientador

(Unipampa)

Prof. Dr. Williansom Alison Freitas Silva (coorientador)

(Unipampa)

Prof. Dr. Silvio Ereno Quincozes

(sigla da instituição)

Profª Drª. Raquel Mainardi Pillat Basso
(Unipampa)

Prof Dr. Rayfran Rocha Lima
(SIDIA)



Assinado eletronicamente por **GILLEANES THORWALD ARAUJO GUEDES, PROFESSOR DO MAGISTERIO SUPERIOR**, em 14/03/2025, às 12:03, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.



Assinado eletronicamente por **RAQUEL MAINARDI PILLAT BASSO, PROFESSOR DO MAGISTERIO SUPERIOR**, em 14/03/2025, às 12:04, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.



Assinado eletronicamente por **WILLIAMSON ALISON FREITAS SILVA, PROFESSOR DO MAGISTERIO SUPERIOR**, em 14/03/2025, às 12:05, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.



Assinado eletronicamente por **Rayfran Rocha Lima, Usuário Externo**, em 14/03/2025, às 12:06, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.



Assinado eletronicamente por **SILVIO ERENO QUINCOZES, PROFESSOR DO MAGISTERIO SUPERIOR**, em 14/03/2025, às 12:06, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.



A autenticidade deste documento pode ser conferida no site https://sei.unipampa.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **1684818** e o código CRC **3C73090E**.

"Não foi isso que eu ordenei? Seja forte e corajoso! Não tenha medo, nem fique assustado, porque o Senhor, seu Deus, estará com você por onde quer que você andar."

— (Josué 1:9)

AGRADECIMENTOS

Agradeço primeiramente a Deus, por ter sempre me dado forças e direção para prosseguir com esse sonho acadêmico, à minha família, em especial a minha esposa e filhos, que estão sempre ao meu lado me apoiando, e aos meus pais e irmã, que são uma presença constante em todos os momentos da minha vida. Sei que posso sempre contar com vocês e tenho um imenso orgulho disso.

Gostaria de expressar minha profunda gratidão ao meu orientador, o Prof. Dr. Gilleanes Thorwald Araujo Guedes e ao Prof Dr. Williamson Silva, por gentilmente aceitarem me orientar. Sempre disponíveis em tirar minhas dúvidas e as correções minuciosas que realizou no texto foram essenciais para a conclusão deste trabalho. Sem a orientação de vocês, seria impossível alcançar esse resultado.

Por fim, não posso deixar de mencionar minha gratidão à Universidade Federal do Pampa (UNIPAMPA) por proporcionar um ensino de qualidade de forma gratuita. Sou imensamente grato pela oportunidade de estudar em uma instituição que valoriza a educação e contribui para o meu crescimento acadêmico e pessoal.

Obrigado a todos!

Este trabalho é dedicado à minha esposa e
filhos que me apoiaram durante seu
desenvolvimento. A minha mãe e minha
irmã, por estarem presentes. Aos meus
amigos pelo companheirismo que ultrapassa
o tempo!

RESUMO

A indústria de software tem crescido exponencialmente, impulsionada pela transformação digital e pela demanda por automação. A Gestão do Conhecimento (GC) tem se tornado essencial para a continuidade desse cenário, permitindo o armazenamento e reuso eficiente de aprendizados. No contexto da GC, a Gestão do Conhecimento Arquitetônico (GCA) documenta e aprimora decisões estratégicas, garantindo a evolução dos sistemas e a criação de soluções escaláveis e inovadoras. Esta pesquisa busca melhorar a aplicação da GCA em uma organização de software, identificando seus desafios e propondo uma solução mais prática, colaborativa e ágil. Para isso, foi realizada uma extensão de uma Revisão Sistemática da Literatura (RSL) para compreender a evolução da GC na indústria de software. Os resultados evidenciaram a necessidade de maior investigação na área de conhecimento arquitetônico (CA), sua gestão e a avaliação de ferramentas específicas para sua aplicação. Por meio do método de pesquisa-ação, constatou-se que o conhecimento já existe na organização objeto da pesquisa, mas está disperso em diversos canais, dificultando o acesso e a disseminação, especialmente para novos colaboradores. Como solução, foi desenvolvido um processo para centralizar esse conhecimento em uma única ferramenta já utilizada pela organização, mas ainda não aplicada para esse fim. A avaliação, baseada no modelo TAM (Technology Acceptance Model), mostrou que a centralização da GCA trouxe ganhos de tempo, melhoria na qualidade do desenvolvimento e maior troca de conhecimento entre as equipes. O processo foi adaptado às necessidades da organização e aceito pelo time de arquitetura, com futuras melhorias em estudo. Entre as principais contribuições, destacam-se a melhoria da comunicação entre gestores e equipes, o acesso rápido às decisões e a padronização de práticas, facilitando a integração dos sistemas e fortalecendo a colaboração organizacional. Os resultados apresentados nesta dissertação permitem o surgimento de novas perspectivas de pesquisa para a Gestão do Conhecimento Arquitetônico (GCA), como: criação de um catálogo de ações práticas para melhoria da estrutura dos projetos e seus respectivos feedbacks e melhoria no modelo sugerido para GCA em relação a sua usabilidade.

Palavras-chave: Gestão do Conhecimento Arquitetural; Engenharia de software; Arquitetura de software.

ABSTRACT

The software industry has experienced exponential growth, driven by digital transformation and the increasing demand for automation. Knowledge Management (KM) has become essential for sustaining this progress, enabling the efficient storage and reuse of learning. Within KM, Architectural Knowledge Management (AKM) plays a crucial role in documenting and enhancing strategic decisions, ensuring system evolution and the development of scalable, innovative solutions. This research aims to improve the application of AKM in a software organization by identifying its challenges and proposing a more practical, collaborative, and agile solution. To achieve this, an extension of a Systematic Literature Review (SLR) was conducted to understand the evolution of KM in the software industry. The results highlighted the need for further investigation in the field of Architectural Knowledge (AK), its management, and the evaluation of specific tools for its application. Through an action research approach, it was observed that architectural knowledge already exists within the studied organization but is dispersed across multiple channels, hindering access and dissemination, especially for new employees. As a solution, a process was developed to centralize this knowledge within a single tool already in use by the organization but not previously leveraged for this purpose. Evaluation based on the Technology Acceptance Model (TAM) demonstrated that centralizing AKM led to time savings, improved software development quality, and enhanced knowledge exchange among teams. The process was adapted to the organization's needs and accepted by the architecture team, with further refinements under consideration. Key contributions include improved communication between managers and teams, quick access to decision records, and standardization of practices, facilitating system integration and strengthening organizational collaboration. The findings presented in this dissertation open new research perspectives for AKM, such as creating a catalog of practical actions to enhance project structures and feedback mechanisms, as well as improving the proposed AKM model in terms of usability.

Keywords: Architectural Knowledge Management; Software Engineering; Software Architecture.

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1 Metodologia adotada na pesquisa..... | 18 |
| Figura 2 Processo de busca e seleção..... | 31 |
| Figura 3 As etapas básicas de um projeto de Pesquisa-ação..... | 51 |
| Figura 4 As etapas de um projeto de Pesquisa-ação..... | 52 |
| Figura 5 Etapas para realização da pesquisa..... | 53 |
| Figura 6 Distribuição do conhecimento organizacional..... | 57 |
| Figura 7 Gestão do conhecimento em arquitetura de software na organização..... | 61 |
| Figura 8 Competências organizacionais..... | 62 |
| Figura 9 Gestão do Conhecimento Arquitetônico (GCA)..... | 63 |
| Figura 10 Modelo para centralização da GCA..... | 68 |
| Figura 11 Processo adotado para aplicação do modelo proposto..... | 71 |
| Figura 12 Cronograma para implantação do processo de GCA..... | 73 |
| Figura 13 Apresentação do produto no Backstage®..... | 76 |
| Figura 14 Modelo de ADR (Architecture Decision Recorder)*..... | 77 |
| Figura 15 Atualização do modelo de GCA aplicado..... | 86 |
| Figura 16 RDA local para ser compartilhada..... | 87 |
| Figura 17 Percepção dos participantes sobre a GCA..... | 92 |

LISTA DE TABELAS

| | | |
|-----------|--|-----|
| Tabela 1 | Questões de Pesquisa e suas Motivações..... | 29 |
| Tabela 2 | String de busca adotada nas bibliotecas. | 30 |
| Tabela 3 | Abordagens identificadas na RSL original de Maciel et al. (2018)..... | 46 |
| Tabela 4 | Abordagens identificadas nos estudos selecionados em nossa RSL | 47 |
| Tabela 5 | Visão Geral dos participantes do estudo..... | 59 |
| Tabela 6 | Questionário elaborado para coletar as percepções dos desenvolvedores..... | 60 |
| Tabela 7 | Participantes por equipe..... | 80 |
| Tabela 8 | Novos integrantes equipe..... | 81 |
| Tabela 9 | Afirmativas nos indicadores do modelo TAM | 90 |
| Tabela 10 | Avaliação Qualitativa a respeito do processo proposto | 90 |
| Tabela 11 | Participantes por equipe..... | 91 |
| Tabela 12 | Respostas Qualitativas 1 | 122 |
| Tabela 13 | Respostas Qualitativas 2 | 123 |
| Tabela 14 | (KT1) Em se tratando do compartilhamento de conhecimento pelo time de arquitetura para com os times de produto e os times de produto entre si, o Backstage® se mostra promissor? Fale sua opinião e sugestões a respeito desse tema. (parte A) | 124 |
| Tabela 15 | (KT1) Em se tratando do compartilhamento de conhecimento pelo time de arquitetura para com os times de produto e os times de produto entre si, o Backstage® se mostra promissor? Fale sua opinião e sugestões a respeito desse tema.(parte B) | 125 |
| Tabela 16 | (KT2) Em relação a geração de conhecimento entre os times, na sua opinião, o backstage facilita esse processo? O que poderia ser melhorado ou acrescentado para que isso ocorra? (parte A) | 126 |
| Tabela 17 | (KT2) Em relação a geração de conhecimento entre os times, na sua opinião, o backstage facilita esse processo? O que poderia ser melhorado ou acrescentado para que isso ocorra? (parte B) | 127 |
| Tabela 18 | (KT3) O backstage pode facilitar o processo de onboarding de novos integrantes ao time, com links úteis e informações essenciais a respeito do produto? (Sonar, Repositório, API's, etc) (parte A)..... | 128 |
| Tabela 19 | (KT3) O backstage pode facilitar o processo de onboarding de novos integrantes ao time, com links úteis e informações essenciais a respeito do produto? (Sonar, Repositório, API's, etc) (parte B)..... | 129 |
| Tabela 20 | (KT4) Ao observar o Backstage como um centralizador de conhecimento organizacional, você acredita que ele se mostra eficiente? Nesse sentido, quais são as informações que você considera como úteis para a centralização do conhecimento organizacional que pode ocorrer entre a arquitetura e entre equipes?(parte A)..... | 129 |
| Tabela 21 | (KT4) Ao observar o Backstage como um centralizador de conhecimento organizacional, você acredita que ele se mostra eficiente? Nesse sentido, quais são as informações que você considera como úteis para a centralização do conhecimento organizacional que pode ocorrer entre a arquitetura e entre equipes? (parte B) | 130 |

| | |
|---|-----|
| Tabela 22 (KT5) O que você mudaria dentro do Backstage e/ou na apresentação dos artefatos disponibilizados em cada produto? Cite os pontos positivos, negativos e limitações que você percebeu na utilização do backstage. Sugestões são sempre bem vindas! (parte A) | 131 |
| Tabela 23 (KT5) O que você mudaria dentro do Backstage e/ou na apresentação dos artefatos disponibilizados em cada produto? Cite os pontos positivos, negativos e limitações que você percebeu na utilização do backstage. Sugestões são sempre bem vindas! (parte C) | 132 |

SUMÁRIO

| | |
|--|-----------|
| 1 INTRODUÇÃO | 15 |
| 1.1 Contexto | 15 |
| 1.2 Objetivo geral | 17 |
| 1.3 Objetivos específicos | 17 |
| 1.4 Metodologia | 17 |
| 1.5 Organização do Trabalho | 19 |
| 2 REFERENCIAL TEÓRICO | 21 |
| 2.1 Gestão do Conhecimento (GC) | 21 |
| 2.2 Gestão do Conhecimento Arquitetural (GCA)..... | 22 |
| 2.3 Considerações Finais | 26 |
| 3 REVISÃO SISTEMÁTICA DA LITERATURA | 27 |
| 3.1 Motivação..... | 27 |
| 3.2 Método de pesquisa..... | 28 |
| 3.2.1 Planejamento..... | 29 |
| 3.2.1.1 Questões de pesquisa | 29 |
| 3.2.1.2 Bases de pesquisa..... | 29 |
| 3.2.1.3 <i>String</i> de busca | 30 |
| 3.2.1.4 Critérios de inclusão e exclusão..... | 30 |
| 3.2.2 Condução da RSL | 31 |
| 3.2.3 Estratégia de extração dos dados | 32 |
| 3.3 Análise dos Resultados | 32 |
| 3.3.1 Propósito do diagnóstico de GC (QP1) | 32 |
| 3.3.2 Características do diagnóstico (QP2)..... | 36 |
| 3.4 Discussões | 42 |
| 3.5 Ameaças à Validade | 44 |
| 3.6 Conclusões e trabalhos futuros | 45 |
| 4 PROTOCOLO DA PESQUISA | 48 |
| 4.1 Motivação..... | 48 |
| 4.2 Pesquisa-ação..... | 49 |
| 4.3 O método e a natureza da pesquisa | 51 |
| 5 FASE INICIAL DA PESQUISA-AÇÃO | 53 |
| 5.1 Identificação do problema | 54 |
| 5.2 Reconhecimento e Descoberta de Fatos sobre o Contexto do Problema..... | 56 |
| 5.2.1 Análise dos Canais de Comunicação Existentes..... | 57 |
| 5.2.2 Análise Diagnóstica com os Times de Produto | 59 |
| 5.2.2.1 Execução do experimento | 60 |
| 5.2.2.2 Resultados Quantitativos..... | 61 |
| 5.2.2.3 Análise Qualitativa..... | 63 |
| 5.2.2.4 Resultados da Análise Qualitativa | 64 |
| 5.2.3 Conclusão sobre o Contexto do Problema | 66 |
| 5.3 Atividades de Planejamento da Solução | 67 |
| 5.3.1 Modelo proposto para centralização da GCA | 68 |
| 5.3.2 Ferramenta sugerida para a centralização da GCA..... | 69 |
| 5.3.3 Planejamento e desenvolvimento de estratégias e ações..... | 70 |
| 5.3.4 Cronograma..... | 72 |
| 5.3.5 Conclusão do planejamento da solução | 73 |
| 5.4 Orientações para aplicação do modelo proposto (<i>guideline</i>) | 74 |
| 5.4.1 Princípios a serem adotados..... | 74 |

| | |
|---|------------|
| 5.4.2 Etapas da implementação..... | 75 |
| 6 PRIMEIRO CICLO DO PROCESSO DE PESQUISA-AÇÃO | 76 |
| 6.1 Implementação da solução proposta | 76 |
| 6.2 Monitoramento da solução..... | 78 |
| 6.3 Avaliação e aperfeiçoamentos necessários | 79 |
| 6.4 Conclusão do primeiro ciclo..... | 83 |
| 7 SEGUNDO CICLO DO PROCESSO DE PESQUISA-AÇÃO | 85 |
| 7.1 Implementação do aperfeiçoamento proposto | 85 |
| 7.2 Monitoramento..... | 88 |
| 7.3 Avaliação do efeito das ações..... | 89 |
| 7.3.1 Visão geral dos participantes | 91 |
| 7.4 Resultados quantitativos obtidos do questionário TAM | 92 |
| 7.5 Percepção dos desenvolvedores sobre a centralização da GCA | 94 |
| 7.5.1 Transferência de conhecimento (KT - <i>Knowledge Transfer</i>)..... | 94 |
| 7.5.2 Eficiência para geração de conhecimento pelo processo sugerido | 96 |
| 7.5.3 Avaliação da Ferramenta: Eficiência e Percepções dos Usuários..... | 98 |
| 7.5.4 Reflexão e avaliação final..... | 99 |
| 7.6 Validação dos Resultados obtidos na pesquisa-ação..... | 101 |
| 8 DISCUSSÃO DOS RESULTADOS | 103 |
| 8.1 Conclusões | 103 |
| 8.2 Limitações do trabalho | 106 |
| 8.3 Contribuições da Pesquisa..... | 107 |
| 8.3.1 Do ponto de vista acadêmico | 107 |
| 8.3.2 Do ponto de vista da indústria de software | 108 |
| 8.4 Publicações e perspectivas futuras | 109 |
| REFERÊNCIAS..... | 111 |
| APÊNDICE A — QUESTIONÁRIO 1 | 122 |
| APÊNDICE B — QUESTIONÁRIO 2..... | 124 |

1 INTRODUÇÃO

Este capítulo apresenta a introdução desta dissertação de mestrado. Com o propósito de contextualizar a pesquisa, sendo apresentados os objetivos, a metodologia seguida e a organização deste trabalho.

1.1 Contexto

Considerando o papel do Conhecimento Organizacional (CO) na indústria de software como um fator crucial para a qualidade dos produtos desenvolvidos, Paiva (2006) afirma que a integração desse conhecimento é mais importante para a competitividade do que o conhecimento em si.

A implementação de estratégias eficazes de aquisição e Gestão do Conhecimento (GC) é fundamental para preservar conhecimentos organizacionais e fornecer meios de compartilhá-los dentro da organização (TOMLIN; TOMLIN, 2018).

Na indústria de software, a constante evolução da tecnologia afeta o processo de desenvolvimento. Rabelo et al. (2015) mostram que o conhecimento da Engenharia de Software (ES) é dinâmico e depende da tecnologia, da cultura organizacional e das necessidades mutáveis das práticas de desenvolvimento de software. Existem diversas formas de compartilhar o conhecimento, conforme apontado por (PEDRAZA-GARCIA; ASTUDILLO; CORREAL, 2016), como o modelo de anotação para aquisição de conhecimento, mapeamento de decisão para recuperação e cenários de geração e utilização para compartilhamento de conhecimento.

Ao analisar os procedimentos adotados no desenvolvimento de software, um dos primeiros passos está diretamente ligado às decisões arquiteturais definidas inicialmente. De acordo com Hilliard (2011), a Norma ISO/IEC/IEEE 42010 destaca a importância de se definir a arquitetura do software, sendo um artefato essencial para a qualidade no desenvolvimento.

De acordo com Solms (2012):

"As responsabilidades de um arquiteto de software incluem projetar, documentar, avaliar e evoluir arquiteturas, dentro das quais as funcionalidades podem ser implantadas e executadas, de tal forma que os requisitos de qualidade do sistema sejam atendidos."

Entre essas responsabilidades, o arquiteto também deve realizar a **Gestão do Conhecimento Arquitetural (GCA)**.

A **GCA**, presente na organização, preocupa-se em identificar e alavancar o conhecimento de arquitetura de software em todos os setores de desenvolvimento (WEINREICH; GROHER, 2016). Portanto, a GCA precisa ser automatizada ou semi-automatizada por ferramentas adequadas (LIANG; AVGERIOU, 2009).

Tendo isso em mente, percebe-se a necessidade de transformar a GCA em algo cultural na organização, e não apenas uma obrigação, resultando em maior qualidade no produto de software desenvolvido. Boer and Vliet (2008) destacam que o conhecimento arquitetural na organização refere-se, primeiramente, ao estado atual do produto de software, refletindo as decisões estruturais de arquitetura tomadas anteriormente. Por outro lado, também pode se referir ao estado desejado para a qualidade do produto de software.

A maior parte das organizações de desenvolvimento de software utiliza o modelo ágil para desenvolver seus produtos. As metodologias ágeis revolucionaram o cenário do gerenciamento de projetos, oferecendo uma abordagem dinâmica e flexível que contrasta fortemente com os métodos tradicionais e mais rígidos (DARAOJIMBA et al., 2024). As práticas ágeis apoiam a gestão do conhecimento por meio de atividades como discussões, reuniões diárias do SCRUM, reuniões de planejamento de sprint, entre outras (ANDRIYANI; HODA; AMOR, 2017). Dessa forma, a representação do conhecimento arquitetural deve ser integrada ao processo ágil, de modo que surja como consequência das atividades de desenvolvimento de software no dia a dia (ELORANTA; KOSKIMIES, 2014).

O conhecimento arquitetural, como visto, é intrínseco às organizações, principalmente na indústria de software. Tuzovsky, Yampolsky and Chirikov (2005) enfatizam que a estrutura dos domínios de conhecimento é formada com base na inspeção das atividades da empresa e na discussão com lideranças e especialistas. Em grandes organizações, canais técnicos e de negócios frequentemente são criados por diversas equipes com formas de armazenamento, consulta e compartilhamento próprios, o que exige esforços significativos para padronizar e centralizar seu acesso.

Sendo assim, para facilitar o processo de desenvolvimento dos produtos de software, as organizações contam com uma equipe de Arquitetura de Software, que tem um papel crucial na estruturação do conhecimento técnico. Dingsøyr and Vliet (2009) afirmam que a arquitetura é uma abstração transferível da estrutura de um sistema de software, a ser lida e inspecionada por outras pessoas.

É evidente que o conhecimento arquitetural forma a base para a reutilização de software. Portanto, percebe-se a necessidade do setor de arquitetura de software centralizar seu conhecimento, de modo que todos os integrantes da organização possam ter fácil acesso a ele.

1.2 Objetivo geral

O objetivo principal desta dissertação de mestrado consiste em melhorar a aplicação da **Gestão do Conhecimento Arquitetural (GCA)** em uma organização de software identificando seus principais desafios e trazendo uma solução para torná-la mais prática, colaborativa e ágil, visando a melhoria da comunicação, do acesso e do compartilhamento de conhecimento.

1.3 Objetivos específicos

Para alcançar o objetivo geral, foram definidos os seguintes objetivos específicos:

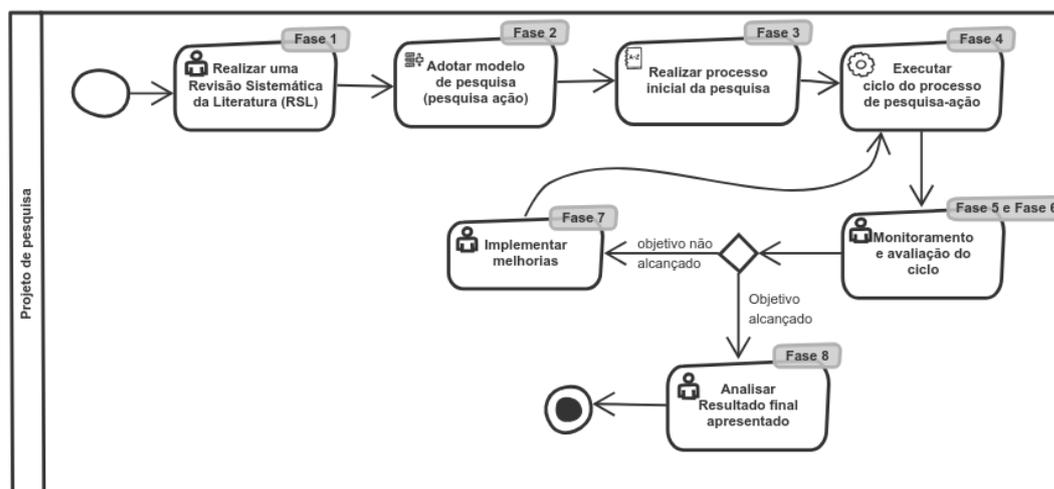
- Compreender o cenário atual da Gestão do Conhecimento na indústria de software por meio de uma Revisão Sistemática da Literatura (RSL);
- Identificar as necessidades apresentadas na literatura e verificar sua validade em uma organização de desenvolvimento de software;
- Sistematizar a aplicação da gestão do conhecimento no setor de arquitetura de software;
- Avaliar, por meio de um estudo experimental, a validade da melhoria no processo de compartilhamento da GCA em uma organização de software.

1.4 Metodologia

Para o desenvolvimento desta pesquisa, foi definida como metodologia de trabalho a proposta apresentada na **Figura 1**. Pretende-se realizar uma avaliação incremental para a inserção de um novo processo na organização, por meio de um software.

Para isso, de acordo com Shull, Carver and Travassos (2001), o processo que se pretende aplicar deve basear-se na observação dos problemas reais enfrentados pela indústria, bem como na eficácia das medidas já existentes e na aplicação de abordagens práticas.

Figura 1 – Metodologia adotada na pesquisa.



Fonte: O autor.

Na primeira fase, foi realizada uma investigação sobre a Gestão do Conhecimento (GC) e sua aplicação na indústria de software. Para isso, desenvolveu-se uma extensão da Revisão Sistemática da Literatura (RSL) de Maciel et al. (2018), a fim de compreender melhor as principais necessidades encontradas na indústria de software até o momento.

Na segunda fase, foi definido o modelo de pesquisa a ser aplicado. Decidiu-se que a melhor opção seria o método pesquisa-ação, que, conforme Staron (2020), é um método coparticipativo no processo de investigação, no qual a academia e a indústria podem trabalhar em conjunto.

Por meio dessa cooperação, em que o pesquisador está diretamente envolvido por ser um colaborador da organização, juntamente com os outros profissionais, torna-se possível desenvolver resultados que contribuem tanto para a prática industrial quanto para as teorias, ferramentas, métodos e o conhecimento acadêmico.

De acordo com McKay and Marshall (2001), o pesquisador-ação deve tomar consciência de um problema do mundo real, o que fornece espaço para a elucidação de temas ou ideias de pesquisa. Sendo assim, na terceira fase, foram realizadas as etapas iniciais na organização objeto de estudo da pesquisa, incluindo a identificação do problema, a des-

coberta de fatos sobre o contexto do problema e a atividade de planejamento da solução.

Na quarta, quinta, sexta e sétima fases, são realizados os ciclos propostos pela pesquisa-ação para avaliação contínua até que se alcance o objetivo proposto. Na quarta fase, foi implementada a solução proposta; na quinta fase foi realizado o monitoramento do que foi aplicado, na sexta fase foi realizada a avaliação de como a aplicação da solução estava sendo recebida e seu retorno pelos usuários. Caso o objetivo proposto não seja alcançado, são realizadas implementações de melhorias na sexta fase, retornando-se à quinta fase.

Na oitava fase, avaliamos os resultados alcançados, após a conclusão de todos os ciclos, e traçamos possíveis melhorias futuras para o projeto. Com o término dessa etapa, foi possível relacionar os dados identificados para a conclusão deste trabalho e suas possíveis contribuições.

1.5 Organização do Trabalho

Esta dissertação está organizada em cinco capítulos, além deste primeiro capítulo, que apresentou os aspectos relevantes para a construção deste trabalho. A organização do texto segue a estrutura abaixo:

- **Capítulo 2 – Referencial Teórico:** Apresenta o embasamento teórico sobre os temas que fazem parte da concepção do processo para aplicação prática da GCA, bem como o *background* necessário em relação à transferência de conhecimento organizacional;
- **Capítulo 3 – Revisão Sistemática da Literatura:** Apresenta uma extensão de uma revisão sistemática da literatura sobre Gestão do Conhecimento na indústria de software e os apontamentos relacionados à GCA;
- **Capítulo 4 – Protocolo da pesquisa:** Descreve a organização objeto de estudo e a abordagem metodológica adotada para a pesquisa;
- **Capítulo 5 – Fase inicial da pesquisa-ação:** Demonstra a primeira fase da pesquisa, incluindo a identificação do problema, a descoberta de fatos sobre o contexto do problema e o planejamento da solução;
- **Capítulo 6 – Primeiro Ciclo do Processo de Pesquisa-Ação:** Apresenta a execução do primeiro ciclo da pesquisa-ação, sua avaliação e o aperfeiçoamento da

solução proposta;

- **Capítulo 7 - Segundo Ciclo do Processo de Pesquisa-Ação** Descreve a execução do segundo ciclo da pesquisa-ação, a avaliação dos resultados finais e as melhorias propostas na ferramenta de GCA;
- **Capítulo 8 - Discussão dos Resultados:** Contém a avaliação e análise do experimento, os limitadores que podem ter influenciado os resultados, uma publicação realizada como contribuição deste trabalho e indicações para a continuidade da pesquisa por meio de possíveis trabalhos futuros.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta o embasamento teórico utilizado neste trabalho, visando definir aspectos inerentes à gestão do conhecimento na indústria de software e a importância de sua aplicação no setor de arquitetura em ambientes de desenvolvimento de software, bem como o método adotado para abordar a temática no desenvolvimento da pesquisa.

2.1 Gestão do Conhecimento (GC)

Para assegurar um desenvolvimento de software saudável e contínuo nas organizações, é necessário que as pessoas envolvidas e os processos operem em harmonia. Para tanto, a compreensão do que é o conhecimento dentro da organização se mostra essencial.

Tal definição surge a partir do momento em que uma informação pode ser explorada de forma a gerar conhecimento que auxilie no desenvolvimento do software. O conhecimento organizacional pode ser visto, então, como uma forma superior de informação, elevada pela natureza e propósito específicos da organização, fornecendo uma oportunidade para explorar suas vantagens estratégicas (BECKETT; WAINWRIGHT; BANCE, 2000). Segundo Lindvall and Rus (2003), a gestão do conhecimento (GC) concentra-se em facilitar o acesso dos indivíduos ao conhecimento certo no momento certo.

Embora o conhecimento seja inerentemente enraizado socialmente, deve-se considerar que suas soluções se desdobram em sistemas humanos complexos, práticas em equipes, zonas de conhecimento e sistemas orgânicos (BJØRNSON; DINGSØYR, 2008). À medida que a GC se estende por esses diferentes estratos, ela se torna uma peça fundamental na promoção da inovação e na integração de práticas eficientes, alinhando-se às demandas dinâmicas da indústria de software (CORBIN; DUNBAR; ZHU, 2007).

Ao compreender a importância desse conhecimento, mostra-se necessário, segundo Kulkarni and Louis (2003), alcançar a maturidade na GC dentro da organização, o que exerce um impacto positivo notável no desenvolvimento de sistemas de informação. A GC, enquanto processo contínuo de aquisição e aplicação do conhecimento adquirido, aprimora a tomada de decisões e reflete minuciosamente as atividades executadas na organização, considerando todas as possíveis restrições e impedimentos em cada projeto (PEDRAZA-GARCIA; ASTUDILLO; CORREAL, 2016).

A GC, nesse cenário, oferece uma compreensão mais profunda do processo de de-

envolvimento de software, trazendo clareza ao domínio do problema (WAHEED et al., 2019). De acordo com Ives, Torrey and Gordon (1997) , ao pesquisar GC na indústria de software, é necessário ser flexível, sensível a diferentes pontos de vista e ter um olhar para a Gestão do Conhecimento como uma jornada exploratória, em vez de um destino definido. Dessa forma, é importante realizar uma auditoria de aprendizagem para compreender os fluxos e sistemas envolvidos, assim como os elementos usados para extrair comportamentos e crenças dentro da organização.

Dito isso, o primeiro passo para a implementação ou melhoria no processo de GC na organização está em definir o que realmente se constitui como conhecimento dentro de um cenário determinado e, dessa maneira, criar uma estrutura contendo as principais competências no departamento de desenvolvimento de software, que devem ser disponibilizadas e divididas pelos tipos de tecnologia utilizadas (BECKETT; WAINWRIGHT; BANCE, 2000). O gerenciamento centralizado de processos por meio de um sistema de GC surge como uma necessidade premente nas organizações, exigindo até o desenvolvimento de ferramentas para conectar processos similares, especialmente no desenvolvimento de software (MONTONI et al., 2004).

2.2 Gestão do Conhecimento Arquitetural (GCA)

É muito comum o início do desenvolvimento de um software ocorrer sem uma arquitetura formal definida (RICHARDS, 2015). No entanto, o resultado dessa prática costuma ser módulos de código-fonte desorganizados, sem funções e relacionamentos claros entre si. De acordo com Fowler (2003), a arquitetura de software pode ser caracterizada como as decisões de design necessárias para iniciar um projeto.

Para a definição do projeto arquitetural de um produto de software, é necessário realizar o levantamento dos requisitos funcionais e não funcionais, pois é por meio deles que se define tanto o que o sistema deve fazer quanto como ele deve operar. Essa abordagem destaca a importância de identificar e gerenciar os aspectos fundamentais de um sistema de software, reconhecendo que a definição do que é "importante" pode variar conforme o contexto e os objetivos do projeto (FOWLER, 2003).

Os requisitos funcionais, segundo Sommerville (2015), especificam o comportamento do sistema, o processamento de dados e as interações que o sistema deve realizar para o usuário. Esses requisitos influenciam a organização estrutural e o design dos componentes do projeto.

Já os requisitos não funcionais, segundo Bass, Clements and Kazman (2021), definem os atributos de qualidade, as restrições de design e as interfaces externas de um sistema de software. Sua principal característica é impactar a arquitetura mais do que os requisitos funcionais, pois é por meio deles que são definidas decisões estruturais, como camadas, distribuição e implantação.

Para a aplicação de um modelo arquitetural em um projeto de software, é necessário ter o Conhecimento Arquitetural (CA) bem claro e acessível para os desenvolvedores. O primeiro passo para a formação do CA está no levantamento dos requisitos funcionais e não funcionais do projeto a ser desenvolvido.

O projeto arquitetural de software é uma pequena parte do CA necessário para o projeto. Esse tipo de conhecimento pode ser categorizado como tácito, documentado ou formalizado, sendo este último organizado de forma sistêmica. No entanto, o CA formalizado pode se tornar inválido após modificações, tornando necessário reter, além da versão atual, seu histórico (KRUCHTEN; LAGO; VLIET, 2006).

Uma vez que o processo de tomada de decisão arquitetural na organização torna-se mais claro e acessível às partes interessadas, também se torna possível documentar os registros do CA e seus fundamentos em um software específico. As bases para a formação dessas decisões arquiteturais fornecem informações sobre os modelos que devem ser considerados no projeto, as forças que influenciaram tais decisões e as restrições que possam ter surgido nesse processo (BOER; VLIET, 2008).

Segundo Farenhorst and Boer (2009), o conhecimento arquitetural pode ser dividido em quatro visões sobre sua utilização e importância:

- **Centrado em padrões:** O principal objetivo dessa visão é permitir que, ao utilizar padrões de projeto, seja possível capturar conhecimento especializado e aplicá-lo na arquitetura de software, necessário para saber quando e como reutilizar projetos e códigos. A comunidade de padrões ampliou seu horizonte além do padrão de projeto orientado a objetos, incluindo padrões para análise (FOWLER, 1997), projeto de arquitetura de software ((FOWLER, 2012; RICHARDS, 2015; AMPATZOGLOU et al., 2015) e no processo de desenvolvimento (KHOMH; GUÉHÉNEUC, 2018);
- **Centrado em dinamismo:** Nessa visão, o conhecimento arquitetural está inserido no próprio código-fonte do software de maneira formal. Bradbury et al. (2004) comentam que quase todas as abordagens formais de especificação para arquiteturas de software dinâmicas são baseadas em representações gráficas. As *Linguagens de*

Descrição de Arquitetura (LDA) são um exemplo de linguagens formais que podem ser usadas para representar a arquitetura de um sistema de software, definindo seus componentes, comportamentos e padrões de interação (SHAW; GARLAN, 1996). Embora nem todas as LDAs sejam adequadas para sistemas dinâmicos de tempo de execução, todas são baseadas em representações gráficas ou textuais (MEDVIDOVIC; TAYLOR, 2000).

- **Centrado em requisitos:** O conhecimento arquitetural desempenha um papel importante ao permitir a rastreabilidade dos requisitos na transição entre a fase de requisitos do sistema e a arquitetura de software. No entanto, muitas vezes as partes interessadas não são capazes de especificar requisitos inovadores com o detalhamento necessário sem terem um conhecimento mais claro sobre a solução pretendida (POHL; SIKORA, 2007). Assim, para especificar requisitos detalhados, é necessário ter conhecimento sobre as possíveis soluções, o que significa que os requisitos e a arquitetura precisam ser co-desenvolvidos.
- **Centrado em decisões:** Cada vez mais pesquisadores concordam que se deve considerar não apenas a arquitetura resultante, mas também as decisões de projeto arquitetural e o conhecimento relacionado, que representa o raciocínio por trás do resultado. Com essa visão, o conhecimento arquitetônico precisa ser gerenciado para orientar a evolução do sistema e evitar a perda do conhecimento (BOSCH, 2004).

O CA, conforme apresentado em suas quatro visões, deve apoiar a transição entre uma aplicação genérica e uma aplicação específica do conhecimento. Lago and Avgeriou (2006) definem **aplicação genérica** como uma "*forma de conhecimento por biblioteca que pode ser utilizada em diversas aplicações, independentemente do domínio*". Já a aplicação específica envolve "todas as decisões tomadas durante o processo de arquitetura de um sistema específico e as soluções arquiteturais que implementaram tais decisões".

Dessa forma, é compreensível a necessidade da gestão desse CA conforme sua estrutura e definição apresentada até aqui. A **Gestão do Conhecimento Arquitetural (GCA)**, de acordo com (OZKAYA; WALLIN; AXELSSON, 2010), pode ser definida como a representação integrada da arquitetura de software aplicada em um sistema ou família de sistemas, juntamente com as decisões arquiteturais e sua influência no ambiente de desenvolvimento.

Tang et al. (2010) destacam que uma nova tendência emergente é armazenar e

documentar o **Conhecimento Arquitetural (CA)** para sua reutilização, onde tal conhecimento compreende vários artefatos, como requisitos, padrões utilizados, decisões de projeto de arquitetura de software e sua lógica de projeto. Dessa forma, é essencial aprofundar-se no conhecimento teórico sobre o assunto e aplicar as melhores técnicas para a implantação ou melhoria da GCA em uma organização.

Durante o processo de organização do CA, foram identificadas algumas abordagens para a realização da GCA de forma sistemática:

1. **Sistemas de gestão racional:** Auxiliam no refinamento, externalização e integração do conhecimento arquitetural, bem como na gestão da lógica do projeto.
2. **Linguagens de padrões e catálogos:** Permitem a utilização e combinação do conhecimento arquitetural, que precisa ser categorizado em linguagens de padrões ou catálogos para ser rapidamente aplicado em contextos específicos.
3. **Linguagens de descrição de arquitetura:** RDAs (Registro de Decisão Arquitetural - Architecture Decision Records - ADR) permitem a utilização e combinação de conhecimento arquitetural de maneira formal, auxiliando na compreensão de sistemas de software e permitindo análises detalhadas.
4. **Codificação do modelo de projeto:** Métodos, ferramentas e técnicas foram desenvolvidos para codificar conceitos de conhecimento arquitetural, com foco nas decisões de projeto arquitetural e em conceitos relacionados, como justificativas e soluções alternativas.

Quando se trata do compartilhamento do conhecimento arquitetural, é importante fornecer suporte para o gerenciamento de vários tipos de conhecimento, atendendo tanto a aplicações genéricas quanto específicas da arquitetura de software.

Boer et al. (2007) afirmam que o CA é amplo e pode ser dividido em duas categorias: (i) o conhecimento presente no produto, refletido nos artefatos técnicos (código-fonte) e nas decisões arquiteturais documentadas, e (ii) o conhecimento do que é esperado pelo cliente (através da documentação das decisões de negócio), refletido em critérios de qualidade.

A utilização de uma estratégia híbrida, envolvendo as duas categorias, pode ser uma opção viável para a aplicação da GCA (BOER et al., 2007). Dessa forma, é possível permitir que o conhecimento arquitetural seja acessível no momento de sua necessidade, de forma ágil, tanto para a gestão de negócio (donos de produto, líderes de time e clientes)

quanto para a gestão técnica (desenvolvedores e profissionais responsáveis pela sustentação de produtos). Ou seja, o fornecimento do conhecimento arquitetural correto, para as pessoas certas, na hora certa.

2.3 Considerações Finais

A transcrição do referencial teórico a respeito do assunto aqui tratado auxilia na criação de uma base sólida para o desenvolvimento desta pesquisa. A necessidade de melhorias na Gestão do Conhecimento e seu aprimoramento na área de arquitetura de software é evidente.

De acordo com Fannoun and Kerins (2019), a natureza e o papel do conhecimento organizacional são complexos e contestados, com diferentes perspectivas sobre sua incorporação implícita na infraestrutura organizacional e nas ações dos indivíduos. Isso corrobora a necessidade de melhorias na compreensão e no aperfeiçoamento da gestão do conhecimento organizacional de maneira ágil para todos os envolvidos no processo.

No que se refere ao desenvolvimento de software, uma arquitetura bem definida e acessível é fundamental. Sua formação envolve um processo intensivo de conhecimento, com a participação de muitas partes interessadas distintas. Vliet (2008) afirma que muito conhecimento arquitetônico não se estabiliza até que seja criado um consenso sobre uma estratégia de personalização para sua gestão.

As tendências recentes, tanto na pesquisa quanto na indústria, exigem suporte aprimorado de ferramentas para arquitetos e desenvolvedores de software, a fim de gerenciar decisões de design arquitetônico e seus conceitos associados (BHAT; SHUMAIEV; MATTHES, 2017). A centralização desses conhecimentos, que geralmente se encontram pulverizados nas organizações, mostra-se promissora para alcançar os objetivos aqui traçados.

3 REVISÃO SISTEMÁTICA DA LITERATURA

Este capítulo apresenta uma Extensão de uma Revisão Sistemática da Literatura sobre *Diagnóstico da Gestão do conhecimento em organizações de software* para compreender o estado atual da Gestão do Conhecimento (GC) na indústria de software. Os resultados deste estudo fornecem uma visão geral, abrangente do estado da arte da GC, ajudando pesquisadores e profissionais a encontrar características, limitações e lacunas nas pesquisas atuais sobre o tema.

A investigação realizada permitiu uma compreensão clara de que uma das principais necessidades na indústria de software é uma *Gestão do Conhecimento Arquitetônico (GCA)* mais bem distribuída e acessível a todos os colaboradores envolvidos, direta ou indiretamente, no desenvolvimento de software. Dessa forma, a pesquisa foi fundamentada na literatura existente, proporcionando o embasamento teórico desta necessidade para continuidade da pesquisa.

3.1 Motivação

O desenvolvimento de Sistemas de Informação transcende a mera documentação formal, implicando em uma melhoria contínua em sua qualidade com base em experiências anteriores. Esse conhecimento prático não apenas mantém a eficácia do processo, mas também assegura de forma contínua a qualidade dos produtos desenvolvidos (FEHÉR; GÁBOR, 2006). A condição para qualquer iniciativa de GC organizacional é a disponibilidade do conhecimento, com membros da equipe detentores de conhecimentos específicos abertos e dispostos a compartilhá-los (BETZ; OBERWEIS; STEPHAN, 2014).

É imperativo que a organização precisa ter em mente que sua base de conhecimento deve residir internamente, não dependendo exclusivamente de seu corpo de funcionários (BETZ; OBERWEIS; STEPHAN, 2014). A definição do conhecimento organizacional e a elaboração de métricas para avaliar o gerenciamento de seus ativos com qualidade são um desafio (KULKARNI; LOUIS, 2003).

Tanto o diagnóstico quanto a gestão da criação de conhecimento organizacional visa não apenas diagnosticar, mas fornecer uma análise detalhada do conhecimento gerido pela empresa, tornando viável a utilização de ferramentas que examinam e resolvem deficiências organizacionais no contexto de criação e GC (WU; SENOO; MAGNIER-

WATANABE, 2010).

Considerando a relevância dessa temática, Maciel et al. (2018) conduziram uma Revisão Sistemática da Literatura (RSL) para compreender e caracterizar a pesquisa existente à respeito do diagnóstico da GC dentro das organizações de desenvolvimento de software. Os resultados mostraram que a GC nas organizações exerce um papel eficaz no monitoramento do desempenho e nas práticas de desenvolvimento de software. Os autores ainda exploraram diferentes abordagens relacionadas ao diagnóstico da GC, como as métricas (GOLDONI; OLIVEIRA, 2010), percepções culturais tácitas (LEVY et al., 2008) e gestão de riscos e técnicas da GC (NEVES et al., 2014a). No entanto, essa RSL foi conduzida até o ano de 2018.

Diante do dinamismo que se tem presenciado na GC, percebemos a necessidade de uma atualização dessa RSL, a fim de proporcionar uma visão mais precisa do cenário atual das prática da GC. Sendo assim, foi realizada uma extensão da RSL de Maciel et al. (2018).

A partir dos resultados, espera-se compreender o cenário atual, visando estabelecer bases sólidas para a elaboração de estratégias eficazes na aplicação da GC nas organizações. Além disso, há a necessidade de identificar as estratégias de GC mais promissoras, as quais auxiliarão na formação de novas equipes, na transmissão de conhecimento e experiências, assim como no desenvolvimento de novos produtos e na manutenção dos produtos já existentes.

3.2 Método de pesquisa

Inicialmente, foi adotado o protocolo de Maciel et al. (2018), que incluíam questões de pesquisa, critérios de inclusão e exclusão, fontes de estudo (bibliotecas digitais) e *string* de busca. Com o objetivo de ampliar ainda mais a base de pesquisa, foi incorporado duas novas fontes de busca, *Engineering Village* e *Springer*.

Para essas duas bases, não foi estabelecido limite de data nas buscas. Ademais, foi incluído dois critérios de exclusão. O primeiro descarta estudos da revisão anterior, garantindo resultados mais atualizados. O segundo critério visa eliminar pesquisas sem testes práticos em organizações de software, garantindo resultados mais fidedignos à realidade da indústria. Essas mudanças visam superar limitações da pesquisa original, contribuindo para uma compreensão mais precisa da gestão do conhecimento em organizações de desenvolvimento de software.

A extensão da RSL foi realizada em três etapas: planejamento, condução e relato dos resultados. As atividades relativas às etapas de planejamento e condução da RSL são descritas nas Subseção 3.2.1 e Subseção 3.2.2 e o relato dos resultados é apresentado na Seção 3.3.

3.2.1 Planejamento

O planejamento desta RSL se mostra essencial para garantir a transparência, reprodutibilidade e rigor metodológico da pesquisa. Esta subseção é formada pela formulação das questões de pesquisa, a definição das bases de dados a serem utilizadas, bem como os critérios de inclusão e exclusão dos estudos.

3.2.1.1 Questões de pesquisa

A Tabela 1 apresenta as duas Questões de Pesquisa (QPs) que guiaram esta RSL e suas motivações.

Tabela 1 – Questões de Pesquisa e suas Motivações.

| # | Questão de Pesquisa | Motivação |
|-----|---|---|
| QP1 | Qual o propósito de aplicar um diagnóstico de GC no contexto de organizações de software? | Identificar o propósito do estudo em aplicar algum tipo de diagnóstico para analisar a GC dentro de organizações de desenvolvimento de software. |
| QP2 | Quais são as principais características do diagnóstico? | Compreender como foram conduzidas as análises nas organizações no que diz respeito à GC, identificando as abordagens já conhecidas na literatura ou novas propostas pelos autores. Além disso, busca-se entender as características e o tipo de conhecimento analisado. |

3.2.1.2 Bases de pesquisa

As bibliotecas digitais foram selecionadas pelos seguintes critérios: (I) cobertura em conferências, journals (periódicos), livros catalogados na área da Ciência da Computação, (II) atualização regular do conteúdo, (III) disponibilidade do texto completo, (IV) qualidade nos resultados e (V) capacidade de exportar os dados encontrados. As bases que melhor preenchem tais requisitos foram: [IEEE Xplore](#), [ACM Digital Library](#), [ScienceDirect](#), [SCOPUS](#), [Engineering Village \(Compendex\)](#) e [Springer](#).

3.2.1.3 String de busca

Utilizamos a mesma *string* adotada na RSL original. As principais palavras utilizadas para formar a *string* de busca, foram divididas em três áreas: “*knowledge management*”, “*diagnosis*” and “*software engineering*”. A *string* foi aplicada em três campos de metadados das bibliotecas digitais: título (*title*), resumo (*abstract*) e palavras-chave (*keywords*).

A Tabela 2 apresenta os termos e a *string* de busca utilizada para esta pesquisa. As buscas foram realizadas em dezembro de 2022.

Tabela 2 – String de busca adotada nas bibliotecas.

| Áreas | Termos e Sinônimos utilizados | Operador |
|----------------------|---|----------|
| Knowledge Management | “knowledge management” | AND |
| Diagnosis | “diagnostic” OR “diagnose” OR “diagnosis” OR “audit” OR “assessment” OR “evaluation” | AND |
| Software Engineering | “software development” OR “software engineering” OR “software process” OR “development companies” OR “software product” OR “software quality” | |

3.2.1.4 Critérios de inclusão e exclusão

Para a seleção dos estudos, foi aplicado um único Critério de Inclusão (CI), o qual determina que: **(CI-1)** A publicação deve apresentar alguma abordagem para diagnosticar a gestão do conhecimento no contexto de organizações de software. Foram definidos também oito Critérios de Exclusão (CE): **(CE-1)** O estudo está escrita na forma de um artigo curto, resumo ou resumo estendido; **(CE-2)** O estudo não está escrito em Inglês; **(CE-3)** O estudo é uma versão mais antiga de outro estudo já considerado; **(CE-4)** O estudo não é um estudo primário, ou seja, é um estudo secundário/terciário, editorial, propaganda ou tutorial; **(CE-5)** O estudo não está disponível para leitura e download; **(CE-6)** O estudo não apresenta uma abordagem para diagnosticar a gestão de conhecimento no contexto de organizações de software; **(CE-7)** O estudo foi selecionado na RSL anterior; e **(CE-8)** O estudo não aplica a sua proposta em uma situação real.

3.2.2 Condução da RSL

A busca por estudos nesta RSL começou nas mesmas bases de pesquisa usadas por Maciel et al. (2018), porém foram considerados apenas estudos publicados desde 2018. Em seguida, foi realizada a busca em duas novas bases adicionadas (*Springer* e *Engineering Village*), sem aplicar filtro por período. Com a aplicação das *strings* de busca, foram obtidos inicialmente 633 estudos. Os processo de seleção pode ser visto na Figura 2.

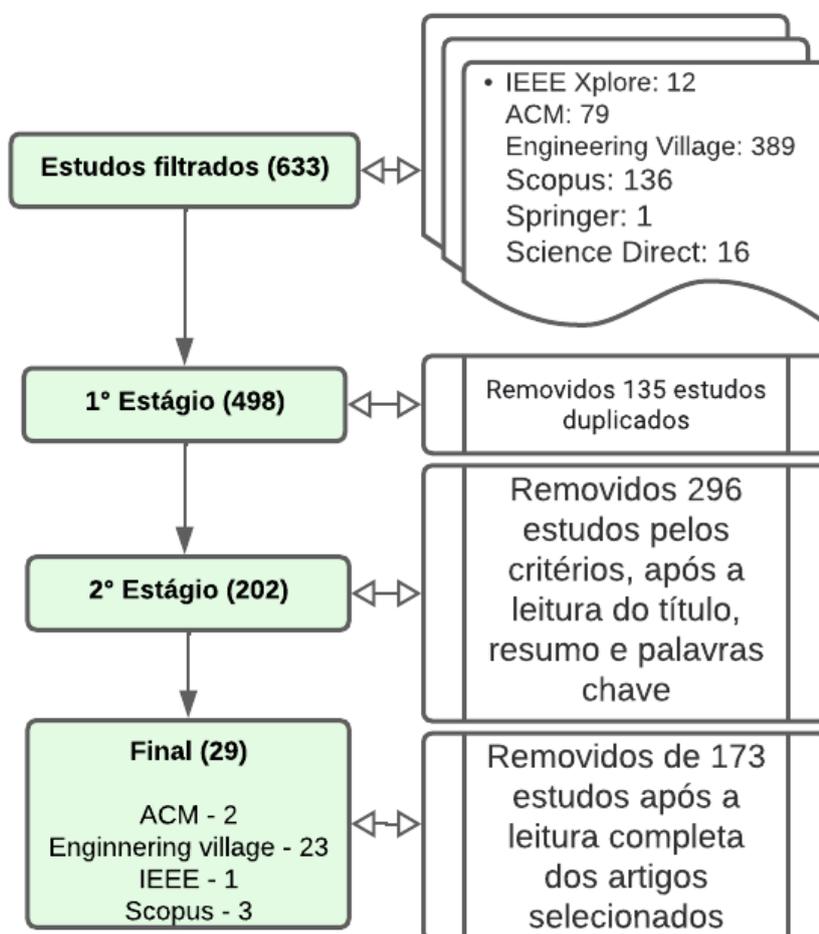


Figura 2 – Processo de busca e seleção.

A seleção dos estudos foi dividida em três estágios. No **1º estágio**, foram removidos os estudos duplicados e os artigos que também já haviam sido selecionados na revisão original, resultando em um total de 498 estudos. No **2º estágio**, foi utilizado os Critérios de Inclusão e Exclusão, examinando o título e resumo de cada estudo, resultando na seleção de 202 estudos. Na **Seleção final**, foi realizado a leitura completa dos artigos,

resultando na seleção de apenas 29 estudos, dos quais extraímos os dados. A lista dos artigos aceitos pode ser vista na Tabela 3 e na Tabela 4.

3.2.3 Estratégia de extração dos dados

Após a seleção dos estudos primários, realizamos análise de cada estudo respondendo às questões de pesquisa. Foi necessário validar o estudo contemplando o objetivo da RSL, que é diagnosticar a GC em uma organização de desenvolvimento de software e as formas como se realiza este diagnóstico para verificar sua validade e replicação em diferentes cenários. Todo o processo desta RSL (planejamento, execução e extração) foi realizada com suporte da ferramenta [Parsifal](#). O protocolo da RSL, bem como os dados extraídos, podem ser encontrados em Souza, Silva and Guedes (2024a).

3.3 Análise dos Resultados

Na presente seção, são analisados os resultados obtidos a partir da investigação das questões de pesquisa que orientaram este estudo. Primeiramente, examina-se o propósito do diagnóstico da gestão do conhecimento, identificando sua relevância para as organizações na identificação de possíveis lacunas na GC, potencialidades e direcionamentos estratégicos. Em seguida, exploram-se as características desse diagnóstico, abordando os elementos essenciais que o compõem, suas metodologias e aplicabilidade prática. A análise dos dados permite compreender como essas dimensões se inter-relacionam e contribuem para a efetividade da gestão do conhecimento, fornecendo subsídios para aprimoramento das práticas organizacionais.

3.3.1 Propósito do diagnóstico de GC (QP1)

A QP1 visa identificar do propósito de cada estudo ao aplicar algum tipo de diagnóstico para analisar a GC dentro de organizações de desenvolvimento de software.

Dentre os trabalhos selecionados, observa-se que, de forma geral, eles apresentam objetivos bastante específicos para diagnosticar a GC nas organizações. Os estudos analisados descrevem o uso de diversas abordagens no desenvolvimento da GC. Tais abordagens serão discutidas nesta subseção.

Visando investigar a Gestão do Conhecimento do Cliente (CKM - *Customer Knowledge Management*), Khosravi et al. (2022), em seu trabalho, afirmam que a CKM se divide em fatores organizacionais, humanos e tecnológicos. Os fatores organizacionais são úteis no diagnóstico da GC para compreender o envolvimento do cliente, a cultura centrada no cliente e o desenvolvimento de estratégias em CKM. Os fatores humanos constituem-se em seus desenvolvedores, que podem ser disponibilizados ou restringidos pelas organizações. Já os fatores tecnológicos, conhecidos como CRM (*Customer Relationship Management* - Gestão de Relacionamento com o Cliente), são responsáveis por absorver e armazenar conhecimento com o auxílio de mapas de conhecimento organizacionais.

Outros estudos utilizaram abordagens de Aprendizagem Organizacional (OL) para diagnosticar a GC (FANNOUN; KERINS, 2019; GASSTON; HALLORAN, 1999; ÖZEN et al., 2013b). A OL mostrou-se promissora para compreender a aplicação De GC na indústria de software. Gasston and Halloran (1999) enfatizam que, devido ao desenvolvimento de software ser baseado no trabalho em equipe, compreender como a gestão da informação ocorre na organização é crucial. Assim, a OL é útil ao permitir a análise de conhecimentos tácitos, preconceitos e experiências geradas pela empresa.

Percebe-se que os trabalhos propostos por Joel-Edgar and Gopsill (2018), Tuzovsky, Yampolsky and Chirikov (2005), Lethbridge (2000), Dawson (2009), Rabelo et al. (2015), Spoladore and Pessot (2022), Kulkarni and Louis (2003) apresentam uma abordagem ampla no que diz respeito à Cultura organizacional (OC - *Organizational Culture*).

Com destaque, Rabelo et al. (2015) afirmam que a cultura organizacional afeta a forma como os membros da organização aprendem, adquirem e compartilham conhecimento. Dentro da GC, é de fundamental importância compreender como ocorre a geração desse conhecimento, assim como o compartilhamento e aplicação do mesmo na melhoria da qualidade do software desenvolvido. Neste sentido, os autores investigaram a relação entre o ciclo de Gestão do Conhecimento (GC) usando o modelo SECI (*Socialization, Externalization, Combination, and Internalization model*) e a cultura organizacional por meio do modelo CVF (*Competing Values Framework*). Os resultados diferem das propostas na literatura, como a falta da relação entre internalização do conhecimento e tipo de cultura hierárquica. No estudo de caso, a maioria das práticas de GC estava ligada à internalização do conhecimento, e a cultura predominante era do perfil cultural de mercado. Essas descobertas ressaltam a importância de considerar as nuances específicas de cada organização ao relacionar práticas de GC com sua cultura.

Tuzovsky, Yampolsky and Chirikov (2005) afirmam que a GC é um processo que necessita de adaptação da estrutura que envolve o desenvolvimento de software dentro da organização as quais envolvem sua cultura (OC - Organization Culture), procedimentos gerenciais e ambiente de software. Os autores afirmam que, usando uma ontologia, é possível tornar o conhecimento da empresa mais estruturado para criar “espaço de conhecimento empresarial” e sistematizar o trabalho com conhecimento.

Ao analisar a OC, no contexto de desenvolvimento de software, Lethbridge (2000) constatou-se que, a interface do usuário tinha que ser capaz de funcionar efetivamente com diferentes visões do conhecimento demonstrado aos usuários, e permitir que esse conhecimento fosse editado a partir de qualquer uma das visualizações.

Nos resultados desta RSL, se mostra notório vários trabalhos visando compreender a transferência de conhecimento (KT - *Knowledge Transfer*) (NAMBU; SUEHIRO; YAMAGUCHI, 2019; WAHEED et al., 2019; BETZ; OBERWEIS; STEPHAN, 2014; SANGAIAH et al., 2017; DEMIRSOY; PETERSEN, 2018; TODOROVA; MILLS, 2018; BADAMPUDI; WOHLIN; GORSCHKEK, 2019). Waheed et al. (2019) e Sangaiah et al. (2017) destacam que a necessidade de transferência de conhecimento se evidencia principalmente devido a muitas empresas adotarem equipes compostas de profissionais por vezes distantes geograficamente, devido à possibilidade do trabalho remoto. Betz, Oberweis and Stephan (2014) acrescentam ainda que um dos principais fatores de sucesso no campo da tecnologia da informação é maximizar a quantidade da transferência de conhecimento.

Já com relação à investigação sobre a gestão do Conhecimento Arquitetural (AK - *Architectural Knowledge*) nas organizações, diversos estudos destacam que este é um fator de muita importância, principalmente no processo de captura e disseminação do conhecimento (PEDRAZA-GARCIA; ASTUDILLO; CORREAL, 2016; BORREGO et al., 2019; ELORANTA; KOSKIMIES, 2014; DAKHLI; CHOUIKHA, 2009). Borrego et al. (2019) definem gestão do conhecimento arquitetural (AKM - *Architectural Knowledge Management*) como a disciplina de criação/captura, compartilhamento/divulgação e aquisição/aplicação no desenvolvimento de um software. Com base nesses fatores, pretende-se compreender a área de documentação baseada em artefatos que é utilizado como suporte no desenvolvimento de software para compartilhar AK.

Houve também uma preocupação em relacionar a arquitetura de software com metodologias ágeis de desenvolvimento. Segundo Eloranta and Koskimies (2014), o AKM deve ser integrado ao desenvolvimento ágil, não sendo necessariamente um peso para o processo, mas sim uma prática que pode apoiar os métodos ágeis a longo prazo, melho-

rando a comunicação entre as partes interessadas.

Para mitigar esses impactos nas organizações, Eloranta and Koskimies (2014) destacaram a importância de integrar o conhecimento arquitetural (AK) em um único sistema de AKM. Isso permite explorar a avaliação do ambiente de arquitetura de software de maneira leve e distribuída, facilitando a codificação do conhecimento de forma sincronizada com o desenvolvimento de software.

A partir de uma perspectiva mais abrangente, Alguns autores discutiram sobre a Melhoria de Processo de Software (SPI - *Software Process Improvement*) (IVARSSON; GORSCHER, 2012; DESOUZA; AWAZU; WAN, 2006; MITCHELL; SEAMAN, 2016; FEHÉR; GÁBOR, 2006; JURADO et al., 2019). O SPI visa aprimorar continuamente os processos de desenvolvimento de software, buscando reduzir custos, melhorar o cronograma e aumentar a qualidade do produto final. Nesse contexto, a GC pode ser vista como um complemento útil para o SPI no desenvolvimento de software (MITCHELL; SEAMAN, 2016). A captura de conhecimento de projetos anteriores para melhoria contínua é uma técnica relevante no contexto da GC para o SPI.

Outros propósitos/aspectos foram abordados apenas uma vez nos trabalhos selecionados. Assim como o CKM (*Customer Knowledge Management*), citado anteriormente, temos a transformação de conhecimento individual em conhecimento multi-organizacional (MONTONI et al., 2004) e o Suporte em Engenharia de Software (SES - *Software Engineering Support*) (CORBIN; DUNBAR; ZHU, 2007), os quais são apresentados nos próximos parágrafos.

Montoni et al. (2004) afirmam que muitas tarefas importantes relacionadas ao desenvolvimento de software dependem de indivíduos dentro da organização, pois a execução de processos de software requer a GC especializada dos membros da organização que cooperam para atingir um objetivo de negócio. Dessa forma, vemos que membros habilidosos da organização devem partilhar os seus conhecimentos para estabelecer um ciclo de aprendizagem organizacional e transformar este conhecimento intelectual em conhecimento multi-organizacional através da GC.

Finalmente, Corbin, Dunbar and Zhu (2007) afirmam que a GC no Suporte à Engenharia de Software (SES) visa melhorar a estabilidade da transição do sistema de software dentro da organização, e dessa forma garantir impacto mínimo no desempenho operacional e promover a rápida adaptação de novas tecnologias às operações do sistema.

3.3.2 Características do diagnóstico (QP2)

Nesta questão de pesquisa, são analisadas as principais características dos diagnósticos realizados sobre GC nas organizações. Isso engloba o formato que foi aplicado, quais itens foram investigados, se houve a utilização de alguma ferramenta para auxiliar na coleta de dados, qual tipo de conhecimento foi abordado na pesquisa e, principalmente, quais abordagens já conhecidas na literatura foram utilizadas ou quais novas abordagens foram propostas.

Para isso, foram investigados primeiramente os métodos aplicados para levantamento de informações sobre GC dentro das organizações. Os resultados indicam que, em sua maioria, foram empregadas somente entrevistas (JOEL-EDGAR; GOPSILL, 2018; WAHEED et al., 2019; BETZ; OBERWEIS; STEPHAN, 2014; DAKHLI; CHOUIKHA, 2009; BADAMPUDI; WOHLIN; GORSCHKEK, 2019; JURADO et al., 2019), somente questionários (NAMBU; SUEHIRO; YAMAGUCHI, 2019; KULKARNI; LOUIS, 2003; TUZOVSKY; YAMPOLSKY; CHIRIKOV, 2005; DESOUZA; AWAZU; WAN, 2006; TODOROVA; MILLS, 2018; LETHBRIDGE, 2000; BORREGO et al., 2019; SANGAIAH et al., 2017) ou entrevistas e questionários (GASSTON; HALLORAN, 1999; FANNOUN; KERINS, 2019; ÖZEN et al., 2013b; IVARSSON; GORSCHKEK, 2012; ELORANTA; KOSKIMIES, 2014; MITCHELL; SEAMAN, 2016; DAWSON, 2009; KHOSRAVI et al., 2022). Alguns outros estudos, em menor quantidade, utilizaram abordagens como observação do ambiente de trabalho da organização e o comportamento dos desenvolvedores (MONTONI et al., 2004; FEHÉR; GÁBOR, 2006), assim como entrevista e a validação de documentação pré-existente (DEMIRSOY; PETERSEN, 2018), entrevista e observação (CORBIN; DUNBAR; ZHU, 2007; RABELO et al., 2015; PEDRAZA-GARCIA; ASTUDILLO; CORREAL, 2016) e por fim questionário e observação (JOEL-EDGAR; GOPSILL, 2018). A seguir esses estudos serão discutidos.

Através de um questionário do tipo Survey, Desouza, Awazu and Wan (2006) realizou um estudo de campo nos escritórios locais de uma organização internacional de consultoria em tecnologia da informação. Como resultado os autores puderam compreender o comportamento dos consumidores da GC os quais se mostraram mais propensos a consumir produtos de conhecimento mais simples e claros de se compreender ao invés de produtos de conhecimento complexos.

Fannoun and Kerins (2019), por sua vez, conduziram uma análise qualitativa da GC na organização. Os autores apresentaram uma visão sobre o desenvolvimento do co-

nhecimento contextual, individual e coletivo. Foi possível também notar que a abordagem que estavam propondo para aplicação da GC tinha o potencial de promover a resiliência do conhecimento e melhorar as práticas de compartilhamento desse conhecimento em pequenas empresas.

Joel-Edgar and Gopsill (2018) realizaram sete entrevistas formais documentadas, gravadas e transcritas que proporcionaram a compreensão de como o conhecimento é acumulado atualmente dentro da organização. Os autores também identificaram a dinâmica social da busca do conhecimento e dessa forma avaliaram como a empresa realiza a GC.

Enquanto isso, Spoladore and Pessot (2022), por meio de observações, entrevistas e análise de notas de campo durante tarefas de desenvolvimento, identificaram três metodologias ágeis de engenharia de ontologias (OEMs - *Ontology Engineering Methodologies*) que apoiam a transformação digital da GC nas empresas.

É possível observar, na Tabela 3, que apenas três trabalhos adotaram o uso de ferramentas conhecidas na literatura para auxiliar no diagnóstico na organização. Desouza, Awazu and Wan (2006) aplicaram uma Web Survey, mas não deixaram explícito qual ferramenta foi usada nesta survey, enquanto Joel-Edgar and Gopsill (2018), Waheed et al. (2019) utilizaram a ferramenta NVivo.

Sendo as organizações de software apontadas como um dos setores mais intensivos em conhecimento, Desouza, Awazu and Wan (2006) realizaram um estudo de campo nos escritórios locais de uma organização internacional de consultoria em tecnologia da informação.

Joel-Edgar and Gopsill (2018), utilizaram uma abordagem de teoria fundamentada para elicitare temas e gerar hipóteses. Nos resultados dessa abordagem os entrevistados descreveram como e por que os indivíduos buscavam conhecimento. Permitindo dessa forma compreender como era realizada a GC e como o conhecimento gerado era transmitido para novos funcionários e para funcionários em transição entre setores na organização.

Em relação aos modelos e métodos empregados em nossa pesquisa, 13 trabalhos se basearam em um modelo, técnica ou processo já existente na literatura. Dentre as abordagens temos o UTEM (BORREGO et al., 2019), Modelo SECI (RABELO et al., 2015), Modelo OLEC (GASSTON; HALLORAN, 1999), PSF (IVARSSON; GORSCHKE, 2012), KMM (JURADO et al., 2019), CKM (KHOSRAVI et al., 2022), CMM (KULKARNI; LOUIS, 2003), CTA (LETHBRIDGE, 2000), K-Flow e K-Map (MITCHELL; SEAMAN, 2016), AIOLOS (ÖZEN et al., 2013b), GTM (PEDRAZA-GARCIA; AS-

TUDILLO; CORREAL, 2016), DEMATEL, TOPSIS e ELECTRE (SANGAIAH et al., 2017) e OEM (SPOLADORE; PESSOT, 2022). As abordagens identificadas são apresentadas brevemente abaixo.

O modelo de Mídia Eletrônica Textual Não Estruturada (UTEM - *Unstructured Textual Electronic Media*), consiste em um conjunto de *logs* originados de mensageiros instantâneos, e-mails, fóruns, etc., que são os meios preferidos empregados no AGSD (*Agile Global Software Development* - desenvolvimento ágil de software global) (BORREGO et al., 2019). Borrego et al. (2019), baseado na marcação de interações pessoais usando UTEM em tempo real, propõe o conceito de condensação do AK, aplicado por meio de um protótipo pelo autor e, dessa forma, busca classificar esse conhecimento adquirido, facilitando sua posterior recuperação.

O modelo SECI (RABELO et al., 2015), foca na criação, exploração e manutenção do conhecimento por meio da socialização, que é o processo de compartilhamento. Isso envolve o compartilhamento de conhecimento tácito entre pessoas por meio de observação, cooperação ou imitação de comportamento.

Na pesquisa conduzida por Gasston and Halloran (1999), os autores observaram que o modelo OLEC (*Organisational Learning Evaluation Cycle*) tem a capacidade de identificar não apenas melhorias no desempenho da Gestão do Conhecimento (GC), mas também alterações nos processos organizacionais. Essas alterações são refletidas por mudanças no contexto formativo da gestão do conhecimento.

O modelo PSF (*Practice Selection Framework*), segundo Ivarsson and Gorschek (2012), consiste em um conjunto de práticas utilizadas na organização em que todos os desenvolvedores tem acesso. As experiências coletadas pelo modelo PSF, fornecem suporte à decisão de forma iterativa, sugerindo quais melhores práticas e profissionais a se utilizar no desenvolvimento de software através da GC.

Ivarsson and Gorschek (2012) afirma que em comparação com outras abordagens de gestão do conhecimento para mapear competências e pessoas com conhecimento específico dentro da organização, o PSF também fornece contexto e valor às experiências que as pessoas têm. Isto permite que seja mais fácil identificar pessoas que têm experiências em determinados contextos e problemas enfrentados pela equipe de desenvolvimento de software.

Jurado et al. (2019) estabelecem tipos de GC com base no grau de maturidade organizacional, utilizando o KMM (*Knowledge Management Model*). O modelo proposto inclui dois componentes principais: o conjunto de processos da GC e o contexto de

aplicação. Os autores sugerem que é possível adaptar o uso da GC em organizações de software por meio de uma abordagem flexível nos processos de geração e conservação do conhecimento utilizando o KMM.

Khosravi et al. (2022) consultaram especialistas em desenvolvimento de SE (Software Empresarial), através de um questionário com uma lista de facilitadores conforme o modelo proposto pelo CKM e suas definições. Com a visão do cliente, através do CKM, foi possível compreender quais os fatores de impacto são observados na fase de desenvolvimento de SE, e a coleta desse conhecimento para a geração de novos conhecimentos a serem catalogados, ajudando na qualidade do produto entregue e na satisfação do cliente.

Kulkarni and Louis (2003) demonstram que o CMM (*Customer Management Maturity*) mensura o nível de conformidade com um conjunto padrão de processos prescritos por uma organização para o desenvolvimento de software, definindo assim a maturidade organizacional. Nesse contexto, Kulkarni and Louis (2003) realizaram uma avaliação da maturidade da GC por meio de uma amostra de várias empresas. A intenção foi avaliar longitudinalmente os benefícios associados ao aumento da maturidade na GC.

CTAs (*Creative Task Assistants*) são modelos aplicados a ferramentas que auxiliam as pessoas na execução de tarefas complexas e criativas, frequentemente associadas a problemas complexos. Em seu trabalho, Lethbridge (2000) aborda a avaliação de um CTA denominado CODE4, cujo objetivo é facilitar a criação de bases de conhecimento conceituais para utilização como GC na organização. Segundo Lethbridge (2000), CODE4 foi estruturado para que pessoas especialistas em algum domínio possam manipular estruturas de GC.

Em seu trabalho, Mitchell and Seaman (2016) empregaram o K-flow (*Knowledge Flow*), que se refere a um movimento dinâmico de conhecimento (como entre indivíduos ou organizações). Durante a sua execução, os autores utilizaram um mapa de conhecimento (K-Map - *Knowledge Map*) para identificar possíveis obstáculos no fluxo do conhecimento dentro da organização, possibilitando a localização de entraves ao K-Flow. Os resultados do estudo evidenciaram a eficácia do K-Map para melhoria de processos de software, GC e o estado da GC no contexto de Engenharia de Software.

Özen et al. (2013b) afirmam que o AIOLoS (*Assessing Organization Learning of Software Development Organizations*) é um modelo projetado para avaliar as capacidades de OL e GC nas organizações. AIOLoS abrange três áreas principais de processos que mapeiam os principais objetivos de uma organização na aprendizagem de software: adquirir, utilizar e transmitir conhecimento. Özen et al. (2013b) utilizaram o modelo AiOLoS para

se desenvolver novas medidas na avaliação das capacidades de OL das organizações.

Pedraza-Garcia, Astudillo and Correal (2016) definem o modelo GTM (*Grounded Theory Method*) como um método proposto para descobrir, desenvolver e verificar uma teoria baseada em um fenômeno, por meio de coleta e análise de dados. Os autores Pedraza-Garcia, Astudillo and Correal (2016) utilizaram o processo do GTM para compreender o ciclo de vida da GC, incorporando diversas técnicas, tais como um modelo de anotação para aquisição de conhecimento, anotação para mapeamento de decisões para recuperação de conhecimento e cenários de geração e uso para compartilhamento de conhecimento.

Sangaiah et al. (2017), adotaram a abordagem *fuzzy* DEMATEL para determinar os pesos de prioridade dos critérios visando uma transferência efetiva de conhecimento. Esses pesos foram, então, aplicados nos modelos TOPSIS e ELECTRE para identificar a classificação e a significância dos atributos, utilizando a abordagem *fuzzy* TOPSIS na avaliação de alternativas, bem como para determinar a classificação final das alternativas com o ELECTRE *fuzzy*. A compreensão da GC com essa abordagem foi obtida por meio de cálculos probabilísticos utilizando os modelos mencionados.

OEMs (*Ontology Engineering Methodologies*) são metodologias que visam fornecer aos engenheiros de ontologia um conjunto de diretrizes para passar do conhecimento informal (documentos, dados, *know-how*, práticas, etc.) para um conhecimento formal (ou seja, uma representação baseada em lógica) que pode ser utilizado na GC. Spoladore and Pessot (2022) investigaram as características das OEMs ágeis que permitem às empresas explorar plenamente o seu potencial numa transformação digital dos processos de GC e consequente aprendizagem organizacional. Como resultados, os autores evidenciaram as seguintes características, clareza e simplicidade, adaptabilidade e flexibilidade, trabalho em equipe e cooperação, esforço, padrões de desenvolvimento e suporte à inovação.

Embora tenha sido constatado, na literatura, a existência de modelos, métodos e frameworks para o diagnóstico de GC no contexto de sistemas de informação, percebe-se que a forma de realizar o diagnóstico e compreender a GC ainda não está bem consolidada. Atualmente, não há, por exemplo, a utilização de uma mesma metodologia em mais de um estudo. Nota-se também que, nos estudos abordados em nossa RSL, são propostas novas abordagens para diagnosticar a GC, como evidenciado em (DAWSON, 2009; DEMIRSOY; PETERSEN, 2018; MONTONI et al., 2004). Acredita-se que esse cenário se deve ao fato de que não há um consenso, especialmente no que diz respeito a estabelecer

um padrão para a GC e como ela é aplicada nas organizações.

Ainda em relação as abordagens realizadas para o diagnóstico da GC nos estudos selecionados, foi realizado uma análise dos itens investigados dentro das organizações para compreender o estado da GC. Dentre eles temos a cultura organizacional (OC) (BADAMPUDI; WOHLIN; GORSCHKE, 2019; DAWSON, 2009; RABELO et al., 2015; DEMIRSOY; PETERSEN, 2018), aprendizado organizacional (OL) (FANNOUN; KERINS, 2019; GASSTON; HALLORAN, 1999; IVARSSON; GORSCHKE, 2012; MONTONI et al., 2004; ÖZEN et al., 2013b; WAHEED et al., 2019), Conhecimento arquitetural (AK) (BORREGO et al., 2019; DAKHLI; CHOUIKHA, 2009; MITCHELL; SEAMAN, 2016; PEDRAZA-GARCIA; ASTUDILLO; CORREAL, 2016), transferência de conhecimento (KT) (DESOUZA; AWAZU; WAN, 2006; ELORANTA; KOSKIMIES, 2014; JOEL-EDGAR; GOPSILL, 2018; SANGAIAH et al., 2017; TODOROVA; MILLS, 2018), Práticas de GC na organização (BETZ; OBERWEIS; STEPHAN, 2014; CORBIN; DUNBAR; ZHU, 2007; FEHÉR; GÁBOR, 2006; JURADO et al., 2019; TUZOVSKY; YAMPOLSKY; CHIRIKOV, 2005; NAMBU; SUEHIRO; YAMAGUCHI, 2019; SPOLADORE; PESSOT, 2022), usabilidade, atratividade e contribuição da GC (LETHBRIDGE, 2000), Gestão do conhecimento do cliente (CKM) (KHOSRAVI et al., 2022) e Maturidade da Gestão do Conhecimento (KMM) (KULKARNI; LOUIS, 2003). Muitos estão relacionados às respostas em RQ1.

Finalmente, em relação ao tipo de conhecimento gerenciado, conclui-se que ele pode ser dividido em tácito e explícito. O conhecimento tácito tem como característica a dificuldade de ser articulado na linguagem formal, todavia, é um tipo de conhecimento mais importante por ser pessoal e incorporado à experiência (LETHBRIDGE, 2000). Por outro lado, segundo Luchesi (2012) o conhecimento explícito pode ser articulado na linguagem formal com maior facilidade, por meio de especificações e expressões, e, por isso, pode ser facilmente transmitido, sistematizado e comunicado. Foi analisado também a forma como a GC tem sido investigada nos estudos, considerando o tipo de conhecimento (tácito (ÖZEN et al., 2013b) ou explícito (JOEL-EDGAR; GOPSILL, 2018; GASSTON; HALLORAN, 1999; TUZOVSKY; YAMPOLSKY; CHIRIKOV, 2005; DESOUZA; AWAZU; WAN, 2006)), a maioria dos estudos aborda ambos os tipos (KHOSRAVI et al., 2022; FANNOUN; KERINS, 2019; NAMBU; SUEHIRO; YAMAGUCHI, 2019; MONTONI et al., 2004; KULKARNI; LOUIS, 2003; PEDRAZA-GARCIA; ASTUDILLO; CORREAL, 2016; IVARSSON; GORSCHKE, 2012; LETHBRIDGE, 2000; BORREGO et al., 2019; ELORANTA; KOSKIMIES, 2014; WAHEED et al., 2019; MIT-

CHELL; SEAMAN, 2016; BETZ; OBERWEIS; STEPHAN, 2014; CORBIN; DUNBAR; ZHU, 2007; FEHÉR; GÁBOR, 2006; DAKHLI; CHOUIKHA, 2009; DAWSON, 2009; SANGAIAH et al., 2017; BADAMPUDI; WOHLIN; GORSCHKEK, 2019; RABELO et al., 2015; DEMIRSOY; PETERSEN, 2018; JURADO et al., 2019; SPOLADORE; PES-SOT, 2022; TODOROVA; MILLS, 2018).

3.4 Discussões

O objetivo deste trabalho é apresentar de que forma tem sido abordado o diagnóstico da gestão do conhecimento em organizações de desenvolvimento de software. Os estudos encontrados destacam diretamente a importância de entender o funcionamento da GC e como o conhecimento tem sido compartilhado entre as equipes.

Conforme os dados coletados em nossa pesquisa, encontramos relatos de que, em algumas situações, as equipes podem estar distantes geograficamente (SANGAIAH et al., 2017; WAHEED et al., 2019), possuindo diferentes culturas e percepções na comunicação. Isso afeta a maneira como o sucesso organizacional pode ser alcançado por meio da GC. A perspectiva de aprimoramento na qualidade do software desenvolvido mostra-se promissora, com a GC sendo o foco da organização. Notamos também que a escassez de modelos, técnicas ou ferramentas conhecidos na literatura destaca um ponto crucial para análise. Isso levanta questões sobre se os modelos e técnicas existentes atendem adequadamente à GC, se as ferramentas disponíveis apresentam falhas ou se há um problema relacionado ao nível de divulgação desses modelos e ferramentas, de modo que não sejam amplamente conhecidos pelas organizações.

Percebemos que a gestão do conhecimento arquitetural (PEDRAZA-GARCIA; ASTUDILLO; CORREAL, 2016; BORREGO et al., 2019; ELORANTA; KOSKIMIES, 2014; DAKHLI; CHOUIKHA, 2009), se mostra promissora devido à sua relevância na organização e ainda com poucos estudos a seu respeito quando se trata de uma abordagem com foco em GC.

Nesse sentido, Weinreich and Groher (2016) conduziram uma RSL reportando que existem muitas abordagens que focam ou apoiam a captura do conhecimento arquitetural. A captura desse conhecimento, normalmente é baseada na web, com conjuntos de ferramentas SAKM (*Software architecture knowledge management*) dedicadas e soluções baseadas em UML, se utilizando de metamodelos que descrevem as informações necessárias ou em estruturas de diagramas mostrando o conhecimento capturado de uma forma

mais clara e acessível à todos. Corroborando, Eloranta and Koskimies (2014) afirmam que a codificação do conhecimento arquitetural deve ser integrada ao processo ágil, permitindo que o conhecimento arquitetural seja desenvolvido como um efeito colateral das atividades que criam o conhecimento. Assim, percebemos a importância da integração entre a Arquitetura de Software e o desenvolvimento Ágil, e como será realizada a gestão desse conhecimento produzido e compartilhado. Sob a ótica das metodologias ágeis, observamos que uma boa parte dos estudos encontrados está neste contexto. No entanto, conforme podemos notar na Tabela 3, nenhuma dessas pesquisas abordou diretamente esse tema como um item de conhecimento a ser investigado.

Identificamos outros trabalhos na literatura corroborando com nossas evidências. Cabral, Ribeiro and Noll (2014) discutem a influência dos fatores humanos e sociais nas equipes e entre a equipe e os clientes. Os autores abordam os fatores humanos e sociais que influenciam o desenvolvimento de software e a geração de conhecimento, incluindo os métodos ágeis aplicados por profissionais e pesquisadores experientes. Por sua vez, Ouriques et al. (2019) destacam que enquanto o desenvolvimento de software tradicional se concentra principalmente no conhecimento explícito, no contexto ágil as atividades de GC focam principalmente no conhecimento tácito. Isso demonstra que o ambiente ágil tem uma influência direta na gestão do conhecimento, dependendo do tipo de conhecimento abordado na criação de novos projetos.

É notório que um dos pontos mais investigados envolvem aprendizado e cultura organizacional (JOEL-EDGAR; GOPSILL, 2018; KULKARNI; LOUIS, 2003; TUZOVSKY; YAMPOLSKY; CHIRIKOV, 2005; LETHBRIDGE, 2000; DAWSON, 2009; RABELO et al., 2015; SPOLADORE; PESSOT, 2022) para melhor compreensão da gestão do conhecimento dentro das organizações. Dawson (2009) afirma que, ao se impor o uso de um sistema sem a realização da GC do ambiente organizacional e uma devida transferência desse novo conhecimento proposto de forma adequada, os usuários imediatamente reagiram negativamente à mudança cultural completa necessária para operar o sistema.

Outro ponto interessante refere-se abordagens encontradas em apenas um trabalho, como CKM (KHOSRAVI et al., 2022), a transformação de conhecimento individual em conhecimento multi-organizacional (MONTONI et al., 2004) e Suporte em Engenharia de Software (SES) (CORBIN; DUNBAR; ZHU, 2007). Essas abordagens são relevantes para a GC em áreas específicas da organização.

Ao compararmos nossos resultados com o estudo original proposto por Maciel

et al. (2018), observamos aspectos interessantes relacionados aos itens investigados. Na versão original, Maciel et al. (2018) identificaram 10 tipos de itens de conhecimento, como fontes diversas de conhecimento e técnicas em GC. Seis artigos foram classificados nessa categoria, evidenciando uma abordagem variada e não linear do conhecimento organizacional naquele período. Em nosso trabalho, evidenciamos oito itens específicos relacionados aos itens de conhecimento investigados na GC. Dois itens se destacaram com maior ênfase: Cultura Organizacional e Transferência do Conhecimento, com sete artigos dedicados a cada um. Destaca-se que 37,5% desses artigos foram publicados após o ano de 2018, indicando uma continuidade e relevância desses temas no cenário contemporâneo da GC em organizações de software. Uma concordância entre os dois estudos é a escassez de ferramentas conhecidas e utilizadas na investigação. Em ambos os estudos, apenas duas ferramentas foram identificadas. Na pesquisa atual, encontramos estudos relacionados diretamente ao diagnóstico ou auditoria da GC nas organizações, de forma prática e validada.

3.5 Ameaças à Validade

A seguir, elencamos as principais ameaças à validade deste estudo.

Ao expandir as bases de busca sem restrições de ano, não identificamos que isso tenha representado uma ameaça à validade da pesquisa. Isso se deve ao fato de que a pesquisa original não estabeleceu limites temporais, e os resultados obtidos estão em consonância com os achados anteriores, revelando os itens de GC que têm sido aplicados nas organizações de software ao longo do tempo.

Outra consideração está relacionada à introdução de um novo critério de exclusão. Apesar da adição, constatamos que apenas seis artigos foram excluídos com base nesse critério. Entretanto, ao analisarmos esses artigos excluídos, observamos que teriam sido eliminados por outros critérios também, não afetando de forma significativa os resultados encontrados neste estudo.

Para mitigar a subjetividade, os resultados de cada revisor foram comparados para detectar possíveis vieses na avaliação. Além disso, como fator de exclusão, incluímos estudos realizados de forma empírica sem a realização de testes reais. Uma outra limitação foi a restrição aos termos de pesquisa utilizados nas bases eletrônicas na pesquisa anterior. Para contornar essa limitação, adicionamos novas bases à pesquisa e ajustamos as strings de busca para otimizá-las de acordo com as particularidades de cada base.

Ressaltamos que este estudo adotou um protocolo robusto (KITCHENHAM; CHARTERS, 2007) para mitigar ameaças à validade. Três pesquisadores conduziram todas as fases, alcançando resultados consistentes após discutirem dúvidas em reuniões de grupo. No entanto, reconhecemos algumas limitações nesta revisão. A seleção de estudos e extração de dados basearam-se no protocolo adotado no estudo anterior, o que pode introduzir alguma subjetividade nos resultados.

3.6 Conclusões e trabalhos futuros

Realizamos uma atualização e evolução de uma Revisão Sistemática da Literatura, sintetizando as pesquisas existentes sobre a GC aplicada em organizações de desenvolvimento de software. Analisamos 29 estudos, e observamos que apenas três deles utilizam ferramentas reconhecidas na literatura para análise dos dados. Além disso, constatamos que, em 13 estudos, foram empregados modelos e métodos conhecidos na literatura. Nos demais trabalhos, foram utilizados métodos próprios para a análise, indicando a ausência de ferramentas e modelos amplamente reconhecidos para a GC.

Outro aspecto relevante é que 62% dos trabalhos abordam como as organizações lidam com a GC no desenvolvimento de software, sem um foco específico em áreas como testes, qualidade de software, projeto, entre outras. A GC está bem inserida no contexto de metodologias ágeis, que priorizam entregas de qualidade com eficiência, evidenciando a importância da transmissão de conhecimento e experiência entre os membros das equipes. Por outro lado, identificamos que a GC relacionada às arquiteturas de software possui aplicação limitada. Este tema se mostra promissor para trabalhos futuros, especialmente em ambientes de metodologia ágil, explorando como o conhecimento sobre arquiteturas de software é aprendido, registrado, transmitido e compartilhado, dada a sua importância na organização.

Em trabalhos futuros, sugerimos investigações mais aprofundadas na área de GC em arquiteturas de software, assim como a avaliação e desenvolvimento de ferramentas específicas para análise de dados nesse contexto. Essas iniciativas podem contribuir para uma gestão mais eficaz do conhecimento nas organizações de desenvolvimento de software. Além disso, pretendemos realizar uma Revisão da Literatura Cinza, que incluirá a análise de informações disponíveis pela indústria de software em *blogs*, *white papers*, relatórios técnicos, fóruns de discussão, sites de perguntas e respostas, dentre outros.

Tabela 3 – Abordagens identificadas na RSL original de Maciel et al. (2018).

| Artigo | Ano | Item do conhecimento investigado | Mét. Ap. | Mod. | Fer. |
|--------------------------------------|------|---|----------------|--------|------|
| (AURUM; DANESHGAR; WARD, 2008) | 2008 | KM practices | EN | NO | Não |
| (CHOUSEINOGLOU et al., 2013) | 2013 | OL - <i>Organization Learning</i> | QT, EN | AiOLoS | Não |
| (DORAIRAJ; NOBLE; MALIK, 2012) | 2012 | KM in Distributed Software Development | EN | NO | Não |
| (GHOBADI; MATHIASSEN, 2017) | 2018 | Knowledge sharing, Risk Management Model | EN | NO | Não |
| (GOLDONI; OLIVEIRA, 2010) | 2010 | Knowledge management metrics | EN | NO | Não |
| (GOPAL et al., 2018) | 2018 | Teams knowledge transfer, Global Software Development (GSD) | QT | NO | Não |
| (HANSEN; KAUTZ, 2004) | 2004 | K-flow (<i>Knowledge flow</i>) | EN, GV | K-map | Não |
| (IULIANA, 2009) | 2009 | Knowledge sources, tools, techniques, methods applied in the organization, KM activities, tacit knowledge and rate the level of effort invested on KM | EN | NO | Não |
| (IVARSSON, 2012) | 2012 | Experience capture | QT | NO | Não |
| (KHOSRAVI; NILASHI et al., 2018) | 2018 | CKM - <i>Customer Knowledge Management</i> | QT | NO | Não |
| (LAGERBERG et al., 2013) | 2013 | OL | QT, EN | AiOLoS | SM |
| (LEHNER, 2009) | 2009 | KM sucess and KM service | QT | NO | Não |
| (LEVY et al., 2008) | 2008 | Tacit cultural perceptions | QT, EN | NO | Não |
| (LEVY et al., 2010) | 2010 | Tacit cultural perceptions | QT, ENa | CKADS | Não |
| (MITCHELL; SEAMAN, 2011) | 2011 | K-flow | AT, EN | K-map | Não |
| (MITCHELL; SEAMAN, 2016) | 2016 | K-flow | QT | NO | Não |
| (NEVES et al., 2014b) | 2014 | Risk Management, KM techniques | QT, OB, EN, DC | NO | Não |
| (ÖZEN et al., 2013a) | 2013 | OL | QT, EN | AiOLoS | Não |
| (PAULZEN et al., 2002) | 2002 | Knowledge processes | MMS | KPQM | Não |
| (RABELO et al., 2015) | 2015 | OC - <i>Organizational Culture</i> | EN, OB | OCAI | Não |
| (SINGH, 2008) | 2008 | Organizational Leadership | QT | NO | Não |
| (TAHERI et al., 2014) | 2014 | Requirement Elicitation, Knowledge components, knowledge sources | QT | KAM | Não |
| (TAHERI et al., 2015) | 2015 | Requirement elicitation process, Knowledge communication | QT | KAM | Não |
| (UNKELOS-SHPIGEL; HADAR; LEVY, 2013) | 2013 | Deployment architecture | QT, EN | SEKAM | Não |

**Legenda

EN (Entrevista), QT (Questionário), GV (Gravação), OB (Observação), DC (documentação)

AT (Artefatos), MMS (Maturity Model Stages)

AiOLoS (Assessing OL of software development organizations), CKADS (CommonKADS)

KPQM (Knowledge Process Quality Model), OCAI (Organizational Culture Assessment Instrument)

KAM (Knowledge Audit Model), SM (Survey Monkey), NO (nenhum)

Não: Não mencionado ou não identificado no trabalho.

Tabela 4 – Abordagens identificadas nos estudos selecionados em nossa RSL

| Artigo | Ano | Item do conhecimento investigado | Mét. Ap. | Mod. | Fer. |
|---|------|---|-------------|----------------|------------|
| (BADAMPUDI; WOHLIN; GORSCHKE, 2019) | 2019 | OC - <i>Organizational Culture</i> | EN | NO | Não |
| (BETZ; OBERWEIS; STEPHAN, 2014) | 2014 | Práticas de GC na organização, Captura de experiência | EN | NO | Não |
| (BORREGO et al., 2019) | 2019 | Transferência de conhecimento, Arquitetura de software AK, aplicação da GC | QT | UTEM | Não |
| (CORBIN; DUNBAR; ZHU, 2007) | 2007 | Práticas de GC, Avaliação de esforço investido | EN, OB e CL | NO | Não |
| (DAKHLI; CHOUIKHA, 2009) | 2009 | Transferência de conhecimento, Arquitetura de software AK, aplicação da GC | EN | NO | Não |
| (DAWSON, 2009) | 2009 | OC - <i>Organizational Culture</i> | EN e QT | MA | Não |
| (RABELO et al., 2015) | 2015 | OC - <i>Organizational Culture</i> | EN, OB e CL | SECI | Não |
| (DEMIRSOY; PETERSEN, 2018) | 2018 | OC - <i>Organizational Culture</i> | EN e DC | MA | Não |
| (DESOUZA; AWAZU; WAN, 2006) | 2006 | Transferência de conhecimento, Modelo de Gestão de Risco | SY | NO | Web Survey |
| (ELORANTA; KOSKIMIES, 2014) | 2014 | Transferência de conhecimento, Arquitetura de software, aplicação da GC | EN | NO | Não |
| (FANNOUN; KERINS, 2019) | 2019 | OL - <i>Organization Learning</i> | EN | NO | Não |
| (FEHÉR; GÁBOR, 2006) | 2006 | Práticas de KM na organização | OB e CL | NO | Não |
| (GASSTON; HALLORAN, 1999) | 1999 | OL - <i>Organization Learning</i> | EN e QT | OLEC | Não |
| (IVARSSON; GORSCHKE, 2012) | 2012 | OL - <i>Organization Learning</i> | QT | PSF | Não |
| (JOEL-EDGAR; GOPSILL, 2018) | 2018 | Transferência de conhecimento | EN | NO | NVivo |
| (JURADO et al., 2019) | 2019 | Práticas de GC, métodos aplicados na organização, atividades de GC | EN | KMM | Não |
| (KHOSRAVI et al., 2022) | 2022 | CKM - <i>Customer Knowledge Management</i> | EN e SY | CKM | Não |
| (KULKARNI; LOUIS, 2003) | 2003 | KMM - <i>Knowledge Management Model</i> | SY | CMM | Não |
| (LETHBRIDGE, 2000) | 2000 | Usabilidade, atratividade e contribuição da GC no desenvolvimento de software | QT | CTA | Não |
| (MITCHELL; SEAMAN, 2016) | 2016 | Transferência de conhecimento, Arquitetura de software AK, aplicação da GC | EN e QT | K-Flow e K-map | Não |
| (MONTONI et al., 2004) | 2004 | OL - <i>Organization Learning</i> | OB e CL | MA | Não |
| (NAMBU; SUEHIRO; YAMAGUCHI, 2019) | 2019 | Fontes de conhecimento e técnicas aplicadas na organização | QT | NO | Não |
| (ÖZEN et al., 2013b) | 2013 | OL - <i>Organization Learning</i> | EN e QT | AIOLos | Não |
| (PEDRAZA-GARCIA; ASTU-DILLO; CORREAL, 2016) | 2016 | Transferência de conhecimento, Arquitetura de software AK, aplicação da GC | EN | GTM | Não |
| (SANGAIAH et al., 2017) | 2017 | Transferência de conhecimento | QT | DTE | Não |
| (SPOLADORE; PESSOT, 2022) | 2022 | atividades de GC, técnicas, métodos aplicados na organização | QT, OB e CL | OEMs | Não |
| (TODOROVA; MILLS, 2018) | 2018 | Transferência de conhecimento, documentação, métodos aplicados na organização | SY | NO | Não |
| (TUZOVSKY; YAMPOLSKY; CHIRIKOV, 2005) | 2005 | Práticas de KM na organização | SY | NO | Não |
| (WAHEED et al., 2019) | 2019 | OL - <i>Organization Learning</i> | EN | NO | NVivo |

EN (Entrevista), QT (Questionário), OB (Observação), CL (Coleta), SY (Quest. Survey), DC (documentação) DTE (DEMATEL, TOPSIS, and ELECTRE), MA (Modelo de Auditoria do próprio autor), NO (nenhum) **Não**: Não mencionado ou não identificado no trabalho.

4 PROTOCOLO DA PESQUISA

Neste capítulo, serão abordados a motivação para o desenvolvimento desta pesquisa, o instrumento de análise utilizado, o objeto de estudo, a natureza e o método empregado na pesquisa, a amostragem e os instrumentos de coleta de dados utilizados.

4.1 Motivação

A gestão do conhecimento na indústria de software é crucial para a sustentabilidade e a evolução dos sistemas. No entanto, muitas organizações subestimam essa estratégia de negócio. A partir da experiência profissional na área, foi possível observar, no dia a dia das empresas, a ausência de processos estruturados para captura, armazenamento e compartilhamento do conhecimento.

Essa lacuna resulta em desafios significativos, como retrabalho, dificuldades na manutenção de sistemas legados e perda de expertise devido à rotatividade de profissionais.

Para aprofundar o entendimento sobre o tema, realizou-se uma pesquisa em artigos científicos relevantes nas principais bases de dados, como [IEEE Xplore](#), [ACM Digital Library](#) e [ScienceDirect](#). Embora diversos estudos tenham abordado a temática, um trabalho específico se destacou: a Revisão Sistemática da Literatura (RSL) conduzida por Maciel et al. (2018), intitulada "*Knowledge Management Diagnostics in Software Development Organizations: a Systematic Literature Review*" (Diagnóstico da Gestão do Conhecimento em Organizações de Desenvolvimento de Software: uma Revisão Sistemática da Literatura). Esse estudo forneceu diretrizes essenciais para a pesquisa, porém sua análise se limitava a dados até 2018, tornando necessária uma atualização.

Dessa forma, foi realizada uma extensão da RSL de Maciel et al. (2018), utilizando os mesmos protocolos de pesquisa, mas ampliando o escopo até o ano de 2024. Os detalhes dessa extensão encontram-se na **Seção 3**.

A análise conjunta da RSL original e da extensão conduzida por Souza, Silva e Guedes (2024b) revelou alguns pontos-chave:

- Ausência de ferramentas e modelos amplamente reconhecidos para a gestão do conhecimento (GC);
- A necessidade aprimorar a qualidade do software desenvolvido;

- O potencial da gestão do conhecimento arquitetural, dada sua relevância para as organizações;
- O aprendizado organizacional e a cultura como um dos aspectos mais investigados na literatura.

Com base nesses achados, definiu-se como pertinente aprofundar a pesquisa sobre a Gestão do Conhecimento Arquitetural (GCA), buscando estratégias ágeis que evitem a perda de conhecimento pré-existente e permitam sua centralização por meio de uma ferramenta de software prática e dinâmica. Dessa maneira, busca-se mudar a cultura organizacional para que o conhecimento seja alimentado e utilizado como parte de sua rotina.

Para alcançar esse objetivo, optou-se pela metodologia de Pesquisa-Ação, conforme descrito na **Seção 4.2**. A escolha se justifica pelo envolvimento direto de um dos pesquisadores, que também atua como colaborador na empresa objeto de estudo (vide **Seção 5.1**). O processo de execução da pesquisa e suas etapas estão detalhados na **Seção 4.3**.

4.2 Pesquisa-ação

O método pesquisa-ação adota uma abordagem democrática e participativa, voltada para o desenvolvimento do conhecimento prático, com o objetivo de alcançar propósitos humanos significativos (STARON, 2020). Sendo o objeto desta pesquisa uma organização com um problema real, que exige dinamismo e praticidade em seu contexto, a pesquisa-ação se mostra como um método adequado. Nessa metodologia, a ação realizada é interativamente melhorada através de múltiplos ciclos (PETERSEN et al., 2014). A flexibilidade no planejamento da pesquisa-ação, em contraste com outros tipos de pesquisa (COSTA; POLITANO; PEREIRA, 2014), traz o dinamismo necessário para sua aplicação na indústria de software. Para Coughlan and Coughlan (2002), uma característica que diferencia a pesquisa-ação em relação a outras metodologias é a participação ativa do pesquisador no contexto de sua investigação, resultando em uma pesquisa comprometida com a produção de conhecimento por meio de situações práticas da organização.

Aprofundando-se na metodologia, Costa, Politano and Pereira (2014) afirmam que a pesquisa-ação possui dois objetivos: resolver um problema dentro de uma organização e **contribuir com a ciência**. Como o pesquisador participa ativamente do processo de

geração do conhecimento, a academia e a indústria trabalham juntos (STARON, 2020). Para reforçar essa união, Coughlan and Coughlan (2002) destacam que a pesquisa-ação deve apresentar características como:

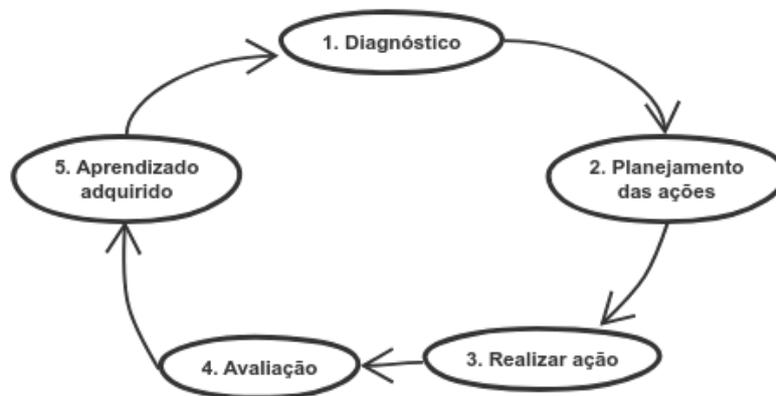
1. Envolvimento direto dos pesquisadores;
2. Realização de pelo menos dois ciclos;
3. Interatividade;
4. Desenvolvimento de uma compreensão holística;
5. Foco em mudanças;
6. Compreensão do quadro ético do objeto de pesquisa;
7. Inclusão de diversos métodos de coleta de dados;
8. Pré-compreensão do problema abordado;
9. Condução em tempo real;
10. Critérios de qualidade próprios.

Essas características servem como guias práticos para dinamizar o processo de pesquisa, tornando-a uma metodologia atrativa (MCKAY; MARSHALL, 2001). Assim, a pesquisa-ação busca reunir ações e reflexões na busca de soluções para questões urgentes nas organizações (STARON, 2020).

Para aplicar a metodologia de pesquisa-ação, Petersen et al. (2014) definem um ciclo básico composto por cinco etapas (Vide **Figura 3**):

1. **Diagnóstico:** Compreende e descreve o problema;
2. **Planejamento das ações:** Identifica alternativas e define um plano de ação;
3. **Realizar a ação:** O plano de ação é implementado;
4. **Avaliação:** Estuda os efeitos da ação por meio de ferramentas como observação, questionários e entrevistas;
5. **Aprendizado adquirido:** Define o conhecimento geral obtido e decide se novos ciclos são necessários.

Figura 3 – As etapas básicas de um projeto de Pesquisa-ação.



Fonte: Adaptado de Petersen et al. (2014).

Dick (2000) reforça que uma pesquisa-ação deve apresentar quatro pontos básicos: (i) atuar em uma situação para ampliar o conhecimento, (ii) possuir natureza cíclica, (iii) incluir reflexão crítica sobre o processo e resultados, e (iv) ser predominantemente qualitativa, embora métodos quantitativos possam ser utilizados em algumas situações.

Para compreender melhor a adaptabilidade da pesquisa-ação, McKay and Marshall (2001) propõem um ciclo de oito etapas (vide **Figura 4**): (1) identificação de um problema real; (2) reconhecimento e busca de fatos sobre o problema; (3) elaboração de um plano de ação; (4) implementação do plano; (5) monitoramento das ações; (6) avaliação dos efeitos; (7) aperfeiçoamento da solução, caso necessário; e (8) conclusão, com a resolução do problema e possíveis ajustes futuros.

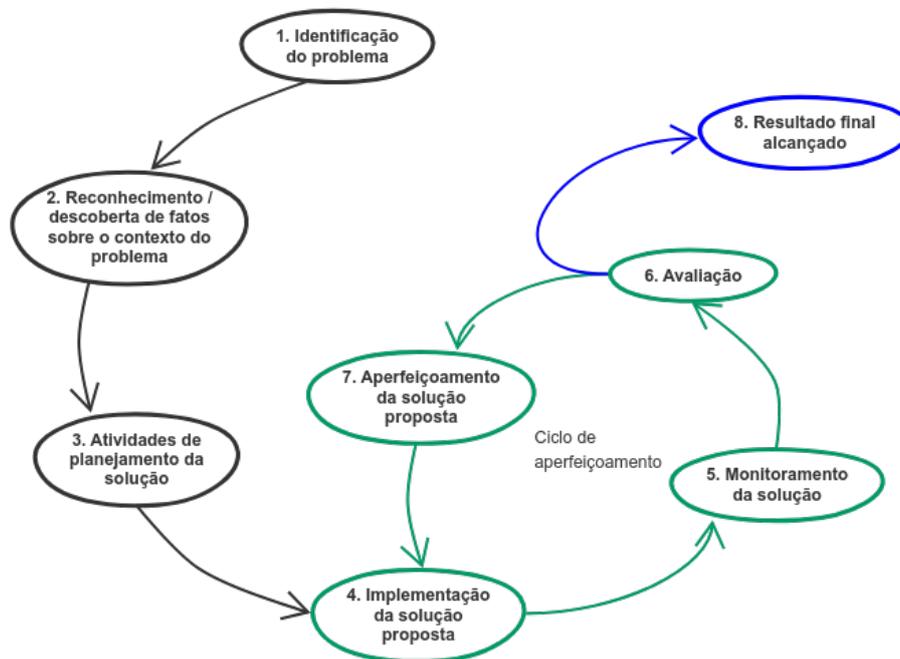
Ao concluir os ciclos, o problema deve ter uma solução aceitável, alinhada aos objetivos da pesquisa (COSTA; POLITANO; PEREIRA, 2014).

4.3 O método e a natureza da pesquisa

Quanto à natureza, esta é uma pesquisa aplicada, pois busca gerar conhecimento com aplicação prática na **Gestão do Conhecimento Arquitetural (GCA)** dentro de uma organização real. O estudo envolve a análise do contexto organizacional, a identificação de desafios na gestão do conhecimento arquitetônico e a implementação de uma solução baseada em software para otimizar esse processo.

Para isso, será necessário compreender detalhadamente o funcionamento do setor de arquitetura da organização, incluindo seus fluxos de trabalho, práticas atuais de documentação e compartilhamento de conhecimento, além de avaliar o impacto da solução

Figura 4 – As etapas de um projeto de Pesquisa-ação.



Fonte: Adaptado de McKay and Marshall (2001).

proposta na colaboração e eficiência das equipes.

Quanto ao método de pesquisa, optou-se pela pesquisa-ação, devido à sua adaptabilidade à rotina da indústria. A representação das etapas que serão executadas durante a pesquisa pode ser visualizada na **Figura 5**.

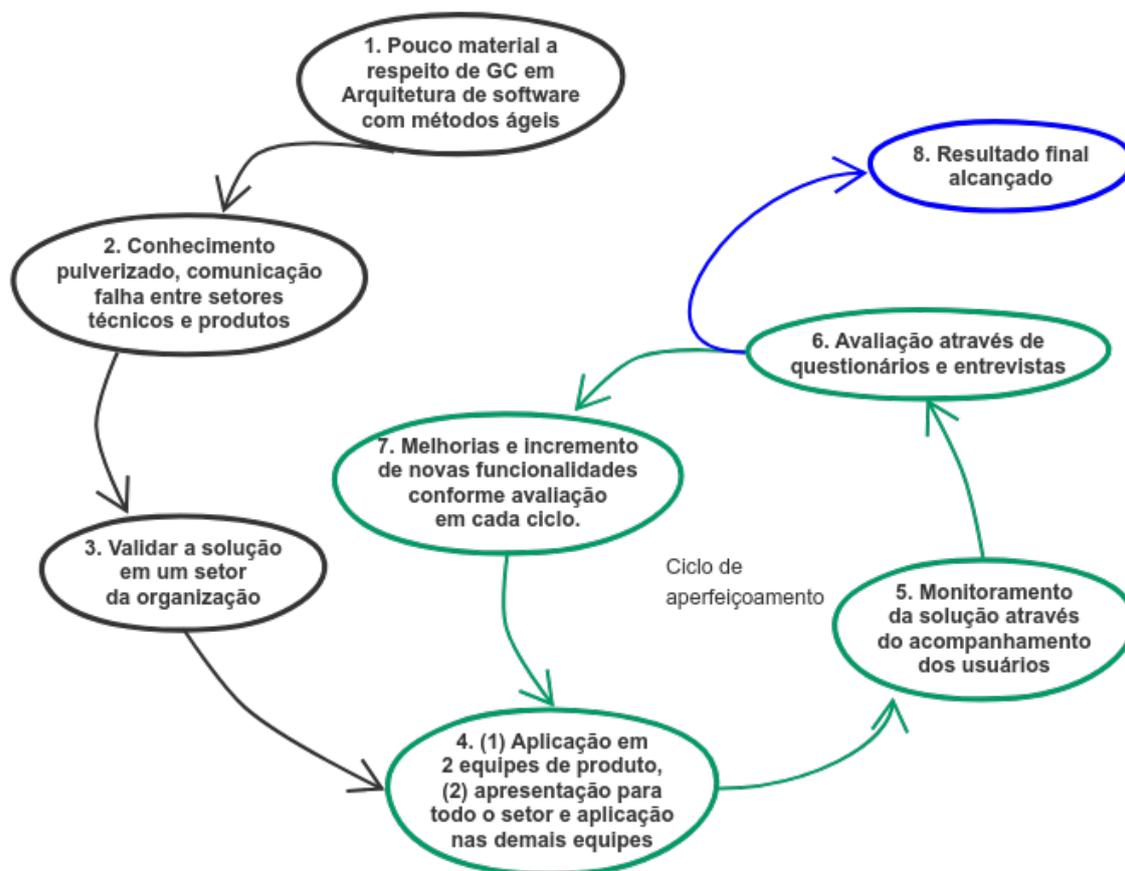
Quanto aos métodos de coleta de dados, foram aplicados questionários de caráter qualitativo e quantitativo, além de entrevistas. A combinação dessas abordagens permite uma análise mais ampla e aprofundada sobre o tema, contribuindo para uma compreensão mais abrangente ao longo da execução do estudo. Segundo Costa, Politano and Pereira (2014), o modelo quantitativo permite obter uma combinação de fatos que aguardam exploração pelo pesquisador, enquanto o modelo qualitativo, por ser essencialmente indutivo, visa construir conceitos, pressuposições ou teorias sobre o tema investigado.

Quanto à função descritiva desta pesquisa, conforme apontado por Jinkings (2022), trata-se de um estudo conclusivo, com foco na descrição e análise das relações entre os fatores investigados. Para isso, são considerados os resultados obtidos por meio dos métodos qualitativo e quantitativo, bem como o processo de implementação da GCA e as melhorias geradas no cotidiano da organização.

5 FASE INICIAL DA PESQUISA-AÇÃO

Neste capítulo serão abordadas as três primeiras etapas da pesquisa conforme demonstrado na Figura 5. De acordo com McKay and Marshall (2001), após a *identificação inicial do problema (etapa 1)*, seguindo da, *atividade de reconhecimento e apuração de fatos (etapa 2)*, onde o pesquisador se esforça para descobrir mais sobre a natureza do problema e o seu contexto, é possível, então, traçar uma estratégia inicial para a *implementação das atividades de planejamento da ação (passo 3)*, para dar, assim, início ao ciclo de processos que trarão a resolução do problema identificado.

Figura 5 – Etapas para realização da pesquisa.



Fonte: Adaptado de Petersen et al. (2014).

5.1 Identificação do problema

Ao analisar os resultados obtidos na literatura por meio da extensão da Revisão Sistemática da Literatura (RSL) conduzida por Souza, Silva and Guedes (2024b), e ao compará-los com o cenário observado na organização objeto da pesquisa — da qual o autor dessa dissertação é colaborador — tornou-se possível compreender de forma mais aprofundada os desafios inerentes à gestão do conhecimento na indústria de software, com ênfase na Gestão do Conhecimento Arquitetônico (GCA).

A organização caracteriza-se como uma empresa multinacional de grande porte, especializada no desenvolvimento de software, contando com aproximadamente 3.000 programadores. Seu portfólio de atuação abrange mais de 12 segmentos de mercado, incluindo agronegócio, logística, manufatura e distribuição, e seu conjunto de clientes supera a marca de 50 mil empresas.

No que tange às abordagens de desenvolvimento adotadas, a organização opera com duas modalidades principais: *Software as a Service* (SaaS), no qual a fornecedora do produto assume a responsabilidade pela infraestrutura necessária para a disponibilização do software, e *On-Premise*, em que a infraestrutura de TI é implementada e mantida localmente pelo cliente.

A investigação realizada se utilizando do modelo pesquisa-ação, será no setor de produção de softwares voltados para o segmento de logística, denominado internamente “**Tribo Logística**”. A estrutura do software desenvolvido nesse setor é baseada no modelo SaaS. Atualmente, o setor conta com 10 times de desenvolvedores responsáveis, em média, por 15 produtos.

O processo de manutenção dos produtos existentes é realizado por equipes compostas por desenvolvedores (*Backend, Frontend, Mobile, etc*), geralmente um *Quality Assurance* (QA), responsável pelos testes de funcionalidades e pela garantia da qualidade das entregas; um *Product Owner* (PO), que atua como ponte entre o time e o cliente e define as funcionalidades a serem implementadas; um gestor com o papel de *Team lead*, responsável por administrar a equipe e suas entregas; e um *User Experience* (UX) *Designer*, encarregado do design das telas do sistema, assegurando a usabilidade do produto para o usuário final.

As linguagens de programação adotadas em sua maioria no *front-end* é o TypeScript utilizando o framework de código aberto o Angular®¹. O *back-end* utiliza como

¹<https://angular.dev/>

linguagem o Java®² e como principal framework o Spring boot®³. Como sistema de mensageria para comunicação entre diferentes produtos gerados na organização o RabbitMQ®⁴.

As equipes seguem metodologias ágeis combinadas com o modelo Kanban®, utilizando o Jira®⁵ como ferramenta de gestão. Conforme sugerido pelo *Agile*, são realizadas reuniões diárias curtas, nas quais os desenvolvedores expõem o que foi feito no dia anterior e apresentam o planejamento para o dia atual.

No fluxo de trabalho, as tarefas a serem executadas são agrupadas em épicos, que representam entregas de valor ao cliente. Por exemplo, no desenvolvimento do faturamento de produtos, o épico engloba todas as tarefas de desenvolvimento distribuídas no Kanban®, assim como os testes sistêmicos para validar se a entrega atende às necessidades do cliente.

Os times de produto contam com o suporte de setores específicos, como o **time de arquitetura de software**, responsável pela gestão da estrutura do código-fonte, disseminação de boas práticas e inclusão de novas tecnologias. Há também o **time de DevOps (*Development Operations*)**, dividido em duas equipes: uma encarregada da segurança e manutenibilidade da infraestrutura dos produtos, e outra responsável pela configuração e manutenção da estrutura de *Continuous Integration/Continuous Delivery (CI/CD)*. Esta segunda equipe atende não apenas à suíte logística, mas a todos os setores que utilizam o modelo SaaS.

Inicialmente, foram realizadas conversas exploratórias, em momentos distintos, com dois membros da equipe de arquitetura de software, seu respectivo gestor, um *Team Lead* responsável por duas equipes de produtos e dois *Product Owners* de três produtos diferentes.

Como resultado dessas conversas, foi descoberto a existência de repositórios de documentação criados pela equipe de arquitetura, contendo descrições de estruturas arquitetônicas, boas práticas sugeridas e roteiros de processos para o desenvolvimento de software nos times de produto. Segundo Farenhorst and Boer (2009), a interação entre conhecimento tácito e explícito é essencial para a formação da GCA.

Embora a organização se utilize de uma proposta inicial do modelo proposto (GCA), ele se encontra fragmentado e pouco divulgado entre os times. Boer et al. (2007)

²<https://www.java.com/pt-BR/>

³<https://docs.spring.io/>

⁴<https://www.rabbitmq.com/>

⁵(SOFTWARE, 2024)

sugerem um modelo centralizado de conhecimento arquitetônico que forneça mais informações sobre a arquitetura do software e os conceitos que devem ser priorizados no suporte à gestão desse conhecimento.

A análise realizada com os times de produto, por meio das conversas com os *Product Owners*, revelou a existência de outro canal de documentação, semelhante a uma **Wiki®**, que funciona como um sistema de gestão de conteúdo contendo informações específicas das equipes e dados úteis sobre os produtos. Nesse canal, podem ser encontrados diagramas de classe, contratos de acesso à API para diferentes times da suíte logística e documentação de negócio referente aos produtos.

É fundamental compreender que os **Requisitos Não Funcionais (RNFs)** influenciam indiretamente as decisões arquiteturais, exigindo um equilíbrio entre os RNFs desejados para o sistema e os benefícios proporcionados pela solução arquitetônica adotada (VERONEZ et al., 2022).

Conforme apontado, o conhecimento organizacional já existe, mas está disperso em diferentes locais. De acordo com a experiência do pesquisador, que é colaborador da organização, não há uma divulgação clara nem um acesso prático e eficiente a cada ferramenta disponível. Isso indica falta de agilidade e clareza na comunicação entre o time de arquitetura e os times de produto.

Segundo o gestor do time de arquitetura, "**João Antônio**⁶", é de conhecimento que há um grande volume de informações distribuídas em diferentes canais de comunicação, além de uma parte significativa estar desatualizada e necessitar de revisão e outras até de remoção.

Após essa análise inicial para identificação do problema na organização, conclui-se que é necessário aprimorar os conceitos que envolvem a GCA, de forma centralizada, visando melhorar a qualidade do software desenvolvido.

Esse problema se mostra relevante para a indústria porque afeta a melhoria na qualidade do software desenvolvido e para a academia porque pode gerar novos conhecimentos sobre a aplicação prática dos conceitos envolvendo a GCA.

5.2 Reconhecimento e Descoberta de Fatos sobre o Contexto do Problema

Com o problema identificado, conforme apresentado na **Seção 5.1**, nesta etapa é realizado um levantamento mais completo de como o **Conhecimento Arquitetural (CA)**

⁶Nome fictício

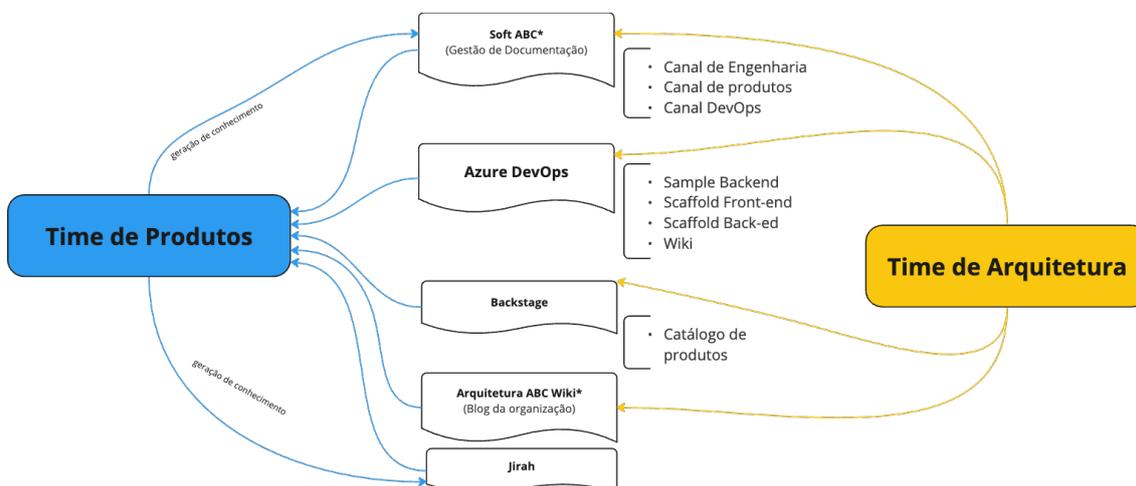
é armazenado e compartilhado na organização. Foram realizadas ações para a descoberta de fatos que corroboram a existência de um problema organizacional. O detalhamento dos canais de comunicação existentes na organização, a possível existência de uma metodologia para a geração de conhecimento e um questionário quali-quantitativo com o time de desenvolvedores de produto foram os elementos investigados.

5.2.1 Análise dos Canais de Comunicação Existentes

Em conversa com o arquiteto de software "**Antônio Afonso**"⁷ responsável pelo *frontend*, foi apontada a estrutura existente para a distribuição de conhecimento técnico e de negócio na organização, conforme demonstrado na **Figura 6**.

Foi questionada a existência de uma metodologia ou padrão para a geração de conhecimento. No entanto, foi informado que o único padrão ocorre no início de cada projeto. São realizadas reuniões para estruturar o software, definir as decisões arquiteturais com base nos requisitos funcionais e não funcionais, e todos os artefatos gerados são documentados e armazenados no "*Soft ABC*". Outros tipos de documentação são gerados esporadicamente, conforme a necessidade relatada pelos times da *Suíte Logística*.

Figura 6 – Distribuição do conhecimento organizacional



Fonte: o autor. (*Nomes fictícios)

Pode-se observar na **Figura 6**, que os "Times de produtos" têm acesso a cinco ferramentas oficiais para o armazenamento do conhecimento existente. No entanto, apenas dois canais podem ser alimentados diretamente: o "*Soft ABC*" e o "*Jira*®".

⁷Nome fictício

Temos também, a representação do "Time de arquitetura" na cor laranja, o qual tem acesso como editor/criador a praticamente todos os canais com exceção do "Jirah®", que é exclusivo para a gestão ágil dos times de produto. Os canais existentes são descritos a seguir:

1. **"Soft ABC⁸"**: Plataforma de gestão de documentação, onde são armazenadas documentações geradas pelas equipes. Internamente, possui vários canais, sendo destacados aqui um mantido pelo time de arquitetura, denominado "*Canal de Engenharia*", e os "*Canais de Produtos*", que podem ser alimentados pelo próprio time.
2. Repositório **"Azure DevOps®⁹"**: Contém códigos de exemplo para novos projetos, denominados *Scaffold*. Essas estruturas são atualizadas sempre que necessário, garantindo que os times de produto estejam cientes da estrutura atual adotada pela arquitetura de software da organização para novos produtos. Dentro desse repositório, foi criado também um canal modelo *wiki®*, contendo algumas documentações geradas pelo time de arquitetura.
3. **Backstage®¹⁰**: Ferramenta denominada "portal do desenvolvedor" fornecida pelo Spotify®, (SPOTIFY, 2024). É Open-source e permite personalização para atender às necessidades das organizações de software e facilitar o dia a dia dos desenvolvedores. Na organização, a ferramenta está sendo subutilizada, servindo apenas como acesso rápido ao CI/CD dos times de forma centralizada e permitindo a visualização dos repositórios dos componentes que fazem parte do produto
4. **"Arquitetura ABC wiki"**: Diferente das ferramentas citadas anteriormente, essa não está dentro da "Suíte Logística", mas sim no canal de comunicação da organização com toda a estrutura que opera no modelo *SaaS*. Contém informações mais genéricas sobre as ferramentas e métodos utilizados na estrutura. Devido ao dinamismo da tecnologia e à falta de manutenção periódica, muitas informações encontram-se desatualizadas.
5. **Jirah®**: Ferramenta utilizada para a gestão do produto durante todo o processo de desenvolvimento e manutenção, empregando práticas ágeis tanto pelos desenvolvedores quanto pelo pessoal de negócio e testes. Possui um histórico completo

⁸Nome fictício pois leva o nome da organização

⁹(MICROSOFT, 2024)

¹⁰(SPOTIFY, 2024)

de todas as etapas do desenvolvimento, incluindo links de documentação e todo o *roadmap* do produto.

5.2.2 Análise Diagnóstica com os Times de Produto

Nesta seção, são apresentados os resultados da análise realizada com os desenvolvedores de software da equipe de produtos. O objetivo dessa análise é compreender a visão da equipe em relação à forma como a organização trata a Gestão do Conhecimento e o Conhecimento Arquitetônico, visando auxiliar o cotidiano dos desenvolvedores de software.

Como ferramenta para a execução desse estudo, utilizou-se o *Google Forms*. O instrumento foi composto por: (i) Apresentação e termo de consentimento, garantindo a confidencialidade dos dados fornecidos e o anonimato dos participantes; (ii) Questionário quantitativo, separado em três seções: *Gestão do Conhecimento em Arquitetura de Software na organização*, *Competências Organizacionais* e *Gestão do Conhecimento Arquitetônico*; (iii) Questionário qualitativo, para proporcionar uma visão mais clara de cada participante sobre o tema abordado.

Tabela 5 – Visão Geral dos participantes do estudo

| Papel | Pessoa | Projetos | Cargo* | empresa* | Profissional* |
|---------------------|---------------|-----------------|---------------|-----------------|----------------------|
| <i>Júnior</i> | P6 | 1 | 1-5 | 1-5 | 1-5 |
| <i>Pleno</i> | P2 | 1 | >5 | 1-5 | >10 |
| | P11 | 2-3 | <5 | 1-5 | >5 |
| <i>Sênior</i> | P1 | >3 | >10 | >10 | >10 |
| | P4 | 2-3 | 5-10 | 1-5 | 5-10 |
| | P10 | 2-3 | 5-10 | 1-5 | 5-10 |
| | P12 | 1 | <1 | <1 | 1-5 |
| <i>Especialista</i> | P3 | 1 | 1-5 | 1-5 | >5 |
| | P5 | 1 | 5-10 | 1-5 | >10 |
| | P7 | 1 | 1-5 | 1-5 | >10 |
| | P8 | 2-3 | <5 | 1-5 | 5-10 |
| | P9 | >3 | 1-5 | >10 | >10 |
| | P13 | 2-3 | >5 | 1-5 | 5-10 |
| | P14 | 2-3 | 1-5 | 1-5 | >10 |
| | P15 | 2-3 | 1-5 | 1-5 | >10 |
| | P16 | 2-3 | 1-5 | 1-5 | >10 |

* *Medida em anos*

Os participantes deste estudo fazem parte de duas equipes, cada uma com oito integrantes. A primeira equipe foi selecionada por incluir o pesquisador, enquanto a segunda

foi escolhida por estar sob a mesma liderança técnica (*Team Lead*). Os participantes possuem diferentes papéis e níveis de experiência, tanto na organização quanto na área de tecnologia. A **Tabela 5** apresenta um resumo dessas informações, destacando, em sua maioria, profissionais experientes em suas respectivas áreas (*frontend ou backend*), com carreiras bem consolidadas.

5.2.2.1 Execução do experimento

Os desenvolvedores selecionados para a pesquisa foram convidados a participar de uma reunião para apresentação do tema envolvendo *Gestão do Conhecimento Arquitetônico* e sua influência na gestão do conhecimento organizacional. Dessa forma, foi possível compreender, na visão dos desenvolvedores, o quanto o assunto tratado influencia a qualidade do produto desenvolvido e o tempo utilizado no dia a dia em suas tarefas rotineiras.

Após a apresentação, todos foram convidados a participar de um questionário avaliativo sobre o tema. Foi permitido que eles respondessem em tempo oportuno, para que não sofressem nenhuma interferência ao responder às perguntas.

Para medir o nível de aceitação dos participantes em relação às questões quantitativas, utilizou-se uma escala Likert de 1 a 5, sendo: 1 - Discordo totalmente; 2 - Discordo parcialmente; 3 - Neutro; 4 - Concordo parcialmente; e 5 - Concordo totalmente. As perguntas realizadas nesse questionário podem ser vistas na **Tabela 6**.

Tabela 6 – Questionário elaborado para coletar as percepções dos desenvolvedores.

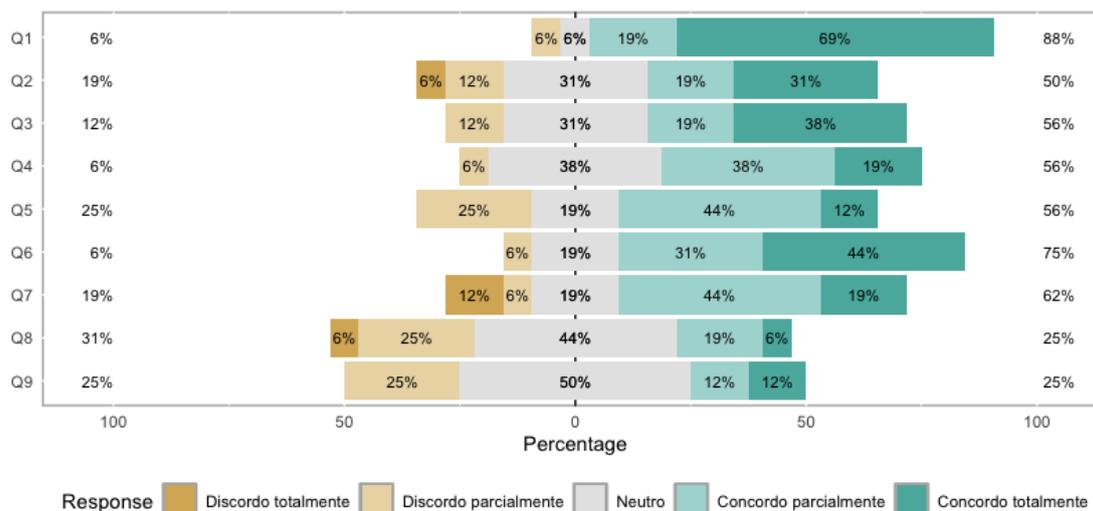
| | |
|--|--|
| Questionário Quantitativo | Gestão do conhecimento em arquitetura de software na organização |
| | Q01 – A gestão do conhecimento para desenvolvimento de software, na sua opinião, se mostra como sendo um tema importante para a empresa. |
| | Q02 – A partilha de ideias dentro da organização é algo comum e frequente. |
| | Q03 – Estimula-se a aprendizagem por meio da ampliação dos contatos e das interações entre as pessoas dentro e fora da organização. |
| | Q04 – A Organização possui os meios, ferramentas e cerimônias ideais para a troca de conhecimento. |
| | Q05 – A Organização possui os meios, ferramentas e cerimônias ideais para manutenção do conhecimento explícito, isto é, aquilo que é documentado. |
| | Q06 – Sinto que o trabalho em equipe é muito valorizado pelos funcionários. |
| | Q07 – Sinto que a empresa promove o trabalho em equipe, inclusive entre equipes distintas. |
| | Q08 – A comunicação é eficiente em todos os níveis e entre as áreas distintas. (Arquitetura, equipes de produto, SRE, Totvs Apps, etc). |
| | Q09 – Percebo que as informações são compartilhadas. Há um grande acesso por parte dos funcionários a base de conhecimento da organização. |
| | Competências Organizacionais |
| | Q10 – Na sua opinião, os funcionários da empresa se mostram qualificados nas funções técnicas que desempenham. |
| | Q11 – Na sua opinião, existe um elevado sentimento de comprometimento dos funcionários com os objetivos estabelecidos pela organização. |
| | Q12 – As pessoas e/ou as equipes de trabalho utilizam de forma adequada os recursos disponíveis (tecnologias de informação e comunicação). |
| | Q13 – Os funcionários possuem amplos conhecimentos do mercado onde a empresa desenvolve a atividade. |
| | Q14 – Os conhecimentos adquiridos pelos funcionários em todos os níveis são devidamente aplicados no desenvolvimento dos projetos. |
| | Q15 – O padrão definido pela tribo (Logística SaaS) no quesito desenvolvimento ágil e arquitetura são respeitados durante o processo de desenvolvimento de software nas equipes. |
| | Gestão do Conhecimento Arquitetônico |
| | Q16 – A transferência de conhecimento da arquitetura para com os times de projeto é eficaz. |
| Q17 – Os atuais sistemas que realizam a gestão do conhecimento organizacional são de fácil acesso. | |
| Q18 – A organização possui canais que contém documentação técnica e o conhecimento arquitetural que é importante para o conhecimento de todos. | |
| Q19 – É claro para mim que a organização proporciona um ambiente de trabalho onde as pessoas são encorajadas a compartilhar ideias, experiências, sucessos e fracassos | |
| Q20 – Os objetivos da estrutura de arquitetura são claros para mim | |
| Q21 – O conhecimento da estrutura da arquitetura de sistemas na suite, permite que eu crie a minha própria linha de aprendizado para minha evolução de carreira de forma sustentável. | |
| Q22 – Eu regularmente me atualizo com as informações trazidas pelo time de arquitetura com sentido de melhoria para o meu projeto. | |
| Q23 – O tempo de resposta do time de arquitetura para os produtos está adequado. | |
| Q24 – A equipe de arquitetura tem controle suficiente das atividades de arquitetura e design. | |
| Pontos a melhorar em relação a gestão do conhecimento de arquitetura dentro da organização | |
| QL01 – Na sua opinião quais pontos deveriam ser melhor comunicados dentro da organização. | |
| QL02 – Na sua opinião quais informações são importantes de se ter um fácil acesso dentro do time, para desenvolvedores experientes no projeto e novos desenvolvedores que chegam ao projeto, trazendo uma melhor comunicação e melhoria no tempo gasto de desenvolvimento. | |

Quali.

5.2.2.2 Resultados Quantitativos

A **Figura 7** traz os resultados obtidos em relação à Gestão do Conhecimento em Arquitetura de Software na organização. Ao examinarmos o gráfico, percebemos que 88% dos desenvolvedores concordam que o tema é importante para a empresa (**Q1**).

Figura 7 – Gestão do conhecimento em arquitetura de software na organização



Fonte: o autor.

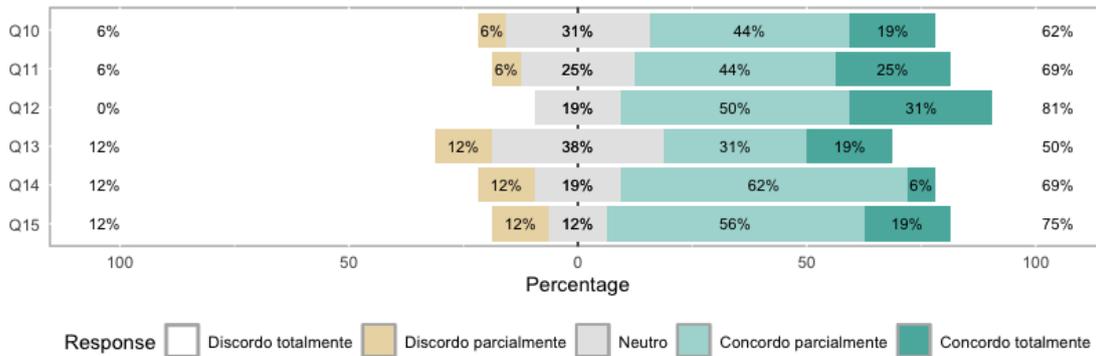
O desenvolvedor P5 afirma que: "(...) acredito que vale a pena refletir sobre quais informações usamos com mais frequência e deixá-las em um local de fácil acesso." Além disso, 75% dos desenvolvedores afirmam que o trabalho em equipe é muito valorizado pelos colaboradores (**Q6**). De acordo com o desenvolvedor P8, "o trabalho em equipe tem que ser prioridade para o sucesso de um projeto. Sem isso, as diferentes habilidades de cada integrante não podem ser aproveitadas ao máximo."

Em relação à comunicação em áreas e níveis hierárquicos distintos (**Q8**) e sobre o fácil acesso à base de conhecimento organizacional (**Q9**), mais de 40% dos participantes se mostram neutros a respeito, e mais de 25% discordam da eficiência do tema. Conforme apontado pelo entrevistado P2, "a comunicação entre os times de outras áreas deveria ser mais fácil". Isso mostra que existe uma necessidade de melhorar a forma como as equipes de diferentes setores se comunicam dentro da suíte logística.

Ao analisar as **Competências Organizacionais**, conforme pode ser constatado na **Figura 8**, dois pontos merecem destaque neste conjunto de perguntas: a referência aos desenvolvedores que se mostraram qualificados tecnicamente para suas funções (**Q10**), na

qual obteve-se uma concordância de 62%, e o segundo referente ao amplo conhecimento dos entrevistados no modelo de negócio em que atuam (Q13), com 50% de concordância, porém com 38% se mostrando neutros e 12% discordando parcialmente.

Figura 8 – Competências organizacionais



Fonte: o autor.

Isso demonstra uma necessidade da organização em focar, além do conhecimento técnico, no conhecimento de negócio dos seus produtos, para repasse aos desenvolvedores. De acordo com o desenvolvedor P10, *"há a necessidade de uma maior exploração a respeito das regras de negócio dos produtos, principalmente para os novos colaboradores, e, se existe [esse material], ele deveria ser divulgado com maior clareza, assim como os canais que o disponibilizam."*

Em relação aos desenvolvedores utilizarem de forma adequada os recursos disponibilizados pela organização (Q12), houve uma concordância de 81%. Segundo P8, ao ser questionado a respeito desta pergunta, disse que: *"Temos boas ferramentas que nos são fornecidas para o trabalho no dia a dia, e realmente utilizamos como nos é sugerido. Porém, temos ciência de que os métodos podem ser revisados e alguns até melhorados."* Essa facilidade de se adaptar ao que lhes é fornecido também pode ser relacionada ao fato de a maioria ocupar o papel de especialista, que apresenta a característica de se adaptar mais rapidamente ao ambiente de trabalho.

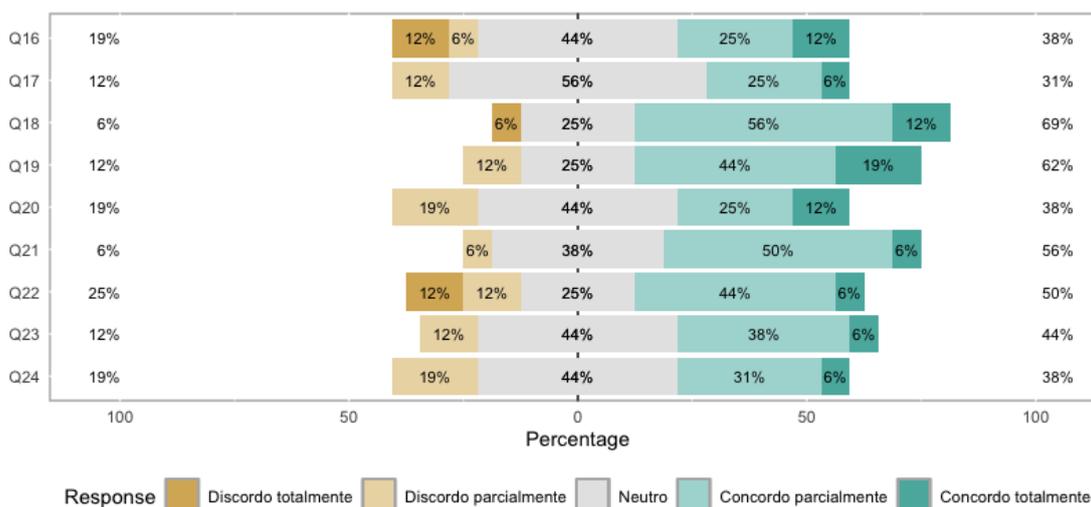
Com relação às questões envolvendo a **Gestão do Conhecimento Arquitetônico (GCA)**, como pode ser constatado na **Figura 9**, temos, em um primeiro momento, uma quantidade elevada de pessoas com opiniões neutras, entre 25% e 55% das respostas. JÚNIOR et al. (2024) sugerem que uma alta incidência de respostas neutras pode refletir a ausência de uma opinião definida ou conhecimento insuficiente sobre o tema abordado.

Quando questionados sobre a facilidade de acesso às informações na organização

(Q17), 56% deram respostas neutras. Isso reforça a importância da abordagem deste trabalho, que busca melhorar o acesso ao *Conhecimento Arquitetônico (CA)*. Como aponta P4, "(...) vale a pena refletir sobre quais informações usamos com mais frequência para deixá-las em um local de fácil acesso."

A organização já conta com um canal centralizado para documentação técnica (Q18), com 69% de concordância. No entanto, P5 destaca um problema: "*Documentações técnicas e sobre a estrutura dos produtos existem, porém a ferramenta possui um sistema de busca muito ruim, de modo que, por vezes, é necessário salvar o link como favorito para não precisar procurar novamente e correr o risco de não encontrar.*" Isso mostra que a recuperação das informações ainda é um desafio.

Figura 9 – Gestão do Conhecimento Arquitetônico (GCA)



Fonte: o autor.

Quando se trata da clareza dos objetivos da estrutura de arquitetura para com os times de produto (Q20), há um grande percentual de entrevistados permanecendo neutros (44%), bem como os que discordam parcialmente (19%) dessa afirmação. De acordo com o relato de P7, "*acredito que falte um boilerplate/scaffold padronizado para que os times possam iniciar novos projetos de forma ágil e adequada, assim como um suporte maior da arquitetura com relação ao entendimento da arquitetura proposta.*"

5.2.2.3 Análise Qualitativa

Foi realizada uma análise detalhada dos comentários feitos pelos desenvolvedores nas duas últimas questões abertas do questionário aplicado. As respostas foram examina-

das qualitativamente, seguindo um processo de codificação.

O propósito dessa análise foi organizar, categorizar e sintetizar os dados, permitindo identificar as necessidades sentidas pelas equipes de produto referentes à comunicação e à facilitação do trabalho cotidiano. Essa abordagem baseou-se no modelo proposto por Silva, Steinmacher and Conte (2019), que descreve um procedimento de análise qualitativa em quatro etapas:

1. **Primeira etapa:** Todas as respostas foram lidas integralmente, proporcionando uma visão abrangente dos dados coletados durante a fase de reconhecimento dos fatos sobre o problema. Além disso, foi realizada uma triagem para eliminar respostas em branco ou fora do contexto;
2. **Segunda etapa:** Foi realizada a codificação aberta, na qual foram definidos códigos correspondentes a conceitos relevantes para a compreensão da percepção dos participantes com base em suas respostas;
3. **Terceira etapa:** Foi realizada a codificação axial, na qual os códigos foram agrupados com base em suas propriedades, dando origem a conceitos que representam subcategorias e categorias;
4. **Quarta etapa:** Foi conduzida uma avaliação abrangente na análise final para assegurar a consistência dos resultados obtidos.

Destaca-se que um pesquisador liderou a análise e a discutiu com outros especialistas em análise qualitativa durante reuniões sucessivas, com o objetivo de reduzir possíveis vieses no processo de codificação. Os achados da análise qualitativa e as discussões realizadas proporcionaram insights valiosos para a etapa de planejamento da solução.

Todas as respostas referentes a esta análise qualitativa estão presentes no **Apêndice A**.

5.2.2.4 Resultados da Análise Qualitativa

Nesta seção do questionário, o principal tópico abordado refere-se aos **pontos a melhorar em relação à gestão do conhecimento arquitetural dentro da organização**. As perguntas realizadas encontram-se descritas na **Tabela 6**.

Dentre os 16 participantes, obtivemos uma baixa participação nas questões abertas. Apenas 4 respostas relevantes na primeira pergunta e 6 na segunda. Porém, obtivemos respostas valiosas que auxiliaram na continuidade da pesquisa.

Em relação aos pontos de melhoria na comunicação organizacional, foi possível destacar três pontos relevantes:

- A necessidade de tornar mais claro qual o objetivo e o porquê de determinadas metas/ações passadas às equipes de produtos pela arquitetura. Segundo o desenvolvedor P3, "*[Deveriam ser comunicados] qual o objetivo e [o porquê de] como as decisões [arquitetônicas que] foram tomadas sobre as definições estruturais [do código] e metas estabelecidas [foram repassadas para as equipes ou interessados].*"
- Não é notório, nem de fácil acesso, o compartilhamento ativo de conhecimento entre as equipes de produtos e outras áreas, como arquitetura, DevOps e outras. Segundo o desenvolvedor P10, "*[É necessário trazer melhorias na] comunicação entre os times de outras áreas [a qual] deveria ser mais fácil.*";
- É necessária a melhoria na comunicação entre as equipes de produtos e a equipe de arquitetura de software. De acordo com o desenvolvedor P6, "*[Deveriam ter como objetivo] incentivar a colaboração entre as equipes de desenvolvimento e promover o compartilhamento ativo de conhecimento por meio de workshops.*";

O outro questionamento realizado refere-se a buscar compreender, na visão dos desenvolvedores de produtos, quais informações são importantes de se ter fácil acesso dentro do time, tanto para desenvolvedores experientes quanto para novos integrantes do projeto, trazendo melhorias na fluidez da comunicação e no tempo gasto durante as rotinas do dia a dia.

A primeira categoria, sobre a Transferência de Conhecimento, relata comentários sobre as necessidades percebidas pelos desenvolvedores. Por exemplo, o desenvolvedor P8 relata a necessidade de mais "*documentação técnica e de fluxos básicos*". De forma mais abrangente, o desenvolvedor P12 diz que "*(...) acredita que vale a pena refletir sobre quais informações usamos com mais frequência para deixá-las em um local de fácil acesso. Existem também aquelas informações mais complexas, que, acredito, seria uma boa prática deixá-las salvas em algum lugar para que o time acesse no dia a dia.*"

A segunda categoria, sobre o onboarding (integração) de novos integrantes às equipes, o desenvolvedor P10 afirma a necessidade da disponibilidade de "*toda a documentação do projeto, técnica e de negócio, para que o novo integrante consiga ficar por dentro o mais rápido possível.*"

A terceira categoria, relatando sobre padrões organizacionais para os projetos, traz relatos de necessidades que podem ser fornecidas pelo time de arquitetura. O de-

desenvolvedor P7 afirma que *"acredita que falte um boilerplate/scaffold padronizado para que os times possam iniciar novos projetos de forma ágil e adequada, assim como um suporte maior da arquitetura com relação ao entendimento da arquitetura proposta."* Complementando, temos o relato de P3, que diz que *"padrões de projeto, estrutura e metodologia de trabalho são pontos que sempre ficam nebulosos na empresa."*

5.2.3 Conclusão sobre o Contexto do Problema

Após o levantamento de informações com gestores e arquitetos, bem como a avaliação diagnóstica com os desenvolvedores de produto, foi possível obter uma visão mais clara dos desafios enfrentados pela organização.

Ao analisar os canais já existentes para a distribuição e armazenamento de conhecimento arquitetônico e seus elementos funcionais e não funcionais, conclui-se que o conhecimento organizacional está presente. No entanto, a ausência de um canal centralizador e a falta de *roadmaps* que facilitem a compreensão do usuário sobre onde encontrar informações específicas tornam o acesso, o compartilhamento e a geração de novos conhecimentos processos pouco intuitivos e muito complexos.

A avaliação diagnóstica realizada com os desenvolvedores confirmou essa realidade, evidenciando a necessidade de um método eficiente para a centralização das informações relevantes ao processo de desenvolvimento de software.

A falta de um padrão de comunicação entre o time de arquitetura e os times de produto dificulta a agilidade no desenvolvimento de software. Dessa forma, é essencial compartilhar entre as equipes de produto os *Registros de Decisões Arquitetônicas* (RDAs) (vide **Seção 2.2**) sobre mudanças estruturais e sugestões de novas metodologias que possam auxiliar os times de produto a tomarem decisões mais assertivas.

A subutilização do Conhecimento Arquitetônico pode acarretar diversos problemas organizacionais, conforme apontado por Borrego et al. (2019), tais como requisitos e soluções técnicas mal compreendidos, falta de transferência de conhecimento entre equipes, ausência de visibilidade no monitoramento de projetos e desperdício de tempo por especialistas ao responder repetidamente às mesmas perguntas ou ao tentar solucionar problemas já resolvidos. Como apontado por P7, há uma necessidade de *"(...) um suporte maior da arquitetura em relação ao entendimento da metodologia de trabalho proposta"*.

Portanto, podemos concluir que foi possível compreender a visão dos times sobre a Gestão do Conhecimento Arquitetônico e levá-los a perceber que o conhecimento

já existe, mas que melhorias em sua distribuição e compartilhamento podem contribuir significativamente para a qualidade do código e para a agilidade na entrega das atividades de desenvolvimento.

5.3 Atividades de Planejamento da Solução

Nos tópicos anteriores, foi possível analisar que a organização possui conhecimento arquitetônico armazenado, abrangendo tanto requisitos funcionais quanto não funcionais. No entanto, conforme apontado, esse conhecimento encontra-se distribuído em diversos canais, o que dificulta o acesso em momentos oportunos, especialmente para novos integrantes da equipe. Isso se aplica tanto a profissionais externos, que precisam compreender o novo cenário de trabalho, quanto a internos, que, ao serem realocados para outros produtos, necessitam assimilar a nova forma de atuação.

Jansen, Bosch and Avgeriou (2009a) afirmam que:

"A falta de um repositório centralizado de conhecimento arquitetônico pode levar à perda de informações valiosas, dificultando a reutilização de decisões anteriores e resultando em ineficiências no desenvolvimento de novos projetos".

Com essa perspectiva, a centralização do conhecimento existente se mostra como uma solução que pode trazer melhorias ao CA tornando-o mais prático, além da facilitação para a geração de novos saberes e seu compartilhamento ágil entre os times e os novos integrantes da organização.

A centralização do conhecimento arquitetônico (CA) em um único canal facilita o acesso e o compartilhamento das decisões tomadas, garantindo que todos os stakeholders possam consultar e compreender as bases do projeto (JANSEN; BOSCH, 2005). Dessa forma, é possível atender a todos os grupos envolvidos no projeto, como desenvolvedores, líderes de negócio e arquitetos. A gestão do conhecimento arquitetônico (GCA) precisa ser apoiada por ferramentas que centralizem essas informações, permitindo que as decisões arquitetônicas sejam documentadas, compartilhadas e mantidas ao longo do ciclo de vida do projeto (KRUCHTEN, 2004).

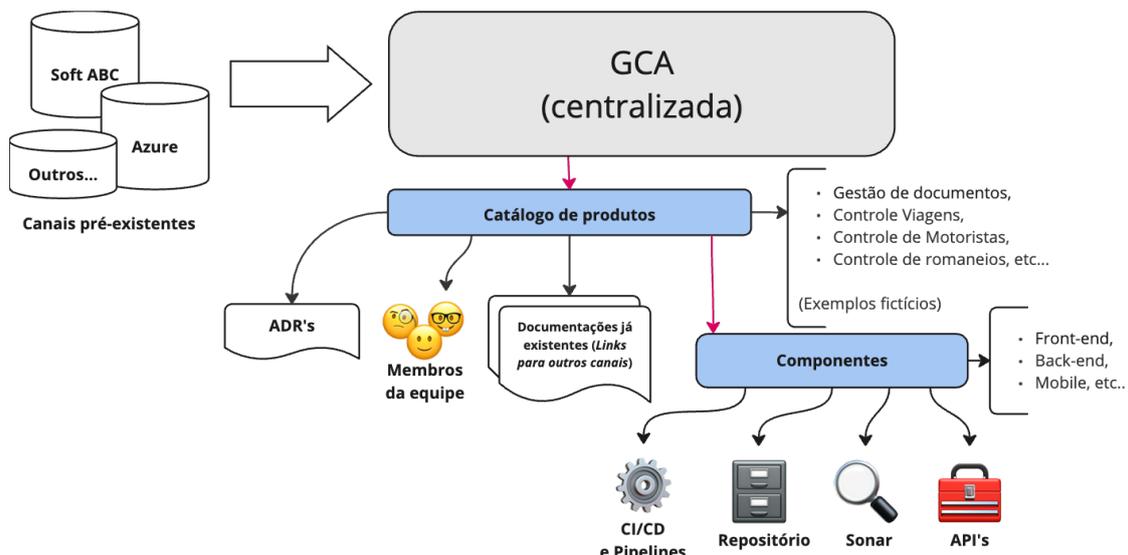
Além disso, a centralização do conhecimento arquitetônico promove transparência e consistência, reduzindo o risco de decisões conflitantes ou duplicadas (TANG; HAN; VASA, 2009). Com base nessa necessidade, foi definida a ação proposta para a realização

dessa centralização da GCA na organização.

5.3.1 Modelo proposto para centralização da GCA

O primeiro passo adotado para a execução dessa solução foi a análise dos canais e conteúdos pré-existentes na organização (vide **Sessão 5.2.1**), com o objetivo de sistematizar sua distribuição entre as equipes. Em seguida, definiu-se um **modelo** para a aplicação da experiência dos arquitetos na extração e formalização de conhecimentos contidos em dados não estruturados, permitindo encontrar, compartilhar e gerenciar essas informações de forma prática (DEMIRSOY; PETERSEN, 2018).

Figura 10 – Modelo para centralização da GCA.



Fonte: o autor.

O modelo adotado, conforme apresentada na **Figura 10**, permite que as equipes tenham acesso a um único canal principal de **GCA (centralizada)**, contendo requisitos funcionais e não funcionais de arquitetura para auxiliar no processo de desenvolvimento de software. As documentações já existentes na organização não são descartadas; pelo contrário, são avaliadas e redistribuídas no novo canal, conferindo maior agilidade à aplicação do processo de melhoria proposto.

O acesso principal ocorre por meio do **Catálogo de produtos**, onde estão listados todos os produtos atendidos na *Tribo Logística*. Cada **Produto** possui acesso aos *Architecture Decisions Records* (ADRs) definidos pelo time de arquitetura, bem como às

informações de contato e papel de cada **membro da equipe**. Além disso, são disponibilizados links específicos para **Documentações pré-existentes**, como decisões arquiteturais, diagramas de classe, requisitos funcionais e não funcionais, entre outros conteúdos pertinentes à equipe.

Cada produto é composto por **Componentes**, que correspondem às aplicações específicas, como *Front-end*, *Back-end*, *Aplicativos móveis*, entre outros. Dentro de cada componente, deve-se garantir um acesso rápido ao controle de **CI/CD e pipelines**, sendo que, nesta organização, utiliza-se o Azure DevOps® como ferramenta. Também estão disponíveis o **Repositório** contendo o código-fonte, o **SonarQube®** específico do componente, bem como a documentação e os testes de API do produto. Para controle e compartilhamento das **API's**, esta organização utiliza o Swagger®¹¹.

Definido a ação a ser tomada é necessário um software que permita tal aplicação desta teoria de forma prática e funcional.

5.3.2 Ferramenta sugerida para a centralização da GCA

Ao analisar o modelo proposto, percebe-se a necessidade de uma ferramenta para a integração dos canais existentes e a manutenção dessas informações. É essencial proporcionar uma visão ampla e acessível de todos os produtos e suas integrações, considerando que fazem parte de um sistema SaaS.

A organização já utiliza uma ferramenta que atende aos requisitos da proposta: o Backstage®. Atualmente, ele é administrado por uma equipe externa à *Suíte Logística*, que apoia os times de produto que utilizam modelos SaaS na organização. Essa equipe realiza ajustes iniciais nas aplicações e oferece suporte estrutural na criação de novos repositórios e pipelines de CI/CD. No entanto, a ferramenta não é utilizada de forma adequada, embora esteja alinhada à proposta desta pesquisa.

O Spotify Open Source Team (2025) afirma que o Backstage® tem como "objetivo principal unificar todas as ferramentas, serviços e documentações de uma organização em uma interface consistente, facilitando o acesso e a gestão desses recursos pelos desenvolvedores".

O Backstage® permite a criação de modelos padronizados para novas aplicações e serviços, garantindo que as arquiteturas sigam as melhores práticas da organização. Modelos estruturados reforçam a conformidade arquitetônica e reduzem desvios no de-

¹¹<https://swagger.io/docs/>

envolvimento de software em larga escala (JANSEN; BOSCH; AVGERIOU, 2009b).

A documentação de decisões arquitetônicas RDA (Registro de Decisões Arquitetônicas – *Architecture Decision Records*), bem como a definição de padrões e boas práticas, é parte essencial da GCA. Uma RDA descreve as circunstâncias técnicas, comerciais, sociais ou políticas que influenciam diretamente uma decisão arquitetural de projeto (KEELING, 2022) (Vide **Sessão 2.2**)

Diante da aderência da ferramenta à proposta desta pesquisa, optou-se por centralizar o Conhecimento Arquitetural (CA) utilizando o Backstage®, pois ele viabiliza a aplicação dos conceitos da GCA ao:

- Monitorar e integrar-se a sistemas de repositórios de produtos para gerenciar suas atividades diárias, uma característica essencial de um GCA (BHAT et al., 2016);
- Permitir a representação do conhecimento do projeto de arquitetura utilizado ou gerado durante a análise da arquitetura de software, atendendo a um dos princípios da GCA (CAPILLA et al., 2016a);
- Viabilizar a representação de LDA (Linguagem de Decisão Arquitetural – *Architecture Decision Language*), permitindo que os arquitetos expressem sua visão de forma prática e dinâmica, conforme defendido por Oquendo (2016) para modelos de GCA;
- Sincronizar informações presentes em diferentes repositórios (SPOTIFY, 2024).

5.3.3 Planejamento e desenvolvimento de estratégias e ações

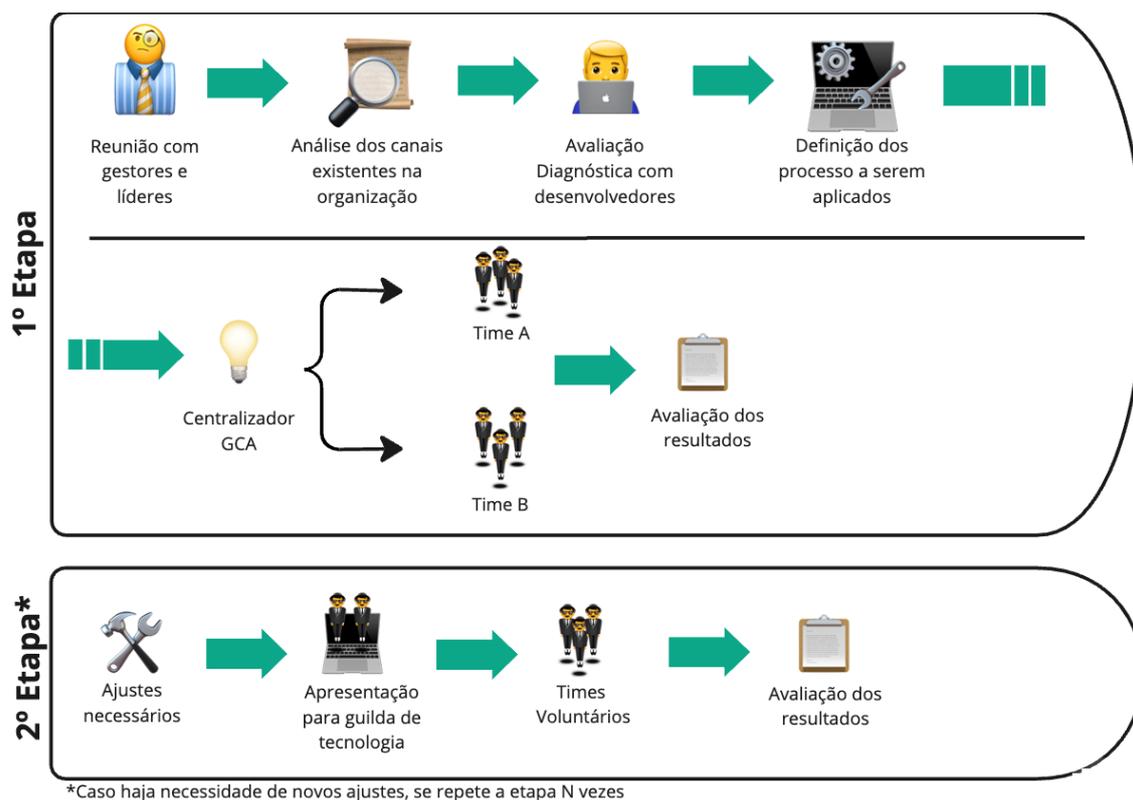
Para a aplicação do modelo proposto, foi estruturado um roteiro para execução da pesquisa-ação, conforme demonstra a **Figura 11**. Definiu-se de antemão a execução de, no mínimo, dois ciclos. Porém devido ao dinamismo do método, poderão surgir outros ciclos posteriormente.

Na fase inicial, houve uma entrevista com arquitetos e gestores (Vide Seção 5.1) para compreender juntamente com eles qual a importância do tema e a compreensão deles a respeito. Foi repassado os canais já existentes na organização e se havia algum padrão de processo a ser seguido.

Logo após, foi realizada uma análise diagnóstica quali-quantitativa com dois times (selecionados da suíte logística) (**vide Seção 5.2.2.2**) por meio da aplicação de ques-

tionários. A análise quantitativa teve como objetivo trazer a visão que os participantes têm a respeito da GCA e sua importância na organização, já a análise qualitativa buscou compreender pontos de melhoria e quais são os tipos de CA importantes que devem ser compartilhados na visão dos times de produto.

Figura 11 – Processo adotado para aplicação do modelo proposto.



Fonte: o autor.

Com a análise das informações coletadas, foi possível definir os elementos necessários para o desenvolvimento do processo de centralização da GCA na organização. Com essa definição em mãos, foi possível definir qual software se adequa mais a ser utilizado como ferramenta nesta pesquisa.

A próxima etapa consiste na realização de um breve treinamento mostrando como a GCA centralizada poderá auxiliar no dia a dia e como alimentá-la quando necessário. A aplicação da GCA será realizada pelos dois times que participaram da *Avaliação Diagnóstica*. Após certo tempo de uso da ferramenta no dia a dia é realizado uma avaliação diagnóstica através de conversas com integrantes dos dois times, sobre a utilização da ferramenta nesse período e validar a eficiência do processo proposto no dia a dia.

na execução da próxima etapa, devem ser realizados alguns **ajustes necessários** ao

processo e a ferramenta a qual está sendo utilizada, e com isso, foi apresentada à "*Guilda de Tecnologia*", que é o encontro semanal de todos os Líderes Técnicos (*Techleads*), Arquitetos de Software e gestores da suíte logística, a proposta da pesquisa assim como os resultados obtidos com as equipes que participaram na primeira etapa.

Sendo, em sequência, aberto a todos os times a opção de se voluntariarem a realizar testes da ferramenta de GCA proposta nesta segunda etapa. Após essa apresentação e a utilização da ferramenta pelos novos voluntários, após certo tempo de uso, é realizado então, uma nova avaliação quali-quantitativa, seguindo o modelo TAM (VENKATESH; DAVIS, 2000) para compreender a aceitação do processo aplicado através da centralização da GCA.

O modelo de questionário TAM foi desenvolvido para capturar a percepção dos participantes sobre os principais motivos que levam os usuários a aceitar ou rejeitar uma determinada tecnologia.

Os indicadores avaliados pelo questionário TAM segundo Venkatesh and Davis (2000) são:

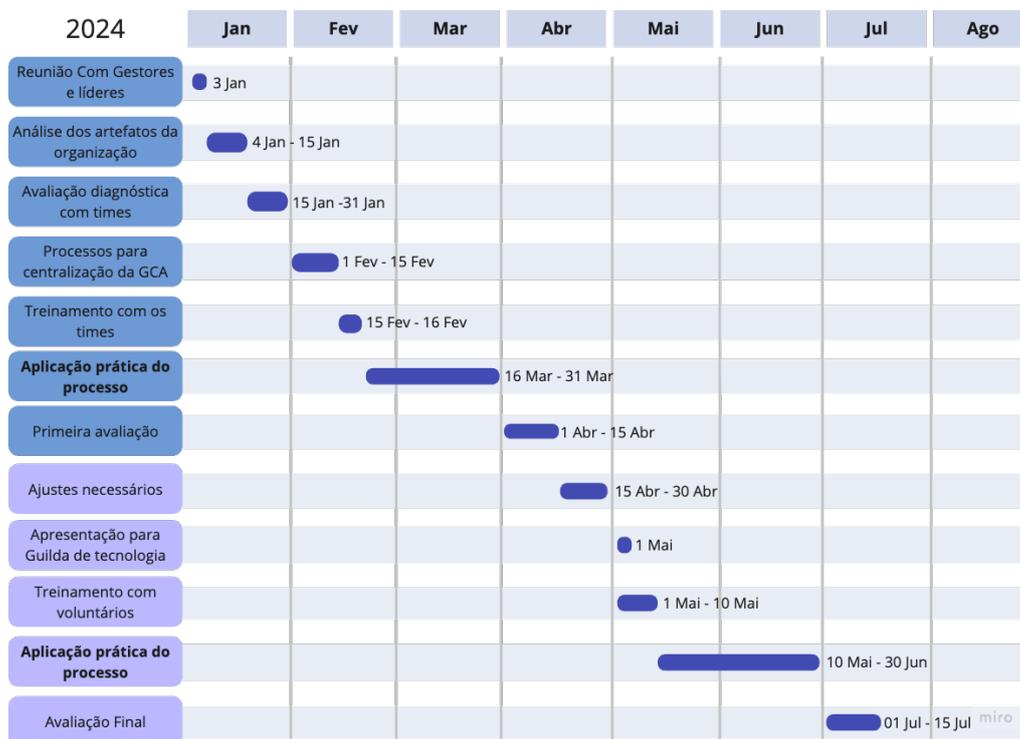
- **Utilidade Percebida**, que define o grau em que o participante acredita que o uso de determinada tecnologia pode melhorar seu desempenho no trabalho;
- **Facilidade de Uso Percebida**, que define o grau em que o participante acredita que a adoção de uma determinada tecnologia poderá ser livre de muito esforço;
- **Intenção de Uso Percebida**, que define o grau em que o participante acredita que poderá utilizar uma determinada tecnologia no futuro. Este ciclo deve se repetir até que o objetivo proposto seja alcançado.

Como medida de sucesso nos questionários avaliativos foi utilizado o padrão likert para mensurações dos resultados obtidos. E através das questões qualitativas, entrevistas e observações, a medida para o sucesso se baseou nos apontamentos realizados pelos usuários da ferramenta.

5.3.4 Cronograma

O período de análise junto com a organização e resolução do problema encontrado foi realizado entre Janeiro e Julho de 2024. A divisão de cada etapa conforme relatado na **Seção 5.3.3** foi dividida conforme descritas na **Figura 12**

Figura 12 – Cronograma para implantação do processo de GCA.



Fonte: o autor.

5.3.5 Conclusão do planejamento da solução

A GCA deve representar a integração da arquitetura de software de um sistema ou de uma família de sistemas, juntamente com as decisões arquiteturais, suas justificativas, influências externas e o ambiente de desenvolvimento (OZKAYA; WALLIN; AXELSSON, 2010). Diversos autores defendem a implementação da GCA em uma única ferramenta, especialmente em contextos nos quais o conhecimento está distribuído em múltiplos repositórios e há grande dificuldade na sua recuperação (BOER et al., 2007; RAZAVIAN; PAECH; TANG, 2023; ELORANTA; KOSKIMIES, 2014; CHANG et al., 2021; CLERC, 2008; TANG et al., 2010).

Após a conclusão desse planejamento, iniciaram-se os ciclos de implementação, monitoramento e validação dos resultados gerados no processo.

5.4 Orientações para aplicação do modelo proposto (*guideline*)

Nesta etapa, são fornecidas diretrizes para a implementação do modelo proposto, garantindo que as informações sejam centralizadas, acessíveis e continuamente atualizadas. Dessa forma, busca-se estabelecer um conjunto de diretrizes para a implementação e aprimoramento da Gestão do Conhecimento Arquitetônico (GCA), promovendo um processo colaborativo e ágil dentro das organizações de software.

A proposta é direcionada para os seguintes profissionais:

- Arquitetos de Software;
- Desenvolvedores de produto;
- Gestores de Tecnologia;
- Profissionais envolvidos na documentação e manutenção de sistemas.

5.4.1 Princípios a serem adotados

Para garantir a aplicação eficiente do modelo proposto, recomenda-se seguir os seguintes princípios:

- **Centralização do conhecimento:** Todos os artefatos arquitetônicos devem estar armazenados em um repositório unificado;
- **Acessibilidade e Transparência:** O acesso deve ser facilitado para todos os membros relevantes da equipe;
- **Manutenção Contínua:** O conhecimento deve ser revisado e atualizado regularmente;
- **Automatização e Integração:** Ferramentas como o Backstage® devem ser utilizadas para facilitar a documentação automática. No entanto, é essencial verificar se a organização já possui uma ferramenta que atenda ao escopo sugerido, garantindo maior aderência ao modelo proposto;
- **Engajamento da Equipe:** A cultura organizacional deve incentivar o compartilhamento de conhecimento de forma ativa e contínua.

5.4.2 Etapas da implementação

- **Diagnóstico Inicial:** Realizar o mapeamento do estado atual da GCA na organização; Identificar principais pontos de dificuldades encontrados pela equipe no acesso e utilização desse conhecimento;
- **Implementação do Modelo Proposto:** Definir uma ferramenta adequada para se tornar o repositório central (ex.: Backstage®, Confluence, Wiki corporativa); Criar um padrão de documentação estabelecendo templates e guias de escrita como os RDAs; Integrar a documentação ao ciclo de desenvolvimento através do incentivo a criação de documentação junto ao desenvolvimento; Treinar a equipe, promovendo workshops e guias de uso da plataforma escolhida;
- **Monitoramento e Melhoria Contínua:** Estabelecer revisões periódicas do conteúdo; Criar um processo de feedback contínuo para aprimoramento do modelo; Implementar mecanismos de automação para garantir atualização constante.

Para avaliar o impacto da implementação do modelo, é essencial adotar métodos qualitativos e quantitativos. Deve-se observar se a equipe percebe melhorias no tempo gasto na busca por informações e se isso impacta positivamente o código desenvolvido. Além disso, é fundamental incentivar a adoção do modelo de GCA como parte da cultura organizacional, sistematizando um processo de curadoria de conhecimento para evitar desatualizações.

Para garantir a eficácia da *guideline* proposta, utilize os seguintes indicadores:

- **Taxa de adesão:** Quantidade de membros que acessam e contribuem para a documentação.
- **Tempo de resposta:** Redução no tempo gasto para encontrar informações arquiteturais.
- **Feedback da equipe:** Pesquisas internas sobre a usabilidade da GCA.

A implementação de um modelo eficiente de GCA pode gerar ganhos significativos para a organização, tornando o compartilhamento de conhecimento arquitetural mais ágil e estruturado. O engajamento da equipe e a adoção contínua são fatores essenciais para o sucesso dessa iniciativa. Por fim, é crucial que o modelo proposto seja refinado continuamente com base nos feedbacks recebidos, além da exploração de novas ferramentas e metodologias que possam aprimorá-lo ainda mais.

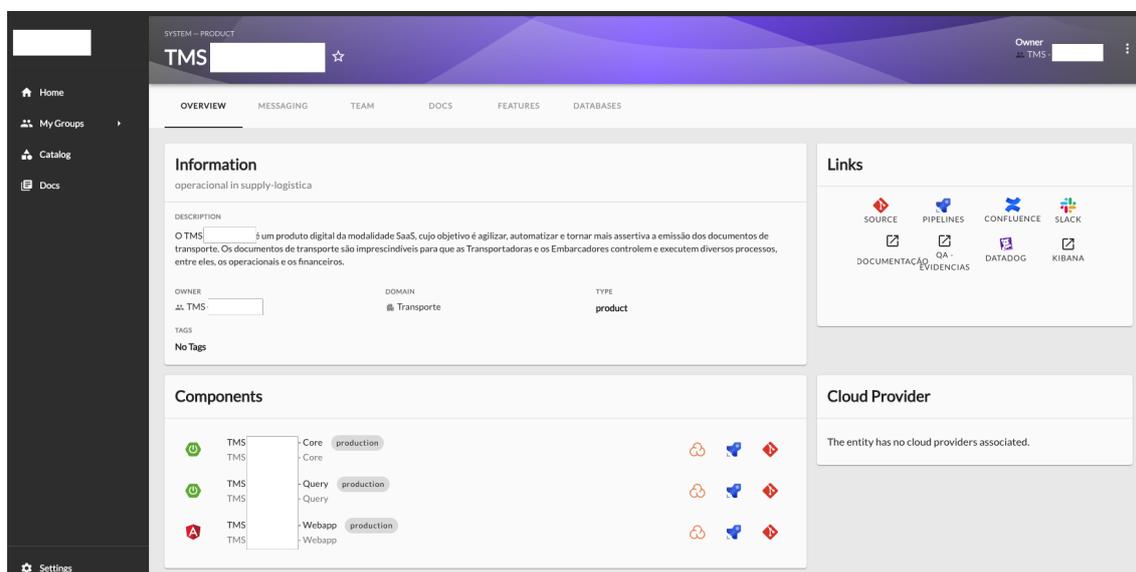
6 PRIMEIRO CICLO DO PROCESSO DE PESQUISA-AÇÃO

Neste capítulo, será demonstrado a execução do primeiro ciclo da pesquisa-ação, a qual aplicamos o processo prático de centralização da GCA em duas equipes, com o objetivo de validar sua eficiência. Esse ciclo abrange as etapas 4, 5, 6 e 7, que serão descritas a seguir.

6.1 Implementação da solução proposta

Nesta etapa, foi implementado o roteiro definido no capítulo anterior. No entanto, como afirma Thiollent (2011), a pesquisa-ação requer o envolvimento ativo dos participantes, incluindo dirigentes e técnicos, que devem consentir e colaborar no processo investigativo, uma vez que as ações propostas interferem na realidade organizacional e dependem da participação coletiva para sua efetividade.

Figura 13 – Apresentação do produto no Backstage®.



Fonte: o autor.

Para a adaptação do Backstage®, conforme apresentado na **Figura 13**, contou-se com a participação do arquiteto Miguel Antunes ¹, responsável pela ferramenta dentro da *Suíte Logística*. Embora a manutenção do Backstage® fosse atribuída a outro setor, Miguel demonstrou interesse na ferramenta, mas carecia da motivação necessária para

¹Nome fictício

aprimorá-la. Em conversas não formalizadas, ele destacou: *"Podemos melhorar, e muito, a eficiência dos desenvolvedores com a ferramenta. Só nos faltava essa visão metódica, oferecida nesta pesquisa, para buscarmos melhorar a maneira de alimentá-la para o uso."*

Um dos fatores que facilitaram o processo foi o fato de que o Backstage® já estava disponível e possuía o **Catálogo de Produtos** da *Suíte Logística* pronto. No entanto, era necessário ajustá-lo adequadamente com o conhecimento já existente na organização. Neste primeiro momento, foram alimentadas as informações referentes aos dois times envolvidos na execução do primeiro ciclo.

As decisões arquitetônicas e a documentação do produto foram redirecionadas dos canais existentes para o **Catálogo de Produtos**. Para os **componentes**, a parte de CI/CD já estava configurada, assim como o acesso aos repositórios do Azure DevOps® e às APIs, disponibilizadas via Swagger®.

As decisões arquitetônicas foram registradas utilizando o modelo Markdown, conforme demonstrado na **Figura 14**. O Markdown é uma ferramenta eficiente para a escrita de documentação técnica e científica, pois permite uma formatação simples e estruturada, sem a complexidade de linguagens mais robustas, além de facilitar a conversão para diversos formatos web (KANE; URBANEK, 2020). Dessa forma, sua adaptação para visualização no Backstage® tornou-se prática e de fácil implementação.

Figura 14 – Modelo de ADR (Architecture Decision Recorder)*

```

# ADR 1: Criação de um service base para reutilização

## Status
Aceito

## Data
2023-06-24
#1 for Command, ML for Cascade

## Decisores
- [redacted] Arquiteto Front-end
- [redacted] Desenvolvedor Sênior

## Consultados
- [redacted] Líder de Equipe

## Informados
- Todos os Membros do Projeto

## Contexto e Declaração do Problema
No ciclo de vida do nosso desenvolvimento de software, percebemos a
necessidade de um service com a estrutura básica de um CRUD com os verbos http
utilizados com certa frequência no projeto. Com esse intuito criamos a classe
base.service.ts para ser implementado pelos serviços básicos utilizados no
projeto.

## Diretrizes da Decisão
* A possibilidade de realizar reuso de código.
* Diminuir a duplicidade de códigos com a mesma funcionalidade.
* Tornar o código essencial mais limpo.

## Opções Consideradas
* **Criação de classe genérica para reuso**
* **Documentação em Comentários de Código**

## Resultado da Decisão
Opção escolhida: A criação dessa classe dentro do sistema irá evitar a
duplicação de pelo menos outras 30 classes que seguem um padrão básico de
utilização dos verbos HTTP.

## Consequências
* Bom, minimiza a possibilidade de se ter códigos duplicados dentro do sistema
pois se torna uma extensão para novas classes, e permite a inserção de novas
funcionalidades quando necessário.

```

ADR 1: Criação de um service base para reutilização

Status

Aceito

Data

2023-06-24

Decisores

- [redacted] Arquiteto Front-end
- [redacted] Desenvolvedor Sênior

Consultados

- [redacted] Líder de Equipe

Informados

- Todos os Membros do Projeto

Contexto e Declaração do Problema

*RDA (Registro de Decisão Arquitetônica) - Fonte: o autor.

Somente foi implementado via código-fonte o acesso síncrono aos dados do SonarQube® e às informações consideradas relevantes pelo time de arquitetura, como requisitos não funcionais e documentações técnicas oficiais referentes às metodologias adotadas no desenvolvimento. Um exemplo disso é a utilização do modelo CQRS®² no *back-end*. Além disso, foi acrescentada a composição das equipes, com informações sobre os papéis e especialidades dos desenvolvedores e líderes técnicos e de negócio.

Após a realização dos ajustes necessários no Backstage® para o início do monitoramento das ações implementadas, foi feita uma apresentação aos arquitetos Luiz Fernando³, responsável pelo *front-end*, e Antônio Afonso³, responsável pelo *back-end*, para validar suas funcionalidades.

Durante a apresentação, o arquiteto Antônio Afonso³ afirmou: "É muito bom poder facilitar o acesso às nossas ADRs pelos times de produto e, dessa forma, garantir uma linearidade técnica durante o desenvolvimento dos produtos em toda a Suíte Logística."

6.2 Monitoramento da solução

Nesta fase de monitoramento, o pesquisador realizou a implementação da centralização da GCA no Backstage® e a adaptação da ferramenta ao modelo proposto. A validação pelo time de arquitetura foi um passo essencial para garantir a adesão ao novo processo.

O foco principal desta etapa foi analisar a experiência das equipes no uso do Backstage® e identificar possíveis dificuldades ou melhorias. A interação contínua com os desenvolvedores permitiu compreender como a solução se integrava ao fluxo de trabalho diário.

As equipes tiveram um período de 15 dias para testes, conforme apresentado na **Figura 12**, durante os quais foram questionadas, aleatoriamente, sobre o uso da ferramenta e a adequação do novo processo ao dia a dia. Esse acompanhamento permitiu avaliar a aceitação da solução e identificar eventuais dificuldades.

Com base na vivência com esse novo processo da GCA, foram coletadas percepções sobre sua eficiência e impacto na organização do conhecimento arquitetônico. Esses insights contribuíram para ajustes e refinamentos no modelo proposto.

²<https://java-design-patterns.com/patterns/command-query-responsibility-segregation/>

³nome fictício

6.3 Avaliação e aperfeiçoamentos necessários

A fase de avaliação e geração de resultados no método pesquisa-ação desempenha um papel fundamental na análise das ações implementadas e de seus impactos na organização, possibilitando que os participantes adquiram conhecimento a partir da experiência. Essa etapa não se limita à mensuração de resultados, mas também fomenta a reflexão crítica e o engajamento colaborativo entre os envolvidos, fortalecendo a capacidade organizacional de responder de maneira adaptativa a desafios e demandas emergentes (THIOLLENT, 2011).

Para garantir a eficácia da implementação do novo processo de Gestão do Conhecimento Arquitetônico (GCA) por meio do método pesquisa-ação, conforme sugerido por Staron (2020), faz-se necessário preparar o ambiente organizacional de forma a minimizar impactos nas rotinas diárias. Dessa maneira, o Backstage® foi devidamente adaptado ao modelo proposto e recebeu a aprovação do time de arquitetura. Foram agendadas duas apresentações, direcionadas às equipes participantes da avaliação da proposta.

A pesquisa-ação caracteriza-se pela sua natureza interativa e colaborativa, promovendo ajustes contínuos a partir da incorporação de novas informações e eventos (THIOLLENT, 1997). Em consonância com essa abordagem, foram realizadas interações frequentes com os participantes ao longo do processo.

No entanto, considerando que os times estavam comprometidos com entregas previamente programadas durante o período de avaliação e que o método pesquisa-ação visa interferir minimamente nas operações da organização, optou-se por basear a avaliação, neste momento, nos relatos espontâneos dos participantes, coletados em interações informais no ambiente de trabalho.

O principal objetivo dessa etapa foi avaliar a viabilidade da utilização da ferramenta em contextos estratégicos, como na criação de novos componentes, garantindo a aderência às decisões arquiteturais e ao modelo estrutural adotado no desenvolvimento dos produtos de software. No entanto, buscou-se, sobretudo, compreender seu impacto no cotidiano dos desenvolvedores e na dinâmica de interação entre as equipes técnicas.

As equipes envolvidas no estudo apresentam formações heterogêneas em termos de papéis e níveis de experiência, conforme demonstrado na **Tabela 7**. Essa diversidade representa um fator relevante para a validação da proposta, pois permite analisar diferentes perspectivas e avaliar como tais variações influenciam o engajamento e a adoção do novo processo.

Durante as reuniões, foi verificado se os participantes compreendiam bem o tema abordado, com foco nos seguintes pontos:

- O conceito teórico e prático da GCA;
- Qual o objetivo em se centralizar a GCA, conforme apresentando na **Figura 10**;
- Os benefícios dessa abordagem no dia a dia do desenvolvedor;
- A importância do comprometimento de todos para a implementação do projeto;
- A facilidade de manipular os itens sem demandar muito tempo, tornando a manutenção mais ágil.

Tabela 7 – Participantes por equipe

| Papel | Pessoa | Projetos | Cargo* | empresa* | Profissional* |
|---------------------|--------|----------|--------|----------|---------------|
| <i>Time A</i> | | | | | |
| <i>Júnior</i> | P6 | 1 | 1-5 | 1-5 | 1-5 |
| <i>Pleno</i> | P5 | 1 | 5-10 | <1 | >510 |
| <i>Sênior</i> | P1 | >3 | >10 | >10 | >10 |
| | P4 | 2-3 | 5-10 | 1-5 | 5-10 |
| <i>Especialista</i> | P3 | 1 | 1-5 | 1-5 | >5 |
| | P4 | 2-3 | 5-10 | 1-5 | 5-10 |
| | P7 | 1 | 1-5 | 1-5 | >10 |
| | P8 | 2-3 | <5 | 1-5 | 5-10 |
| | P9 | >3 | 1-5 | >10 | >10 |
| <i>Time B</i> | | | | | |
| <i>Pleno</i> | P2 | 1 | >5 | 1-5 | >10 |
| | P11 | 2-3 | <5 | 1-5 | >5 |
| <i>Sênior</i> | P10 | 2-3 | 5-10 | 1-5 | 5-10 |
| | P12 | 1 | <1 | <1 | 1-5 |
| <i>Especialista</i> | P13 | 2-3 | >5 | 1-5 | 5-10 |
| | P14 | 2-3 | 1-5 | 1-5 | >10 |
| | P15 | 2-3 | 1-5 | 1-5 | >10 |
| | P16 | 2-3 | 1-5 | 1-5 | >10 |
| * Medida em anos | | | | | |

Segundo o participante **P4**:

"Ter informações importantes para o dia a dia centralizadas em um único lugar fomenta a criação de documentação pela equipe, o que auxilia muito no desenvolvimento, especificação e controle de qualidade do software do time."

Esse relato trouxe um viés de importância para o assunto, e diversos participantes aprovaram a iniciativa da pesquisa.

O desenvolvedor **P10**, ao final da apresentação, levantou uma questão pertinente:

"Temos muita dificuldade quando estamos em uma fase decisiva do projeto e novos integrantes são inseridos na equipe. O tempo de adaptação e aprendizado do nosso modelo de trabalho é um desafio. Como o processo sugerido pode nos ajudar?"

Esse questionamento revelou-se especialmente oportuno, pois, durante a execução dessa etapa, houve a inserção de novos membros nas equipes, conforme demonstrado na **Tabela 8**. Para acelerar as entregas, foram adicionados novos integrantes às duas equipes, incluindo a realocação de P8 do **Time A** para o **Time B**.

Em resposta ao questionamento de P10, o pesquisador destacou que:

"os novos integrantes podem acessar, em um único local, informações técnicas sobre a estrutura do código-fonte, a metodologia adotada por meio das RDAs e o funcionamento do processo de publicação dos desenvolvimentos, disponibilizado via CI/CD na ferramenta".

Além disso, informações sobre o negócio e requisitos não funcionais foram incorporadas à página do produto, facilitando o entendimento e a integração dos recém-chegados.

Tabela 8 – Novos integrantes equipe

| Papel | Pessoa | Projetos | Cargo* | empresa* | Profissional* |
|---------------------|--------|----------|--------|----------|---------------|
| <i>Time A</i> | | | | | |
| <i>Sênior</i> | P18 | 1 | >4 | <1 | >7 |
| | P19 | 1 | >5 | <1 | 5-10 |
| | P20 | 1 | >5 | <1 | 5-10 |
| <i>Especialista</i> | P21 | 1 | >3 | <1 | >10 |
| <i>Time B</i> | | | | | |
| <i>Sênior</i> | P22 | 1 | 5-10 | <1 | 5-10 |
| <i>Especialista</i> | P23 | 1 | >5 | <1 | 5-10 |
| | P8 | 2-3 | 1-5 | 1-5 | >10 |
| * Medida em anos | | | | | |

A percepção positiva dos novos integrantes reforçou a eficácia da proposta. O desenvolvedor **P19** afirmou:

"Muito bom ter toda a informação necessária para iniciar um projeto em apenas um canal. Isso é um assunto que deveria ser mais valorizado por outras empresas."

De forma semelhante, **P23** destacou:

"É bem mais simples, ao ser integrado a uma nova organização, possuir apenas um canal com todas as informações necessárias para desenvolver nosso papel na equipe."

Entretanto, alguns desafios foram apontados. **P21** comentou:

"O fácil acesso ao controle de APIs deste produto facilitou minha atuação como back-end. Porém, senti falta da integração com os outros times, pois foi necessário realizar essa comunicação e tive dificuldades."

Esse apontamento reforça a importância da integração da documentação com o ciclo de desenvolvimento, conforme argumentado por Bass, Clements and Kazman (2021), é importante garantir que todas as partes envolvidas, incluindo outros sistemas, tenham acesso a informações atualizadas, especialmente em ambientes de entrega contínua.

Outro ponto relevante foi levantado pelos participantes P4, P5, P7 e P8, que sugeriram a definição de decisões arquitetônicas específicas para cada produto. Tal abordagem permitiria que integrantes vindos de outras equipes compreendessem melhor a estrutura do código-fonte e os princípios técnicos adotados. A heterogeneidade das decisões arquitetônicas, conforme discutido por Garlan, Allen and Ockerbloom (1995), é resultado das constantes mudanças em tecnologias, requisitos e práticas de desenvolvimento, especialmente em produtos criados em períodos distintos. Lidar com essa diversidade representa um desafio crítico, e o processo sugerido pode auxiliar na garantia da manutenibilidade, da consistência do sistema e na rápida integração de novos membros às equipes.

A diversidade de abordagens e estratégias de gestão da estrutura do código-fonte evidencia a necessidade de um trabalho colaborativo na formulação dos Registros de Decisões Arquitetônicas (RDAs). Quando essas decisões são tomadas conjuntamente entre o time de desenvolvimento e os arquitetos de software, promovem um alinhamento estratégico, facilitam a manutenção do sistema e reduzem o risco de inconsistências futuras (CAPILLA et al., 2016a). Além disso, a prática de registrar RDAs específicas para cada produto contribui para o amadurecimento técnico da equipe, sendo uma referência fundamental para novos integrantes.

Os arquitetos também ressaltaram a importância da geração de documentação completa dos produtos, pois tal documentação não apenas facilita a avaliação da aplicação das melhores práticas, mas também aprimora a disseminação do conhecimento dentro da organização. No entanto, enfatizou-se que essa documentação deve ser ágil e de fácil manutenção, demandando o menor tempo possível das equipes envolvidas na sua elaboração.

A colaboração contínua entre o time de arquitetura de software e o time de desenvolvimento é fundamental para assegurar que o código seja bem estruturado, padronizado e alinhado com os princípios arquiteturais previamente definidos (BASS; CLEMENTS; KAZMAN, 2021). Essa integração permite um acompanhamento mais próximo dos times de produto, promovendo a manutenção eficiente do sistema e reduzindo o risco de degradação arquitetural ao longo do tempo.

O participante P5, que atua como *Quality Assurance* (QA), sugeriu a facilitação do acesso aos testes automatizados, de modo que os próprios desenvolvedores possam conhecer os fluxos utilizados para validar as entregas dos times de produto. Essa sugestão ressalta a importância da transparência no processo de validação e da colaboração entre as equipes para aprimorar a qualidade do software produzido.

Neste momento, tornou-se evidente a existência de alguns pontos que demandam melhorias, os quais, até então, haviam passado despercebidos, principalmente devido ao foco inicial na estruturação e formalização do conhecimento organizacional existente. No que se refere às RDAs desenvolvidas pelos times de produto, observou-se que essa prática representa uma inovação dentro da organização, ainda que, de maneira informal, tais decisões já fossem tomadas anteriormente, sem o devido registro documental.

Em relação à documentação do código-fonte, tanto no front-end quanto no back-end, identificou-se que essa prática não era realizada de forma sistemática, uma vez que era considerada excessivamente dispendiosa em termos de tempo e esforço. Contudo, a possibilidade de compartilhar os testes automatizados foi apontada como uma alternativa viável para facilitar a validação das implementações realizadas pelos desenvolvedores, permitindo que eventuais falhas sejam detectadas antes mesmo da etapa de validação pelo QA.

Dessa forma, conclui-se que há subsídios concretos para o desenvolvimento da próxima etapa, que consiste no aperfeiçoamento da solução proposta, garantindo que os ajustes necessários sejam implementados de maneira eficiente e alinhada às necessidades da organização.

6.4 Conclusão do primeiro ciclo

O primeiro ciclo desta pesquisa-ação teve como objetivo principal validar, de forma prática, a aplicação de uma melhoria no processo de GCA (Gestão do Conhecimento Arquitetônico) da organização, por meio da centralização do conhecimento exis-

tente e da facilitação para a geração de novos conhecimentos.

Para isso, foram implementadas ações como a revisão do conhecimento arquitetônico existente, envolvendo seus requisitos funcionais e não funcionais. Como resultado, observou-se o interesse dos times em melhorar seus processos, principalmente levando em consideração a participação com sugestões de melhorias.

No entanto, alguns desafios foram identificados, como a dificuldade em trazer a cultura da geração de conhecimento por meio do processo sugerido nesta pesquisa. Conscientizar os participantes da simplicidade em inserir ações que gerarão facilidades no dia a dia e, principalmente, tornar mais clara a comunicação com o time de arquitetura é algo que exige esforço e fixação do assunto, tanto por parte dos gestores e líderes técnicos quanto dos próprios desenvolvedores.

Este ciclo permitiu o aprendizado de que a comunicação contínua entre arquitetos e desenvolvedores é crucial para a adoção de padrões. Além disso, ficou claro que, quando temos a GCA clara e acessível, isso facilita o dia a dia dos desenvolvedores em momentos cruciais como, por exemplo, a necessidade do *front-end* em localizar um contrato do *back-end*. Para o próximo ciclo, planeja-se validar as melhorias propostas no processo. O foco será consolidar a padronização na geração e no compartilhamento do Conhecimento Arquitetônico (CA), de modo a torná-lo parte da cultura organizacional.

7 SEGUNDO CICLO DO PROCESSO DE PESQUISA-AÇÃO

Neste capítulo, será discutido o segundo ciclo da pesquisa-ação, no qual pretende-se aplicar as melhorias propostas na **Seção 6.3** do ciclo anterior. Para validar o método de aplicação da GCA (Gestão do Conhecimento Arquitetônico), este segundo ciclo foi realizado nas equipes do primeiro ciclo e em outras equipes que se voluntariaram após uma apresentação para líderes técnicos e gestores da Suíte Logística. Este ciclo envolve uma segunda passagem pelas etapas de **Implementação, Monitoramento e Avaliação**, que serão descritas a seguir.

7.1 Implementação do aperfeiçoamento proposto

A fase de aperfeiçoamento na pesquisa-ação é de suma importância para garantir que as soluções propostas ocorram de forma contínua e adaptativa. Essa etapa permite que os participantes revisitem suas práticas, incorporem feedback e ajustem suas ações com base em resultados concretos, promovendo um ciclo constante de aprendizado e melhoria (TRIPP, 2005).

Com base no levantamento dos resultados obtidos na etapa anterior, foi de compreensão dos pesquisadores que os seguintes pontos deveriam ser desenvolvidos:

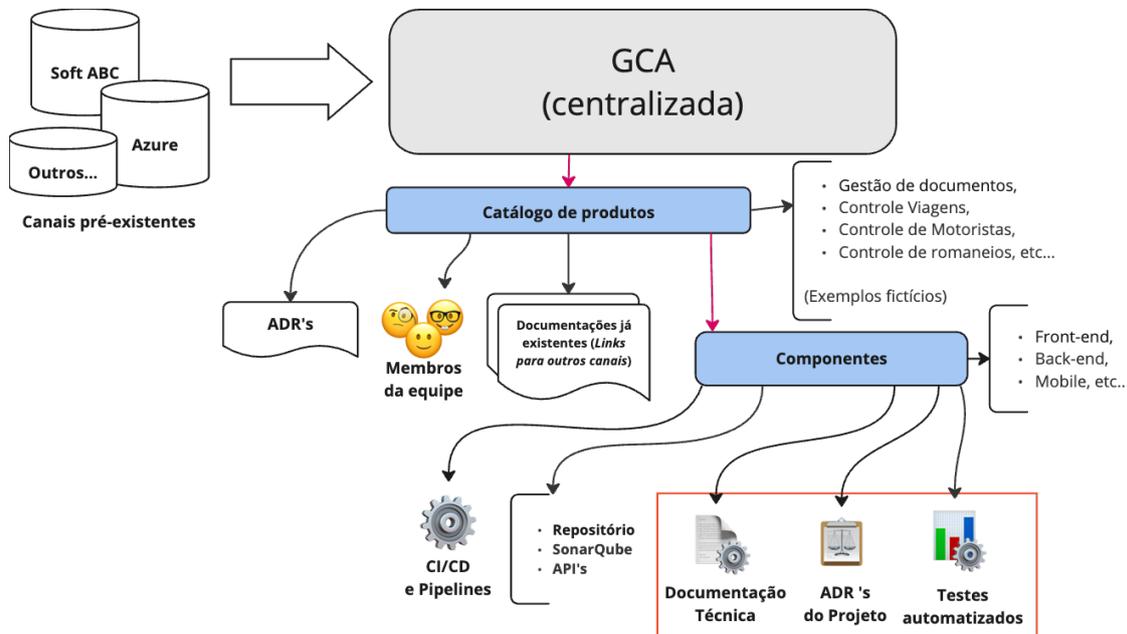
- ADR's locais nos componentes do produto;
- Documentação técnica de *front-end*;
- Documentação técnica do *back-end*;
- Disponibilização dos testes automatizados;
- Definição de uma meta para inserção dos demais produtos da suíte logística.

Esses pontos foram observados nos comentários realizados pelos participantes apontados pela **Seção 6.3**. O que se mostrou bastante evidente, principalmente, por duas situações vivenciadas: a integração de novos colaboradores e a mudança de equipe de um dos colaboradores participantes.

Processos de integração eficazes são críticos para a incorporação de novos membros em equipes de software. Portanto, fornecer documentação clara, mentoria e acesso às ferramentas necessárias pode reduzir significativamente o tempo necessário para que

novos membros se tornem contribuidores produtivos (BALDAUF; FROHLICH; SALOMON, 2019). A correção realizada no processo aplicado de GCA pode ser vista na **Figura 15**.

Figura 15 – Atualização do modelo de GCA aplicado.



Fonte: o autor.

A implementação de uma forma de disponibilizar RDAs (Registros de Decisões Arquitetônicas) gerados pela própria equipe, de maneira dinâmica e simples, para serem anexados de forma síncrona à ferramenta adotada, mostrou-se primordial. Sabendo que a ferramenta permite apontar para um servidor e realizar a leitura de arquivos do tipo Markdown®, conforme demonstrado na **Figura 16**, foi acrescentada, no código-fonte, uma pasta denominada "*architecture-decisions*".

Essa inserção poderá ser feita no código-fonte de qualquer componente. Foi alterado no Backstage® para que ele lesse essa pasta dentro dos componentes e realizasse a listagem dos arquivos Markdown® de forma clara, permitindo que qualquer pessoa com acesso à ferramenta pudesse visualizá-los.

Para reforçar o compromisso da pesquisa em trazer agilidade ao processo e o menor envolvimento possível dos participantes, a única tarefa necessária aos desenvolvedores é redigir, de forma direta, as alterações realizadas no projeto que se tornaram decisões arquitetônicas no produto.

Sobre a aplicação de documentação técnica, neste momento, pensou-se nos com-

Figura 16 – RDA local para ser compartilhada.

```

1 # ADR 1: Criação de um modelo para tabelas do Gr
2
3 ## Status
4 Aceito
5
6 ## Data
7 2023-06-24
8
9 ## Decisores
10 - [redacted], Arquiteto Front-end
11 - J [redacted], Desenvolvedor Sênior
12
13 ## Consultados
14 - C [redacted], Líder de Equipe
15
16 ## Informados

```

Fonte: o autor.

ponentes de *front-end* e *back-end*. Para o *front-end*, o time de arquitetura, juntamente com os participantes da pesquisa, optou pelo **TypeDoc**¹, por gerar documentação de forma prática conforme ocorre a codificação.

O TypeDoc[®] é uma ferramenta poderosa no quesito praticidade na geração de documentação de projetos que utilizam TypeScript, sendo particularmente útil no desenvolvimento front-end. Ele extrai automaticamente informações de tipos e comentários do código-fonte, produzindo documentação abrangente e sustentável que melhora a colaboração e o entendimento do código entre os membros da equipe (CONTRIBUTORS, 2023b).

Em relação à documentação do *back-end*, atualmente, já é realizado um controle para consulta de suas APIs. Foi proposta, então, a implementação da documentação para o sistema de mensageria utilizado entre os produtos da organização. Após a análise do time de arquitetura, uma ferramenta que se mostrou promissora para ser utilizada foi o SpringWolf^{®2}.

O SpringWolf[®] simplifica a documentação de APIs para aplicações Spring Boot, gerando automaticamente documentação AsyncAPI para Kafka, RabbitMQ e outros sistemas de mensagens (CONTRIBUTORS, 2023a). Dessa forma, atende ao processo de GCA proposto, pois fornece, de maneira clara e estruturada, documentação para arquiteturas orientadas a eventos, por meio de anotações no código-fonte, facilitando o entendimento e a manutenção de sistemas back-end complexos.

¹<https://typedoc.org/documents/Overview.html>

²<https://www.springwolf.dev/>

A adição dos testes automatizados no catálogo de produtos foi simples, pois, basicamente, foi acrescentado ao Backstage® o repositório de testes automatizados, sendo o caminho algo comum entre os repositórios, variando apenas o localizador do produto para determinar sua localização.

A migração das informações existentes de outros produtos mostrou-se, conforme comentado na **Seção 6.3**, um passo importante para tornar o processo de GCA (Gestão do Conhecimento Arquitetônico) algo cultural na organização. Conforme a proposta demonstrada na **Seção 5.3.3**, pensou-se em realizar a preparação dos demais produtos pelos próprios desenvolvedores no Backstage®, inclusive para incentivar a utilização do processo de GCA na suíte Logística.

7.2 Monitoramento

Seguindo o roteiro proposto na Figura 11, foi realizada a apresentação do processo de centralização da GCA, já com resultados práticos obtidos pelas duas equipes que participaram do primeiro ciclo. A apresentação foi realizada na "*Guilda de Tecnologia* já com as melhorias propostas na **Seção 7.1**.

Essa reunião que ocorre semanalmente com a finalidade de divulgar inovações que os times tem aplicado em seu produto, melhorias sugeridas pelo time de arquitetura e mudança em processos no desenvolvimento de software. Seguindo esse contexto foi sugerido a apresentação do processo de centralização da GCA, demonstrando os benefícios alcançados e as possibilidades de melhorias.

O gestor de tecnologia aprovou a apresentação e sugeriu que fosse falado dos benefícios já alcançados, sendo eles:

- Agilidade para encontrar documentação de técnica e de negócio;
- Melhoria na comunicação com o time de arquitetura;
- A possibilidade de se gastar mais tempo desenvolvendo rotinas diárias ao invés de buscar informações em diversos canais;
- Permitir o acompanhamento mais de perto tanto do time de arquitetura quanto líderes técnicos e de negócio;
- Melhoria na integração de novos colaboradores e na migração entre diferentes produtos.

Neste encontro temos uma média de 20 participantes, sendo todos eles líderes técnicos e arquitetos, geralmente um arquiteto de *front-end* e um arquiteto de *back-end*. A Reunião não pode ser gravada por determinação da organização, principalmente por essa ser uma reunião estratégica.

Dois pontos se mostraram bem relevantes para os participantes, sendo um deles a possibilidade de compartilhar decisões arquitetônicas exclusivas do produto, principalmente por facilitar o processo de adaptação de novos integrantes. Outro ponto que se mostrou importante, foi a facilitação para geração de documentação técnica no *front-end* e a possibilidade de documentar os eventos de comunicação entre produtos com o SpringWolf®.

Após a apresentação, foi realizado um convite para testar a utilização da ferramenta no dia a dia. Tivemos uma grande aceitação dos *Tech leads* e foi passado o acesso para 26 desenvolvedores participantes de 10 produtos diferentes em 7 equipes.

O monitoramento foi realizado em conversas informais com todos os participantes, dessa forma se evitou a interrupção no trabalho cotidiano que poderiam gerar respostas rasas a respeito da sua usabilidade devido a urgência na entrega das demandas diárias. Para avaliação pretende-se aplicar um questionário modelo TAM para validar a eficiência do processo aplicado.

7.3 Avaliação do efeito das ações

Neste segundo ciclo, essa etapa revelou-se fundamental para a consolidação da proposta. Conforme evidenciado no questionário baseado no modelo TAM, houve a participação de integrantes de dez equipes, que foram constantemente consultados pelo pesquisador quanto à utilização da ferramenta no contexto da Gestão do Conhecimento Arquitetônico (GCA) na organização.

Houve algumas equipes com apenas um desenvolvedor listado. Foi repassado para eles realizarem o repasse de conhecimento para os outros integrantes da equipe para que dessa forma fosse possível disseminar a aplicação do processo centralizador da GCA, sendo aos poucos inserido na cultura organizacional.

Durante essa fase, avaliou-se se os objetivos estabelecidos na etapa anterior foram alcançados de maneira eficiente.

Após a liberação de uso do Backstage®, para as equipes que se voluntariaram, foi aplicado um questionário quali-quantitativo, sendo que na parte quantitativa utilizou-se

o modelo TAM (*Modelo de Aceitação de Tecnologia*). Segundo Silva, Finger and Silva (2022), o modelo TAM é um questionário desenvolvido para capturar a percepção dos participantes sobre os principais motivos que levam os usuários a aceitar ou rejeitar uma determinada tecnologia.

Tabela 9 – Afirmativas nos indicadores do modelo TAM

| Utilidade Percebida | |
|-----------------------------|---|
| UPI | O uso do Backstage melhora minha experiência sobre gestão do conhecimento (acredito ter entendido mais sobre gestão do conhecimento em um tempo menor do que levaria sem usar o Backstage). |
| UP2 | O uso do Backstage me ajuda a identificar ações e práticas da Arquitetura através de ADR's (Architecture Decisions Record) para aumentar minha produtividade |
| UP3 | Considero o Backstage útil para identificar ações e práticas de gestão do conhecimento. |
| UP4 | As ações e práticas de Gestão de Conhecimento Arquitetônico (AKM) através do compartilhamento de ADR's (Architecture Decisions Records) para visualização de todos os times é promissor. |
| Facilidade de Uso Percebida | |
| FU1 | A apresentação do Backstage foi clara para melhorar minha eficácia na gestão do conhecimento existente nas outras equipes. |
| FU2 | Utilizar o Backstage para identificar aspectos da arquitetura dos produtos não requer muito esforço mental. |
| FU3 | O Backstage é fácil de usar (tanto para ter acesso ao ambiente do meu projeto, quanto para tirar dúvidas em relação a outros projetos). |
| FU4 | Considero fácil obter informações a respeito de outros times pelo Backstage (tais como os integrantes do time, repositório, API's, entre outros). |
| Intenção de uso futuro | |
| IU1 | Supondo que eu tenha acesso ao Backstage, eu tenho a intenção de utilizá-lo quando necessário |
| IU2 | Dado que tenha acesso ao Backstage, com todas funcionalidades ativas, eu prevejo que irei utilizá-lo quando necessário |
| IU3 | Eu prefiro usar o Backstage como ferramenta padrão do produto do que abrir cada artefato separadamente (Swagger, Sonar, ADR, Documentação, Git). |

O questionário quantitativo modelo TAM foi dividido em 3 partes: **utilidade percebida, facilidade de uso e intenção de uso.**

Tabela 10 – Avaliação Qualitativa a respeito do processo proposto

| Compartilhamento de conhecimento (KT - <i>Knowledge Transfer</i>) | |
|--|--|
| KT1 | Em se tratando do compartilhamento de conhecimento pelo time de arquitetura com os times de desenvolvimento, e entre os próprios times de desenvolvimento, o Backstage® se mostra promissor. Qual a sua opinião e quais sugestões você daria a respeito desse tema?. |
| KT2 | Em relação à geração de conhecimento entre os times, na sua opinião, o Backstage® facilita esse processo? O que poderia ser melhorado ou acrescentado para aprimorá-lo? |
| KT3 | O Backstage® pode facilitar o processo de onboarding de novos integrantes ao time, fornecendo links úteis e informações essenciais sobre o produto, como Sonar, repositório, APIs, entre outros? |
| KT4 | Ao considerar o Backstage® como um centralizador do conhecimento organizacional, você acredita que ele se mostra eficiente? Nesse sentido, quais informações você considera essenciais para a centralização do conhecimento organizacional, tanto no contexto da arquitetura quanto na interação entre as equipes? |
| KT5 | O que você mudaria dentro do Backstage e/ou na apresentação dos artefatos disponibilizados em cada produto? Cite os pontos positivos, negativos e limitações que você percebeu na utilização do backstage. Sugestões são sempre bem vindas. |

Além dos dados quantitativos do TAM, foram acrescentadas questões abertas para permitir que os participantes expressem suas opiniões sobre o processo aplicado e descrevam quais os aspectos positivos e os aspectos negativos durante o período de testes. As perguntas se encontram na **Tabela 10**.

7.3.1 Visão geral dos participantes

Conforme demonstrado na **Tabela 11**, o estudo contou com a participação de 26 profissionais, incluindo desenvolvedores, arquitetos e outros membros envolvidos no processo. Esses participantes integram nove equipes de produto, além de outras equipes de suporte, como o time de arquitetura, que teve dois representantes, T2 e T14, e o time de DevOps, representado pelo participante T3.

Tabela 11 – Participantes por equipe

| Papel | Pessoa | Projetos | Cargo* | empresa* | Profissional* |
|-----------------------|--------|----------|--------|----------|---------------|
| <i>Time A</i> | | | | | |
| <i>Júnior</i> | T8 | 1 | 1-5 | 1-5 | 1-5 |
| <i>Sênior</i> | T20 | 1 | <1 | <1 | 5-10 |
| | T23 | 1 | >10 | <1 | >10 |
| | T24 | 1 | 5-10 | <1 | >10 |
| | T25 | 1 | 5-10 | <1 | >10 |
| <i>Especialista</i> | T1 | 1 | 1-5 | 1-5 | >10 |
| | T5 | 2-3 | 5-10 | 1-5 | 5-10 |
| | T6 | 2-3 | 1-5 | 1-5 | >10 |
| | T7 | >3 | 1-5 | >10 | >10 |
| | T9 | >3 | <1 | >10 | >10 |
| | T10 | 1 | >10 | 1-5 | >10 |
| | T21 | 1 | <1 | <1 | >10 |
| <i>Time C</i> | | | | | |
| <i>Sênior</i> | T26 | 2-3 | 1-5 | 1-5 | >1-5 |
| <i>Time D</i> | | | | | |
| <i>Sênior</i> | T19 | 1 | <1 | <1 | >10 |
| <i>Time E</i> | | | | | |
| <i>Sênior</i> | T13 | 2-3 | 1-5 | 1-5 | 5-10 |
| <i>Time B</i> | | | | | |
| <i>Pleno</i> | T22 | 2-3 | 1-5 | 1-5 | 1-5 |
| <i>Sênior</i> | T12 | 2-3 | 1-5 | 1-5 | 5-10 |
| <i>Especialista</i> | T11 | 2-3 | >10 | 1-5 | >10 |
| | T18 | >3 | 1-5 | 1-5 | 5-10 |
| <i>Time F</i> | | | | | |
| <i>Especialista</i> | T17 | 2-3 | 1-5 | 1-5 | >10 |
| <i>Time G</i> | | | | | |
| <i>Especialista</i> | T16 | 2-3 | 1-5 | 1-5 | 5-10 |
| <i>Time H</i> | | | | | |
| <i>Especialista</i> | T | 2-3 | <1 | 1-5 | >10 |
| <i>Time I</i> | | | | | |
| <i>Especialista</i> | T4 | 2-3 | 1-5 | >10 | >10 |
| <i>Outros Setores</i> | | | | | |
| <i>Arquiteto</i> | T2 | 1 | 1-5 | 1-5 | 5-10 |
| | T14 | >3 | 5-10 | 1-5 | >10 |
| <i>SRE</i> | T3 | 2 | 1-5 | 1-5 | >10 |

* Medida em anos

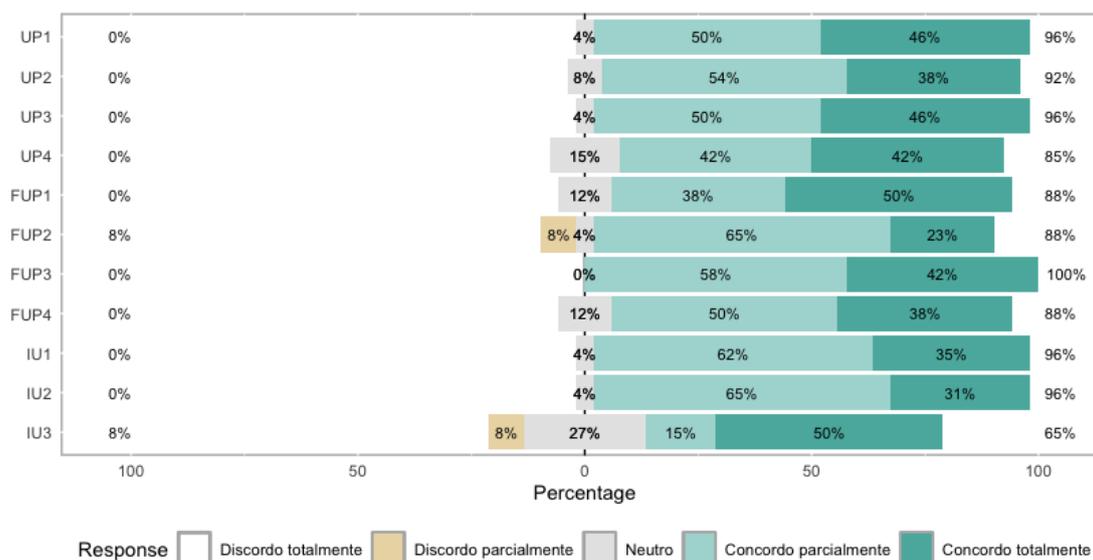
Diferentemente do primeiro ciclo, não houve a inserção de novos integrantes no projeto. No entanto, o time de arquitetura formalizou seu posicionamento em relação à proposta, consolidando sua visão sobre o impacto e a viabilidade da iniciativa. Durante o período de avaliação, o pesquisador recebeu diversas chamadas informais, sobretudo para esclarecimento de dúvidas relacionadas ao projeto e à implementação de determinados aspectos apontados no processo.

Outro ponto relevante diz respeito à representatividade das equipes na avaliação. Das sete equipes de produto envolvidas, apenas uma contou com um único representante na fase de monitoramento. No caso do Time B, que participou da primeira etapa, houve uma redução de quatro participantes em relação ao ciclo anterior. Em contrapartida, no Time A, a participação foi integral, incluindo os quatro novos integrantes mencionados na **Tabela 8**, o que possibilitou uma avaliação mais abrangente sobre a adoção do processo.

7.4 Resultados quantitativos obtidos do questionário TAM

Neste momento, a percepção dos participantes foi baseada nas afirmativas do modelo TAM, conforme referenciado na **Tabela 9**. A **Figura 17** apresenta uma visão geral dos resultados em relação à percepção dos participantes sobre o processo de centralização da GCA adotado.

Figura 17 – Percepção dos participantes sobre a GCA



Fonte: o autor.

No que se refere à percepção dos pesquisados sobre a **Utilidade Percebida** do Backstage® como um centralizador da GCA, observa-se, de modo geral, uma boa aceitação. A maioria das respostas indica um nível de concordância (parcial ou total) superior a 85% em relação aos itens desse indicador. Ressalta-se que não foram registradas percepções de discordância em nenhum dos itens avaliados. No entanto, observa-se que alguns participantes preferiram manter-se neutros, representando 4% e 15% das respostas, de-

pendendo da afirmação analisada.

Na afirmação **UP4**, nota-se uma alta concordância entre os participantes. Entretanto, quatro participantes (15%) permaneceram neutros, o que pode indicar a necessidade de aprimorar ações mais eficazes para o compartilhamento das RDAs e seu significado no processo de desenvolvimento de software.

Quanto à **Facilidade de Uso Percebida**, percebe-se que a maioria dos participantes (>88%) concorda (total ou parcialmente) com esse indicador. Contudo, no item **FUP2**, dois participantes (8%) expressaram discordância parcial. Esse dado evidencia a necessidade de compreender os motivos dessa percepção e de esclarecer eventuais dúvidas dos pesquisados sobre o tema.

Em relação à afirmação **FU3**, que avalia a facilidade de uso do Backstage®, obteve-se 100% de concordância (parcial ou total), o que impede questionamentos sobre essa métrica. Nos itens **FUP1** e **FUP4**, no máximo três participantes (12%) optaram por uma posição neutra.

Sobre a **Intenção de Uso Futuro** do Backstage® como uma ferramenta consolidada para a *Suíte Logística*, os feedbacks foram bastante positivos. Para as afirmações **IU1** (“Pretendo utilizar o Backstage® no futuro”) e **IU2** (“Caso tenha acesso ao Backstage® com todas as suas funcionalidades ativas, utilizá-lo-ei quando necessário”), 96% dos pesquisados concordaram, enquanto apenas um participante optou pela posição neutra.

No que tange à afirmação **IU3** (“Utilizar o Backstage® como centralizador de conhecimento em vez de utilizar cada ferramenta separadamente”), 65% dos entrevistados declararam concordância total ou parcial. No entanto, dois participantes demonstraram discordância parcial quanto à possibilidade de priorizar o uso da ferramenta no dia a dia. Além disso, sete participantes (27%) mantiveram-se neutros, sugerindo que pode haver lacunas na experiência de uso que motivaram essa resposta. Esse aspecto reforça a necessidade de um levantamento mais detalhado para identificar possíveis falhas nos testes ou nas estratégias de implementação.

Os resultados quantitativos gerados pelo modelo TAM foram bastante satisfatórios, indicando um alto índice de aprovação. Para complementar a etapa de avaliação, foi realizada uma análise qualitativa, na qual foram coletadas informações adicionais sobre a visão dos participantes no processo.

7.5 Percepção dos desenvolvedores sobre a centralização da GCA

Após a análise quantitativa dos resultados pelo modelo TAM, realizou-se, nesta etapa, a transcrição das respostas dos participantes, seguindo os procedimentos de codificação. Para a organização dos dados, foi adotada a **Codificação Axial**, na qual os códigos são agrupados em categorias e subcategorias, estabelecendo relações entre eles.

Conforme afirmam Cassiani, Caliri and Pelá (1996):

"Todo o processo analítico nesta etapa, tem por objetivos: construir a teoria, trazer ao processo científico o rigor metodológico necessário, auxiliar o pesquisador a desenvolver o fundamento, a densidade, a sensibilidade e a integração necessária para gerar uma teoria"

O objetivo da análise qualitativa nesta fase foi sintetizar os dados, a fim de identificar as dificuldades e os benefícios percebidos pelos participantes após o uso da ferramenta adotada para a centralização da GCA na organização. A análise foi conduzida por um pesquisador e discutida com os demais pesquisadores em reuniões sucessivas.

Foram elaboradas cinco questões abertas, as quais estão descritas na **Tabela 10**. Todas as respostas referentes a esta análise qualitativa estão presentes no **Apêndice B**. As respostas foram organizadas em três categorias principais: (i) Transferência de Conhecimento (*Knowledge Transfer – KT*); (ii) Eficiência na geração de conhecimento pelo processo sugerido; e (iii) Opinião sobre a ferramenta utilizada e sua eficiência. A seguir, apresentamos uma análise detalhada de cada uma dessas categorias.

7.5.1 Transferência de conhecimento (KT - *Knowledge Transfer*)

As questões **KT1** e **KT3** investigam a percepção dos participantes sobre o compartilhamento de conhecimento entre os times de desenvolvimento e a equipe de arquitetura, bem como o suporte à integração de novos colaboradores.

A análise das respostas permitiu a identificação de quatro sub-categorias principais: (i) Necessário melhorar o engajamento do time com o processo; (ii) É um motivador para compartilhamento de conhecimento; (iii) Facilita o processo de integração de novos membros; e (iv) realiza a centralização do conhecimento.

Sobre a necessidade de melhorar o engajamento do time com o processo, mesmo a ferramenta sendo promissora, há a necessidade de reservar um momento para treina-

mento mais aprofundados com as equipes. Segundo **P1**:

“Excelente ferramenta, porém a empresa e o time precisa comprar a ideia da documentação”

Esse ponto é essencial para que o processo se torne parte da cultura organizacional. Conforme apontado na pesquisa-ação, os líderes precisam estar envolvidos na mudança para garantir sua adoção efetiva.

Além disso, o comprometimento da equipe também se mostra um fator crucial. **P8** destaca:

“Sim, acho que só precisamos trabalhar mais em cima disso para criar o hábito de compartilhar.”

Essa necessidade de tornar a GCA parte da cultura organizacional, e não apenas um processo formal sem adesão genuína, é reforçada por **P2**:

“Acredito que em termos de ferramentas estamos bem servidos, o que falta é o engajamento justamente para que possamos ir melhorando ela (...).”

É perceptível que os desenvolvedores valorizam uma ferramenta que realmente se torne uma referência para a **GCA** dentro da organização. Nesse sentido, **P21** sugere:

“pode [facilitar], através de uma jornada de cursos e materiais definidos segundo a posição do colaborador.”

O processo adotado também se mostrou um incentivador para o compartilhamento de conhecimento. **P7** observa que:

“O backstage ajuda a melhorar a comunicação entre o time de arquitetura e os [times] de desenvolvimento por ter a documentação unificada(...).”

Nessa mesma linha, **P12** reforça a importância do time de arquitetura como referência no crescimento profissional:

“(...) usá-los [time de arquitetura] como referência para o crescimento é algo surreal, essa foi a minha experiência quando precisei solicitar apoio desse time.”

A facilidade no processo de integração de novos membros foi amplamente reconhecida pelos entrevistados. **P14** destaca:

“Com certeza. O Backstage® deveria ser a home de todo desenvolvedor. Aha pinada no browser!”

Além disso, **P22** menciona:

"Queria ter visto por primeiro o backstage durante o meu processo de onboarding"

A respeito da centralização do conhecimento, os depoimentos também validam a importância do processo aplicado para a centralização da **GCA**. **P13** ressalta:

“Sim, as informações centralizadas evitam que os novos membros precisem se preocupar em lembrar onde buscar conhecimento.”

De forma semelhante, **P16** compartilha sua experiência ao ingressar na **ABC Company**³:

"Sim, quando entrei na ABC Company, tive que descobrir muitas coisas por conta própria, pois as informações não estavam centralizadas em um único local. Ter acesso centralizado a conhecimentos e informações será incrível, não apenas para o onboarding, mas também para o dia a dia dos desenvolvedores."

7.5.2 Eficiência para geração de conhecimento pelo processo sugerido

As questões **KT2** e **KT4** investigam a geração de conhecimento pelos times de produto por meio do processo de centralização da **GCA** sugerido. A análise das respostas permitiu a identificação das seguintes sub-categorias: (i) Facilidade na geração do conhecimento arquitetônico pelos times; (ii) Não traz valor para geração de conhecimento; e (iii) dúvidas em relação ao que é proposto.

Sobre a facilidade na geração do conhecimento arquitetônico pelos times, a satisfação com o modelo proposto como gerador de conhecimento entre os times é evidente. Isso pode ser observado na resposta de **P8**:

“A documentação de todos os produtos em um único lugar vai facilitar muito o entendimento dos processos de cada produto, fora que vai ser uma mão na roda, quando precisamos integrar os produtos.”

³Nome fictício

A disponibilidade de agentes para o monitoramento dos produtos liberados de forma síncrona também auxilia significativamente os desenvolvedores na absorção do conhecimento. Alguns participantes destacaram pontos relevantes nesse contexto, como a facilidade de acesso a APIs via **Swagger®**, conforme mencionado por P1:

"Sim, as APIs (swagger)"

Além disso, a disponibilização de documentações importantes foi ressaltada por **P2** afirma que;

"Mapa de relacionamento ("Diagrama de classe") pra mim é um dos pontos mais legal pois mostra de maneira visual e simplificada as ligações entre os componentes."

P7 também destacou aspectos fundamentais relacionados ao conhecimento arquitetônico:

" Sim. É interessante ter disponível os padrões de codificação e exemplos de uso, repositórios de código, anotações/atas de reunião e histórico de versões das ferramentas de software."

Isso mostra que quanto mais ferramentas que apoiam as decisões arquiteturais serem engajadas ao processo, maior visibilidade de eficiência trará aos desenvolvedores.

Algumas respostas apontaram que o processo aplicado não gera valor significativo para a criação de conhecimento. Apesar desse contraponto, tais observações contribuíram para identificar pontos de melhoria na pesquisa. **P13** mencionou:

"Infelizmente o Backstage não possui uma solução integrada para criação/edição de documentos. Embora a proposta do Backstage seja utilizar mais o DocAsCode (...)"

Dessa análise, emergem dois pontos principais:

- A geração de conhecimento, também, ocorre a partir da pesquisa e consulta dos dados de diferentes produtos de software, como o proposto;
- O **Backstage®**, sendo uma ferramenta **DocAsCode**, não tem como objetivo principal a criação de documentos, mas sim a centralização do conhecimento conforme a metodologia já adotada pela organização.

Sobre as dúvidas sobre a proposta apresentada, alguns participantes demonstraram incerteza quanto à adequação da ferramenta para a proposta. **P16** mencionou:

"Não sei se seria a melhor ferramenta, mas pode auxiliar."

Embora exista interesse na proposta, observa-se que a necessidade de uma solução eficaz para centralização da GCA ainda é um ponto de atenção. Essa percepção pode ser interpretada de forma positiva, pois reforça a importância de aprimorar a comunicação da proposta e buscar estratégias para maior adesão. **P11** também destacou a necessidade de evoluir a implementação para que a ferramenta se torne mais eficiente:

"Em como está hoje não vejo que ele é completamente eficiente, porém precisaria analisar implementações futuras, mas claramente é uma ferramenta que pode se tornar eficiente sim."

Dessa forma, torna-se essencial promover a iniciativa dentro da organização e reforçar a comunicação dos pesquisadores com os participantes, visando esclarecer dúvidas e aumentar a aceitação do modelo proposto.

7.5.3 Avaliação da Ferramenta: Eficiência e Percepções dos Usuários

Na questão KT5, foi solicitado aos participantes que apontassem os pontos positivos e negativos da ferramenta de apoio ao processo de GCA, assim como possíveis limitações. As respostas consideradas válidas foram categorizadas conforme sua importância nas seguintes áreas: (i) Centralizador de informações, (ii) Controle de acessos, (iii) melhoria no monitoramento do produto e (iv) aplicações gerais.

Um dos principais pontos positivos destacados no processo de GCA foi como "Centralizador do conhecimento arquitetônico", em um único local, facilitando o acesso às informações. Segundo **P20**:

"(...) O intuito é ajudar em momentos de estudos de problemas que podem acontecer, facilitando encontrar o vínculo entre pessoas, times e APIs."

Já **P23** ressaltou os benefícios dessa centralização para novos integrantes e para a eficiência na consulta de informações:

"Entendo que centralizar a informação sobre os projetos simplifica a adaptação de quem está iniciando, evita a redundância de estratégias de acesso a documentação do produto, e simplifica consultas sempre que forem necessárias."

Outro ponto relevante foi a questão do Controle de acesso. Os participantes indicaram que o processo para inserção de novos usuários ainda é burocrático. No entanto, ao realizar uma análise dentro da organização, foi identificado que o controle de acesso dos desenvolvedores já é centralizado em uma base de dados específica. Essa base contém todas as informações necessárias para integração automática ao Backstage®, o que pode ser explorado para otimizar o processo.

Se tratando da melhoria no monitoramento do produto, **P26** sugeriu melhorias na ferramenta para tornar essa funcionalidade mais robusta:

“Criação de uma sessão para monitoramento da aplicação, com mais detalhes, alertas, dashboards etc.”

A categoria aplicações gerais aborda funcionalidades já presentes na ferramenta, mas que alguns participantes sugeriram aprimorar. **P2** destacou:

“Automatizar a atualização de conteúdos como swagger, async api, etc”

Entretanto, essas funcionalidades já estão disponíveis na ferramenta, indicando que pode haver uma lacuna no conhecimento dos usuários sobre os recursos existentes. Esse aspecto reforça a necessidade de treinamentos mais aprofundados para que os desenvolvedores utilizem o **Backstage®** de forma eficiente.

Algumas respostas não permitiram uma análise mais aprofundada, pois foram muito genéricas, como simples afirmações de aprovação, preferências sobre layout ou sugestões fora do escopo da pesquisa sobre **GCA**.

Ainda assim, algumas respostas demonstraram uma visão positiva sobre a ferramenta, como observado no comentário de **P6**:

“Não mudaria nada, só tentaria alimentar cada vez mais a ferramenta.”

Esse tipo de afirmação reforça a motivação para o uso contínuo e aprimoramento do **Backstage®** dentro da organização.

7.5.4 Reflexão e avaliação final

Nesta etapa da pesquisa, teve-se por finalidade compreender se o principal objetivo do modelo de GCA (Gestão do Conhecimento Arquitetônico) proposto havia sido alcançado após a aplicação das melhorias destacadas na **Seção 7.1**. A avaliação desta

etapa foi dividida em uma parte quantitativa, baseada no modelo TAM (*Technology Acceptance Model*), e uma parte qualitativa, para compreender de forma mais clara a visão dos envolvidos na avaliação e tentar identificar a eficiência (ou não) do modelo.

Analisando os tópicos do questionário TAM – **Utilidade Percebida (UP)**, **Facilidade de Uso (FU)** e **Intenção de Uso (IU)** –, podemos afirmar que há uma grande aceitação em integrar o modelo de GCA aplicado com o Backstage® na suíte logística como um centralizador do CA (Conhecimento Arquitetônico) organizacional.

Outro ponto interessante a se observar, entre as 11 questões do modelo TAM, é que, em 10 delas, tivemos pelo menos uma pessoa se posicionando como neutra. Isso demonstra que existe a necessidade de melhorar o esclarecimento dos pontos positivos de se utilizar a GCA de forma prática, a fim de trazer maior eficiência ao trabalho.

Por meio da análise do cenário qualitativo, obtivemos as seguintes impressões advindas dos pesquisados:

- **Transferência de conhecimento (KT - Knowledge Transfer):** O modelo proposto para a GCA (Gestão do Conhecimento Arquitetônico) foi bem recebido e incentivou a troca de informações entre os times. Porém se mostrou necessário melhorar o engajamento dos times com a realização de treinamentos para reforçar a importância da documentação e contar com o apoio da liderança para que esse hábito se torne natural dentro da equipe. Além disso, foi sugerida a criação de uma trilha de aprendizado dentro do modelo, o que poderia ser uma espécie de guia estruturado, contendo materiais de referência, tutoriais e boas práticas para que os novos integrantes aprendam de forma eficiente e organizada. No geral, o texto evidencia que, apesar dos benefícios da centralização do conhecimento e do compartilhamento estruturado de informações, é preciso um esforço maior para incentivar o uso contínuo da ferramenta e torná-la parte da rotina dos desenvolvedores.
- **Eficiência para geração de conhecimento pelo processo sugerido:** O principal objetivo da ferramenta dentro do processo proposto, é facilitar o acesso às informações que já foram documentadas pelos times durante o desenvolvimento. Isso significa que a eficiência na geração de conhecimento depende diretamente da forma como os times utilizam a ferramenta e como mantêm a documentação atualizada. 65% dos participantes entenderam essa funcionalidade e reconheceram sua utilidade para organizar e disseminar conhecimento. No entanto, alguns pesquisadores apontaram que não perceberam esse valor ou ainda têm dúvidas sobre como o Backstage® pode ajudar na geração de conhecimento. Isso indica que talvez falte

treinamento ou uma comunicação mais clara sobre como utilizar a ferramenta de forma eficiente. Ele não substitui a criação de conhecimento, mas potencializa sua disseminação e acessibilidade dentro da organização.

- **Avaliação da Ferramenta: Eficiência e Percepções dos Usuários:** Os participantes elogiaram a ferramenta por centralizar o conhecimento arquitetônico, facilitando o acesso à informação e a adaptação de novos membros. Também destacaram sua capacidade de simplificar consultas e reduzir redundâncias na documentação. Entretanto, apontaram desafios, como a burocracia no controle de acessos, embora uma base de dados centralizada possa otimizar esse processo.

Outro ponto destacado foi a necessidade de melhorias no monitoramento do produto, com sugestões de mais alertas e dashboards. Funcionalidades existentes, como a atualização automatizada de conteúdos (Swagger, AsyncAPI, etc.), foram citadas como áreas a melhorar, indicando a necessidade de treinamento para melhor utilização dos recursos. Apesar das críticas, a ferramenta foi reconhecida como valiosa, com sugestões para maior adoção e aprimoramento na organização.

Nesta etapa, avaliou-se se o objetivo do modelo de **GCA** proposto foi alcançado após as melhorias da **Seção 7.1**. A análise quantitativa (baseada no TAM) e qualitativa mostrou alta aceitação do Backstage® como centralizador do Conhecimento Arquitetônico (CA) na suíte logística. No entanto, respostas neutras em 10 das 11 questões do TAM indicam a necessidade de melhorar a comunicação sobre os benefícios da GCA para aumentar sua eficiência.

A análise qualitativa destacou pontos positivos, como a facilitação da transferência de conhecimento (KT), mas também desafios, como a necessidade de maior engajamento, treinamentos e suporte da liderança. Sugestões incluíram a criação de uma trilha de aprendizado e melhorias nas funcionalidades da ferramenta. Conclui-se que, embora o modelo tenha potencial, sua eficácia plena depende de esforços contínuos para integrá-lo à cultura organizacional.

7.6 Validação dos Resultados obtidos na pesquisa-ação

Esta seção apresenta a conclusão da pesquisa-ação, consolidando os resultados obtidos nos dois ciclos realizados e avaliando o alcance do objetivo principal: *"Identificar e analisar a aplicação da Gestão do Conhecimento Arquitetural (GCA) dentro da*

organização, identificar seu principal problema e propor uma solução para torná-la mais prática, colaborativa e ágil".

Durante as primeiras etapas, a investigação evidenciou que o maior desafio estava na centralização do Conhecimento Arquitetônico (CA) já existente, tornando seu acesso mais fácil e estruturado para a organização. Para abordar essa questão, foi criado um processo centralizado para a Gestão do Conhecimento Arquitetônico (GCA) (vide **Seção 5.3.1**). Como ferramenta para atender aos requisitos de centralização, optou-se pelo uso do Backstage®.

Foram realizadas análises quantitativas e qualitativas, nas quais se discutem os impactos da implementação, os desafios enfrentados e as lições aprendidas. Além disso, são apresentadas recomendações para consolidar o modelo e garantir sua adoção contínua (Vide **Seção 7.3** e **Seção 7.5.4**).

Os resultados das avaliações realizadas ao longo das duas etapas indicam que o principal problema relacionado à GCA foi resolvido inicialmente. No entanto, foram identificados ajustes a serem implementados, embora esses não comprometam o objetivo principal da pesquisa, mas sim representem oportunidades de aprimoramento para tornar o processo ainda mais simples e ágil. Como afirmado por Jr. and Werner (1997), a utilização de padrões em processos de software pode apoiar a evolução do produto, permitindo a reutilização do conhecimento e a adaptação contínua dos processos.

8 DISCUSSÃO DOS RESULTADOS

Neste capítulo, são abordadas as considerações finais relacionadas à proposta apresentada nesta dissertação. A **Seção 8.1** discorre sobre as conclusões levantadas pelos pesquisadores, com o apoio da literatura e das observações realizadas na indústria de software. A **Seção 8.2** apresenta as limitações identificadas durante a pesquisa. A **Seção 8.3** destaca as contribuições da pesquisa do ponto de vista acadêmico e da indústria de software, bem como a contribuição do método pesquisa-ação em todo o processo. Por fim, a **Seção 8.4** traz os trabalhos publicados pelos pesquisadores sobre o tema e as perspectivas futuras para o desenvolvimento e aplicação do conhecimento gerado.

8.1 Conclusões

A transferência de conhecimento é um processo crítico para o sucesso organizacional, pois permite que informações valiosas sejam compartilhadas e aplicadas de forma eficiente (SZULANSKI, 2000). Levando em consideração essa premissa, foi realizado um estudo sobre os problemas enfrentados pela organização em relação ao Conhecimento Arquitetônico (CA) e como ele é gerenciado.

Foi possível verificar a existência desse conhecimento e um esforço inicial na formação da Gestão do Conhecimento Arquitetônico (GCA), principalmente considerando que a organização está no mercado há mais de 30 anos. Conforme Alavi and Leidner (2001), o conhecimento organizacional frequentemente está disperso e desorganizado, armazenado em repositórios ou formatos não estruturados, dificultando sua localização e compartilhamento. Esse foi exatamente o cenário encontrado pelos pesquisadores.

Tendo em mãos o real problema em relação à aplicação da GCA na organização, a proposta feita foi: *"Gerar um processo para a centralização da GCA, tornando seu acesso e a obtenção de novas informações algo dinâmico e ágil"*.

Em posse do conhecimento do real problema envolvendo a GCA na organização (vide **Seção 5.4.2**), foi possível focar na criação de um processo que pudesse centralizar o conhecimento existente e facilitar a inserção e o compartilhamento de novos conhecimentos. Ferramentas de software que suportam a GCA, como repositórios centralizados, facilitam o acesso e a disseminação do conhecimento arquitetônico, promovendo a colaboração entre equipes e reduzindo redundâncias (CAPILLA et al., 2016b).

Com o processo definido, foi necessário utilizar um software para aplicar o mo-

delo proposto. O Backstage® se mostrou promissor, pois a organização já o utilizava, porém de forma muito limitada em relação ao que a ferramenta realmente oferece como solução. Ferramentas que centralizam o conhecimento, como o Backstage®, facilitam o acesso à informação e promovem a colaboração entre equipes, reduzindo redundâncias e melhorando a eficiência organizacional (ENGINEERING, 2020).

Após a preparação do Backstage® para atender ao modelo proposto, ele foi implementado inicialmente em duas equipes responsáveis por produtos da *Suíte Logística*. Neste momento, devido à organização estar em um período de finalização de projeto e entrega, a avaliação realizada foi de modo informal. Essa abordagem foi adotada nesta pesquisa para obter insights profundos sobre as práticas e percepções dos participantes nesta primeira etapa.

Conforme Stringer (2014):

"As entrevistas informais são uma ferramenta valiosa na pesquisa-ação, pois permitem a coleta de dados de forma flexível e contextualizada, adaptando-se ao ambiente natural em que o estudo é conduzido."

Com base no levantamento dos resultados obtidos nesta etapa, foram levantados os seguintes pontos a serem incluídos no modelo proposto:

- **RDA's** locais nos componentes do produto;
- Documentação técnica de *front-end*;
- Documentação técnica do *back-end*;
- Disponibilização dos testes automatizados;
- Definição de uma meta para inserção dos demais produtos da suíte logística.

A integração do conhecimento arquitetônico em ferramentas de software permite que decisões arquiteturais sejam documentadas, compartilhadas e reutilizadas, melhorando a eficiência e a qualidade do desenvolvimento de software (JANSEN; BOSCH; AVGERIOU, 2009a). Todos os pontos levantados vieram dos desenvolvedores que testaram o processo de **GCA** e dos arquitetos que identificaram esses pontos como relevantes. A ferramenta também auxiliou, permitindo que a inserção dessas informações fosse simples e dinâmica.

Vale ressaltar os pontos a respeito da documentação técnica, que foi uma colocação do time de arquitetura para validar se as propostas feitas nos Registros de Decisões

Arquitetônicas (**RDA**) estão sendo utilizadas pelos times de produto. Os frameworks escolhidos para o processo interrompem minimamente o fluxo diário, exigindo uma quantidade de tempo irrisória para o desenvolvimento da documentação técnica.

Após os ajustes necessários terem sido realizados, a proposta foi apresentada em uma reunião envolvendo todos os líderes técnicos das equipes, denominada internamente como **Gilda de Tecnologia**. Foi demonstrada a proposta da pesquisa, juntamente com os resultados alcançados no primeiro ciclo. Os times foram convidados a utilizar o processo de **GCA** proposto.

Nesta fase, participaram 26 profissionais integrantes de 10 equipes diferentes. Conforme Wang and Noe (2007), o sucesso no compartilhamento de conhecimento está diretamente relacionado ao envolvimento ativo dos colaboradores. Quando se consegue o engajamento voluntário dos colaboradores, eles tendem a contribuir mais e a adotar práticas que facilitam a disseminação eficaz do conhecimento organizacional. Esse engajamento foi fundamental para os resultados obtidos nesta pesquisa.

Para realizar a avaliação nesta etapa, utilizaram-se dois questionários: um quantitativo, baseado no modelo TAM, e cinco perguntas qualitativas para obter a opinião clara dos participantes (vide **Seção 7.4** e **Seção 7.5**).

Os resultados em relação à execução do processo de GCA de forma centralizada, no método **TAM**, obtiveram mais de 65% de aprovação, sendo que, na maioria dos casos, a aprovação superou 80%.

Na pesquisa qualitativa, as respostas foram categorizadas como:

- Transferência de conhecimento (KT - Knowledge Transfer);
- Eficiência para geração de conhecimento pelo processo sugerido;
- Avaliação da Ferramenta: Eficiência e Percepções dos Usuários.

A implementação do processo de **GCA** centralizado utilizando o Backstage® como ferramenta, demonstrou resultados significativos, com mais de 65% de aprovação, pelos resultados do modelo TAM, alcançando mais de 80% em diversos quesitos. A ferramenta facilitou o acesso ao conhecimento arquitetônico, promoveu a colaboração entre equipes e reduziu redundâncias, conforme destacado pelos participantes.

A documentação técnica, tanto de *front-end* quanto de *back-end*, foi aprimorada com o uso de frameworks como **TypeDoc®** e **SpringWolf®**, que demandam tempo mínimo dos desenvolvedores, não afetando suas atividades diárias. Além disso, a inserção

de Registros de Decisões Arquitetônicas (**RDAs**) de forma dinâmica e a disponibilização de testes automatizados no catálogo de produtos foram pontos amplamente elogiados, evidenciando a eficácia do modelo proposto.

O engajamento voluntário de 26 profissionais de 10 equipes diferentes de forma voluntária, reforçou a importância do envolvimento ativo dos colaboradores para o sucesso do compartilhamento de conhecimento, conforme apontado por Wang and Noe (2007).

Apesar dos avanços observados, alguns desafios foram identificados. A necessidade de maior engajamento e treinamento dos colaboradores foi um ponto recorrente, especialmente para garantir que todos compreendam os benefícios da **GCA** e saibam utilizar a ferramenta de forma eficiente.

Além disso, a burocracia no controle de acessos e a falta de alertas e dashboards mais robustos foram citados como áreas a serem aprimoradas. A criação de uma trilha de aprendizado estruturada, com tutoriais e boas práticas, foi sugerida para facilitar a integração de novos colaboradores e consolidar o uso da ferramenta.

Por fim, a expansão do processo para todos os produtos da suíte logística e a melhoria na integração entre times de diferentes áreas são metas que demandarão esforços contínuos, visando consolidar a **GCA** como parte da cultura organizacional.

8.2 Limitações do trabalho

Durante a condução de estudos experimentais, podem existir limitações que afetem a validade dos resultados. A seguir, apresentamos algumas das limitações enfrentadas durante esta pesquisa:

- Na **Pesquisa-Ação**, o pesquisador está imerso no ambiente de estudo e mantém interação constante com os participantes. Esse envolvimento pode ter criado espaço para que algumas incertezas não fossem percebidas pelo pesquisador. Para mitigar essa limitação, foi necessário consultar continuamente as sugestões da literatura e conciliá-las com o contexto organizacional para validar a relevância de cada assunto.
- No **primeiro ciclo**, a apresentação e o treinamento foram realizados para duas equipes, compostas por um total de oito pessoas. Esse número pode ser considerado um limitador, ao levarmos em consideração eventuais atrasos de alguns participantes na

apresentação ou a possibilidade de que alguns não estivessem totalmente focados durante a atividade;

- No **segundo ciclo**, a apresentação foi realizada somente para líderes técnicos das equipes, não sendo estendida diretamente aos desenvolvedores de cada uma das dez equipes que optaram por participar da pesquisa. Para mitigar esses limitadores, os pesquisadores se mantiveram disponíveis para esclarecer dúvidas pontuais sobre o processo e o experimento conduzido;
- A **população selecionada** para o primeiro ciclo foi composta por participantes escolhidos por conveniência, o que poderia gerar um viés nos resultados analisados. Já no segundo ciclo, os participantes se voluntariaram após uma apresentação realizada na *Guilda de Tecnologia* para os *Techleads*, reduzindo parcialmente esse viés;
- Em relação à **análise de dados**, foi aplicado o mesmo procedimento (entrevistas e questionários) para todos os voluntários em cada ciclo. O principal limitador nesse aspecto foi o tamanho da amostra em relação ao total da suíte logística, que não representa a proporção ideal. Dessa forma, os resultados obtidos devem ser interpretados como indícios e não como conclusões definitivas.

8.3 Contribuições da Pesquisa

8.3.1 Do ponto de vista acadêmico

Nesta pesquisa, foi proposto e validado um modelo prático para a **Gestão do Conhecimento Arquitetônico (GCA)**, integrado a uma ferramenta de software (Backstage®). Essa validação contribui para a literatura ao demonstrar, de forma eficiente, a centralização do **Conhecimento Arquitetônico (CA)** em ambientes de desenvolvimento de software, promovendo a colaboração e a consistência entre as equipes. Além disso, foi possível evidenciar como tecnologias já existentes podem ser aprimoradas para atender às necessidades específicas da **Gestão do Conhecimento**.

Outro ponto relevante é a **evidência empírica** dos benefícios da GCA, como a melhoria na eficiência do desenvolvimento e a facilitação da transferência de conhecimento. Esses resultados reforçam a importância da GCA em ambientes de desenvolvimento de

software, consolidando sua relevância na literatura. O estudo também identifica desafios comuns na implementação da GCA, como a necessidade de engajamento dos colaboradores, além da importância de **treinamentos e suporte da liderança**. Essas descobertas oferecem insights valiosos para pesquisas futuras, destacando áreas que necessitam de maior atenção e desenvolvimento.

Ao abordar um problema prático da indústria de software e propor soluções baseadas em evidências, este trabalho contribui para a cultura de pesquisa em Engenharia de Software, demonstrando como academia e indústria podem colaborar para solucionar desafios reais e gerar conhecimento científico.

Em resumo, as contribuições acadêmicas deste estudo incluem a **validação de um modelo de GCA**, a **integração de ferramentas e processos**, a **apresentação de evidências empíricas**, a **identificação de desafios e oportunidades**, a **proposição de direções futuras** e a **publicação de uma RSL**. Essas contribuições enriquecem a literatura e abrem caminho para novos estudos na área de **Gestão do Conhecimento Arquitetônico e Engenharia de Software**.

8.3.2 Do ponto de vista da indústria de software

Como contribuição para a melhoria organizacional, este estudo propõe e valida um processo prático para a **centralização do conhecimento arquitetônico**, utilizando o **Backstage®**. Essa solução permite que as equipes acessem informações de maneira ágil e organizada, reduzindo redundâncias e aumentando a eficiência no desenvolvimento de software.

Ao consolidar o conhecimento arquitetônico em uma única plataforma, é facilitado o acesso a **informações críticas**, como **decisões arquitetônicas (RDAs)**, **documentação técnica** e **testes automatizados**. Isso é consideravelmente útil para **novos colaboradores** e **equipes** que necessitam de uma visão clara da arquitetura dos produtos. Além disso, ao estruturar e documentar decisões arquiteturais de forma acessível, reduz-se o tempo gasto na busca por conhecimento e na resolução de dúvidas, **aumentando a produtividade** das equipes e permitindo que os desenvolvedores se concentrem em atividades de maior valor agregado.

O estudo também identifica **desafios práticos** na implementação da **Gestão do Conhecimento Arquitetônico (GCA)**, destacando a necessidade de **maior engajamento dos colaboradores**, **treinamentos** e **suporte da liderança**. Dessa forma, alcançamos

insights valiosos para que a indústria direcione esforços estratégicos na **adoção bem-sucedida de práticas de GCA**. Para isso, é sugerida a criação de uma **trilha de aprendizado** e a realização de **treinamentos contínuos**, promovendo uma cultura organizacional que valoriza o **compartilhamento de conhecimento** e a **documentação de decisões arquiteturais**.

Ao garantir que as decisões arquiteturais possam ser documentadas e compartilhadas de maneira estruturada, este trabalho contribui diretamente para a **melhoria da qualidade dos produtos desenvolvidos**. Já que essa abordagem é especialmente relevante em **ambientes ágeis**, onde a **consistência** e a **manutenibilidade do código** são aspectos cruciais para a evolução dos sistemas.

A **centralização do conhecimento** e a **redução de redundâncias** também resultam em **economia de recursos**, tanto em termos de **tempo** quanto de **esforço**, beneficiando organizações que buscam **otimizar processos e reduzir custos operacionais**.

Em síntese, este estudo contribui para a indústria de software ao propor um **processo validado** para a **Gestão do Conhecimento Arquitetônico (GCA)**, promovendo **maior colaboração entre equipes, facilitando a integração de novos colaboradores, reduzindo o tempo e esforço no desenvolvimento de software e melhorando a qualidade dos produtos**. Além disso, a pesquisa valida o uso de **ferramentas já existentes**, identifica pontos de melhoria e promove uma **cultura organizacional centrada no conhecimento**. Essas contribuições geram um impacto direto na **eficiência** e na **competitividade** das organizações de software.

8.4 Publicações e perspectivas futuras

Foi realizada uma contribuição desta dissertação, conforme relatado no Capítulo 3 com uma RSL publicada em conferência. Sua referência segue abaixo:

- SOUZA, Wesley; SILVA, Williamson; GUEDES, Gilleanes Thorwald Araujo. Knowledge Management Diagnosis in Software Organizations: A Systematic Literature Review Extension. In: Proceedings of the 20th Brazilian Symposium on Information Systems. 2024. p. 1-10. DOI: <https://doi.org/10.1145/3658271.3658273>.

Os resultados apresentados nesta dissertação permitem o surgimento de novas perspectivas de pesquisa para a Gestão do Conhecimento Arquitetônico (GCA). A seguir,

citamos algumas possibilidades:

- **Criação de um catálogo de ações práticas para melhoria da estrutura dos projetos e seus respectivos *feedbacks*:** Mostrar como as práticas sugeridas pela GCA foram utilizadas pelas equipes, validar sua contribuição no projeto, verificar se trouxeram melhorias na qualidade e no tempo de desenvolvimento ou se não se mostraram favoráveis para determinado produto.
- **Melhoria no modelo sugerido para GCA:** O sistema pode ser aprimorado para incluir funcionalidades como um scaffold para o início de projetos, acesso à mensagem e ao microfrontend, além de um mapeamento completo de todas as ferramentas e funcionalidades disponíveis em todos os produtos. Isso garantiria a obediência ao padrão arquitetural definido pela empresa e tornaria cada vez mais clara a comunicação entre os produtos em uma estrutura no modelo SaaS.
- **Realização de testes de usabilidade:** O sistema centralizador de GCA pode ser aprimorado para ter uma interface mais fluida e adaptativa aos usuários, trazendo somente informações relevantes para cada cenário. Por exemplo, não é interessante que o pessoal responsável pelo front-end tenha acesso a estruturas arquiteturais de back-end, mas essa opção pode ser disponibilizada para aqueles que desejam transitar entre as áreas.

REFERÊNCIAS

- ALAVI, M.; LEIDNER, D. E. Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues. **MIS quarterly**, JSTOR, v. 25, n. 1, p. 107–136, 2001.
- AMPATZOGLOU, A. et al. The effect of gof design patterns on stability: a case study. **IEEE Transactions on Software Engineering**, IEEE, v. 41, n. 8, p. 781–802, 2015.
- ANDRIYANI, Y.; HODA, R.; AMOR, R. Understanding knowledge management in agile software development practice. In: SPRINGER. **Knowledge Science, Engineering and Management: 10th International Conference, KSEM 2017, Melbourne, VIC, Australia, August 19-20, 2017, Proceedings 10**. [S.l.], 2017. p. 195–207.
- AURUM, A.; DANESHGAR, F.; WARD, J. Investigating knowledge management practices in software development organisations—an australian experience. **Information and Software Technology**, Elsevier, v. 50, n. 6, p. 511–533, 2008.
- BADAMPUDI, D.; WOHLIN, C.; GORSCHER, T. An evaluation of knowledge translation in software engineering. In: IEEE. **International Symposium on Empirical Software Engineering and Measurement (ESEM)**. [S.l.], 2019. p. 1–6.
- BALDAUF, M.; FROHLICH, P.; SALOMON, R. Onboarding in software teams: Challenges and best practices. **Proceedings of the 2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)**, IEEE, p. 1–6, 2019.
- BASS, L.; CLEMENTS, P.; KAZMAN, R. **Software architecture in practice**. [S.l.]: Addison-Wesley Professional, 2021.
- BECKETT, A. J.; WAINWRIGHT, C. E.; BANCE, D. Knowledge management: strategy or software? **Management Decision**, MCB UP Ltd, v. 38, n. 9, p. 601–606, 2000.
- BETZ, S.; OBERWEIS, A.; STEPHAN, R. Knowledge transfer in offshore outsourcing software development projects: An analysis of the challenges and solutions from german clients. **Expert Systems**, Wiley Online Library, v. 31, n. 3, p. 282–297, 2014.
- BHAT, M. et al. Meta-model based framework for architectural knowledge management. In: **Proceedings of the 10th European Conference on Software Architecture Workshops**. [S.l.: s.n.], 2016. p. 1–7.
- BHAT, M.; SHUMAIEV, K.; MATTHES, F. Towards a framework for managing architectural design decisions. In: **Proceedings of the 11th European Conference on Software Architecture: Companion Proceedings**. [S.l.: s.n.], 2017. p. 48–51.
- BJØRNSON, F. O.; DINGSØYR, T. Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used. **Information and software technology**, Elsevier, v. 50, n. 11, p. 1055–1068, 2008.
- BOER, R. C. D.; VLIET, H. van. Architectural knowledge discovery with latent semantic analysis: Constructing a reading guide for software product audits. **Journal of Systems and Software**, Elsevier, v. 81, n. 9, p. 1456–1469, 2008.

BOER, R. C. de et al. Architectural knowledge: Getting to the core. In: SPRINGER. **Software Architectures, Components, and Applications: Third International Conference on Quality of Software Architectures, QoSA 2007, Medford, MA, USA, July 11-23, 2007, Revised Selected Papers 3**. [S.l.], 2007. p. 197–214.

BORREGO, G. et al. Towards a reduction in architectural knowledge vaporization during agile global software development. **Information and Software Technology**, Elsevier, v. 112, p. 68–82, 2019.

BOSCH, J. Software architecture: The next step. In: SPRINGER. **Software Architecture: First European Workshop, EWSA 2004, St Andrews, UK, May 21-22, 2004. Proceedings 1**. [S.l.], 2004. p. 194–199.

BRADBURY, J. S. et al. A survey of self-management in dynamic software architecture specifications. In: **Proceedings of the 1st ACM SIGSOFT workshop on Self-managed systems**. [S.l.: s.n.], 2004. p. 28–33.

CABRAL, A. R. Y.; RIBEIRO, M. B.; NOLL, R. P. Knowledge management in agile software projects: A systematic review. **Journal of Information & Knowledge Management**, World Scientific, v. 13, n. 01, p. 1450010, 2014.

CAPILLA, R. et al. 10 years of software architecture knowledge management: Practice and future. **Journal of Systems and Software**, Elsevier, v. 116, p. 191–205, 2016.

CAPILLA, R. et al. Architectural knowledge management in global software development: A systematic review. **Journal of Software: Evolution and Process**, Wiley Online Library, v. 28, n. 7, p. 563–588, 2016.

CASSIANI, S. H. D. B.; CALIRI, M. H. L.; PELÁ, N. T. R. A teoria fundamentada nos dados como abordagem da pesquisa interpretativa. **Revista latino-americana de enfermagem**, SciELO Brasil, v. 4, p. 75–88, 1996.

CHANG, F. et al. A maintenance decision-making oriented collaborative cross-organization knowledge sharing blockchain network for complex multi-component systems. **Journal of cleaner production**, Elsevier, v. 282, p. 124541, 2021.

CHOUSEINOGLU, O. et al. Aiolos: A model for assessing organizational learning in software development organizations. **Information and Software Technology**, Elsevier, v. 55, n. 11, p. 1904–1924, 2013.

CLERC, V. Towards architectural knowledge management practices for global software development. In: **Proceedings of the 3rd international workshop on Sharing and reusing architectural knowledge**. [S.l.: s.n.], 2008. p. 23–28.

CONTRIBUTORS, S. W. **Spring Wolf: AsyncAPI Documentation for Spring Boot**. 2023. <<https://github.com/springwolf/springwolf-core>>. Acessado em: 10 out. 2023.

CONTRIBUTORS, T. **TypeDoc: Documentation Generator for TypeScript**. 2023. <<https://typedoc.org/>>. Acessado em: 10 out. 2023.

CORBIN, R. D.; DUNBAR, C. B.; ZHU, Q. A three-tier knowledge management scheme for software engineering support and innovation. **Journal of systems and software**, Elsevier, v. 80, n. 9, p. 1494–1505, 2007.

COSTA, E. P.; POLITANO, P. R.; PEREIRA, N. A. Exemplo de aplicação do método de pesquisa-ação para a solução de um problema de sistema de informação em uma empresa produtora de cana-de-açúcar. **Gestão & Produção**, SciELO Brasil, v. 21, p. 895–905, 2014.

COUGHLAN, P.; COGHLAN, D. Action research for operations management. **International journal of operations & production management**, MCB UP Ltd, v. 22, n. 2, p. 220–240, 2002.

DAKHLI, S. B. D.; CHOUIKHA, M. B. The knowledge-gap reduction in software engineering. In: IEEE. **2009 Third International Conference on Research Challenges in Information Science**. [S.l.], 2009. p. 287–294.

DARAOJIMBA, E. C. et al. Comprehensive review of agile methodologies in project management. **Computer Science & IT Research Journal**, v. 5, n. 1, p. 190–218, 2024.

DAWSON, R. 12 steps to successful knowledge management implementation. **Proceedings of the European Conference on Knowledge Management**, Loughborough University, p. 209–215, 2009.

DEMIRSOY, A.; PETERSEN, K. Semantic knowledge management system to support software engineers: implementation and static evaluation through interviews at ericsson. **e-Informatica Software Engineering Journal**, v. 12, n. 1, p. 237–263, 2018.

DESOUZA, K. C.; AWAZU, Y.; WAN, Y. Factors governing the consumption of explicit knowledge. **Journal of the American society for information science and technology**, Wiley Online Library, v. 57, n. 1, p. 36–43, 2006.

DICK, B. **A beginner's guide to action research**. 2000.

DINGSØYR, T.; VLIET, H. van. Introduction to software architecture and knowledge management. In: **Software Architecture Knowledge Management: Theory and Practice**. [S.l.]: Springer, 2009. p. 1–17.

DORAIRAJ, S.; NOBLE, J.; MALIK, P. Knowledge management in distributed agile software development. In: IEEE. **2012 Agile Conference**. [S.l.], 2012. p. 64–73.

ELORANTA, V.-P.; KOSKIMIES, K. Lightweight architecture knowledge management for agile software development. In: **Agile Software Architecture**. [S.l.]: Elsevier, 2014. p. 189–213.

ENGINEERING, S. Backstage: A developer portal for the spotify ecosystem. **Spotify Engineering Blog**, 2020. Available from Internet: <<https://engineering.atspotify.com/2020/03/17/backstage-a-developer-portal-for-the-spotify-ecosystem/>>.

FANNOUN, S.; KERINS, J. Towards organisational learning enhancement: assessing software engineering practice. **The learning organization**, Emerald Publishing Limited, v. 26, n. 1, p. 44–59, 2019.

FARENHORST, R.; BOER, R. C. de. Knowledge management in software architecture: State of the art. **Software Architecture Knowledge Management: Theory and Practice**, Springer, p. 21–38, 2009.

FEHÉR, P.; GÁBOR, A. The role of knowledge management supporters in software development companies. **Software Process: Improvement and Practice**, Wiley Online Library, v. 11, n. 3, p. 251–260, 2006.

FOWLER, M. **Analysis patterns: reusable object models**. [S.l.]: Addison-Wesley Professional, 1997.

FOWLER, M. Who needs an architect? **IEEE SOFTWARE**, IEEE INSTITUTE OF ELECTRICAL AND ELECTRONICS, v. 20, n. 5, p. 11–13, 2003.

FOWLER, M. **Patterns of enterprise application architecture**. [S.l.]: Addison-Wesley, 2012.

GARLAN, D.; ALLEN, R.; OCKERBLOOM, J. Architectural mismatch: Why reuse is so hard. **IEEE Software**, IEEE, v. 12, n. 6, p. 17–26, 1995.

GASSTON, J.; HALLORAN, P. Continuous software process improvement requires organisational learning: An australian case study. **Software Quality Journal**, Springer, v. 8, p. 37–51, 1999.

GHOBADI, S.; MATHIASSEN, L. Risks to effective knowledge sharing in agile software teams: A model for assessing and mitigating risks. **Information systems journal**, Wiley Online Library, v. 27, n. 6, p. 699–731, 2017.

GOLDONI, V.; OLIVEIRA, M. Knowledge management metrics in software development companies in brazil. **Journal of knowledge management**, Emerald Group Publishing Limited, v. 14, n. 2, p. 301–313, 2010.

GOPAL, J. et al. Integration of fuzzy dematel and fmcdm approach for evaluating knowledge transfer effectiveness with reference to gsd project outcome. **International Journal of Machine Learning and Cybernetics**, Springer, v. 9, p. 225–241, 2018.

HANSEN, B. H.; KAUTZ, K. Knowledge mapping: A technique for identifying knowledge flows in software organisations. In: SPRINGER. **European Conference on Software Process Improvement**. [S.l.], 2004. p. 126–137.

HILLIARD, R. **ISO/IEC/IEEE 42010**. 2011.

IULIANA, S. A knowledge management practice investigation in romanian software development organizations. **WSEAS Transactions on Computers**, World Scientific and Engineering Academy and Society (WSEAS) Stevens Point . . . , v. 8, n. 3, p. 459–468, 2009.

IVARSSON, M.; GORSCHKEK, T. Tool support for disseminating and improving development practices. **Software Quality Journal**, Springer, v. 20, p. 173–199, 2012.

IVARSSON, T. G. M. Praction selection framework. **International Journal of Software Engineering and Knowledge Engineering**, World Scientific and Engineering Academy and Society (WSEAS) Stevens Point . . . , p. 17–58, 2012.

IVES, W.; TORREY, B.; GORDON, C. Knowledge management: an emerging discipline with a long history. **Journal of Knowledge Management**, MCB UP Ltd, v. 1, n. 4, p. 269–274, 1997.

JANSEN, A.; BOSCH, J. Software architecture as a set of architectural design decisions. In: IEEE. **5th Working IEEE/IFIP Conference on Software Architecture (WICSA'05)**. [S.l.], 2005. p. 109–120.

JANSEN, A.; BOSCH, J.; AVGERIOU, P. Architectural knowledge management: An empirical study into support for architects. **Journal of Systems and Software**, Elsevier, v. 82, n. 8, p. 1450–1464, 2009.

JANSEN, A.; BOSCH, J.; AVGERIOU, P. Documenting after the fact: recovering architectural design decisions. **Journal of Systems and Software**, Elsevier, v. 81, n. 4, p. 536–557, 2009.

JINKINGS, E. Y. de M. Gestão do conhecimento: um estudo de caso sobre a criação do conhecimento em empresas do polo industrial de manaus. 2022.

JOEL-EDGAR, S.; GOPSILL, J. Understanding user requirements in context: a case study of developing a visualisation tool to map skills in an engineering organisation. In: IEEE. **2018 International Conference on Information Management and Processing (ICIMP)**. [S.l.], 2018. p. 6–10.

JR., F. M. d. V.; WERNER, C. M. L. Suporte dos padrões à evolução de processos de desenvolvimento de software. In: SBC. **Anais do XI Simpósio Brasileiro de Engenharia de Software**. 1997. p. 131–146. Available from Internet: <<https://sol.sbc.org.br/index.php/sbes/article/view/24045>>.

JÚNIOR, J. F. d. C. et al. Um estudo sobre o uso da escala de likert na coleta de dados qualitativos e sua correlação com as ferramentas estatísticas. **Contribuciones a Las Ciencias Sociales, São José dos Pinhais**, v. 17, n. 1, p. 360–376, 2024.

JURADO, J. L. et al. Model for the improvement of knowledge management processes based on the use of gamification principles in companies in the software sector. In: SPRINGER. **7th International Conference on Software Process Improvement (CIMPS 2018) 7**. [S.l.], 2019. p. 142–151.

KANE, M. J.; URBANEK, S. On the programmatic generation of reproducible documents. **arXiv preprint arXiv:2007.12631**, 2020. Available from Internet: <<https://arxiv.org/abs/2007.12631>>.

KEELING, M. Love unrequited: The story of architecture, agile, and how architecture decision records brought them together. **IEEE Software**, IEEE, v. 39, n. 4, p. 90–93, 2022.

KHOMH, F.; GUÉHÉNEUC, Y.-G. Design patterns impact on software quality: Where are the theories? In: IEEE. **2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER)**. [S.l.], 2018. p. 15–25.

KHOSRAVI, A.; NILASHI, M. et al. Toward software quality enhancement by customer knowledge management in software companies. **Telematics and Informatics**, Elsevier, v. 35, n. 1, p. 18–37, 2018.

KHOSRAVI, A. et al. Customer knowledge management in enterprise software development companies: organizational, human and technological perspective. **Management Systems in Production Engineering**, v. 30, n. 4, p. 291–297, 2022.

KITCHENHAM, B.; CHARTERS, S. Guidelines for performing systematic literature reviews in software engineering. Citeseer, 2007.

KRUCHTEN, P. **The Rational Unified Process: An Introduction**. 3. ed. [S.l.]: Addison-Wesley, 2004.

KRUCHTEN, P.; LAGO, P.; VLIET, H. V. Building up and reasoning about architectural knowledge. In: SPRINGER. **International conference on the quality of software architectures**. [S.l.], 2006. p. 43–58.

KULKARNI, U.; LOUIS, R. S. Organizational self assessment of knowledge management maturity. **Proceedings of the 9th Americas Conference on Information Systems**, 2003.

LAGERBERG, L. et al. The impact of agile principles and practices on large-scale software development projects: A multiple-case study of two projects at ericsson. In: IEEE. **International Symposium on Empirical Software Engineering and Measurement**. [S.l.], 2013. p. 348–356.

LAGO, P.; AVGERIOU, P. First workshop on sharing and reusing architectural knowledge. **ACM SIGSOFT Software Engineering Notes**, ACM New York, NY, USA, v. 31, n. 5, p. 32–36, 2006.

LEHNER, F. Measuring km success and km service quality with knowmetrix—first experiences from a case study in a software company. In: SPRINGER. **International Conference on Knowledge Science, Engineering and Management: , (KSEM 2009)**. [S.l.], 2009. p. 335–346.

LETHBRIDGE, T. C. Evaluating a domain-specialist-oriented knowledge management system. **International Journal of Human-Computer Studies**, Elsevier, v. 52, n. 6, p. 961–990, 2000.

LEVY, M. et al. Knowledge management culture audit: Capturing tacit perceptions and barriers. **Americas' Conference on Information Systems (AMCIS)**, p. 1–8, 2008.

LEVY, M. et al. Uncovering cultural perceptions and barriers during knowledge audit. **Journal of Knowledge Management**, Emerald Group Publishing Limited, v. 14, n. 1, p. 114–127, 2010.

LIANG, P.; AVGERIOU, P. Tools and technologies for architecture knowledge management. In: **Software Architecture Knowledge Management: Theory and Practice**. [S.l.]: Springer, 2009. p. 91–111.

LINDVALL, M.; RUS, I. Knowledge management for software organizations. **Managing software engineering knowledge**, Springer, p. 73–94, 2003.

LUCHESE, E. S. F. Gestão do conhecimento nas organizações. **CET–Companhia de Engenharia de Tráfego de São Paulo**. São Paulo, 2012.

MACIEL, C. P. et al. Knowledge management diagnostics in software development organizations: a systematic literature review. In: **Proceedings of the XVII Brazilian Symposium on Software Quality**. [S.l.: s.n.], 2018. p. 141–150.

MCKAY, J.; MARSHALL, P. The dual imperatives of action research. **Information Technology & People**, MCB UP Ltd, v. 14, n. 1, p. 46–59, 2001.

MEDVIDOVIC, N.; TAYLOR, R. N. A classification and comparison framework for software architecture description languages. **IEEE Transactions on software engineering**, IEEE, v. 26, n. 1, p. 70–93, 2000.

MICROSOFT. **Azure devops**. 2024. Available from Internet: <<https://azure.microsoft.com/pt-br/products/devops>>.

MITCHELL, S. M.; SEAMAN, C. B. A knowledge mapping technique for project-level knowledge flow analysis. In: IEEE. **2011 international symposium on empirical software engineering and measurement**. [S.l.], 2011. p. 347–350.

MITCHELL, S. M.; SEAMAN, C. B. Could removal of project-level knowledge flow obstacles contribute to software process improvement? a study of software engineer perceptions. **Information and Software Technology**, Elsevier, v. 72, p. 151–170, 2016.

MONTONI, M. et al. Knowledge acquisition and communities of practice: An approach to convert individual knowledge into multi-organizational knowledge. In: SPRINGER. **Advances in Learning Software Organizations: 6th International Workshop, LSO 2004, Banff, Canada, June 20-21, 2004. Proceedings 6**. [S.l.], 2004. p. 110–121.

NAMBU, R.; SUEHIRO, K.; YAMAGUCHI, T. A knowledge transfer support system from text-based work reports with domain ontologies. In: SPRINGER. **12th Joint Conference on Knowledge-Based Software Engineering (JCKBSE 2018)**. [S.l.], 2019. p. 137–146.

NEVES, S. M. et al. Risk management in software projects through knowledge management techniques: cases in brazilian incubated technology-based firms. **International Journal of Project Management**, Elsevier, v. 32, n. 1, p. 125–138, 2014.

NEVES, S. M. et al. Risk management in software projects through knowledge management techniques: cases in brazilian incubated technology-based firms. **International Journal of Project Management**, Elsevier, v. 32, n. 1, p. 125–138, 2014.

OQUENDO, F. Software architecture challenges and emerging research in software-intensive systems-of-systems. In: SPRINGER. **Software Architecture: 10th European Conference, ECSA 2016, Copenhagen, Denmark, November 28–December 2, 2016, Proceedings 10**. [S.l.], 2016. p. 3–21.

OURIQUES, R. A. B. et al. Knowledge management strategies and processes in agile software development: a systematic literature review. **International journal of software engineering and knowledge engineering**, World Scientific, v. 29, n. 03, p. 345–380, 2019.

ÖZEN, G. et al. Assessing organizational learning in it organizations: an experience report from industry. In: IEEE. **2013 Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement**. [S.l.], 2013. p. 253–258.

ÖZEN, G. et al. Assessing organizational learning in it organizations: an experience report from industry. In: IEEE. **2013 Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement**. [S.l.], 2013. p. 253–258.

OZKAYA, I.; WALLIN, P.; AXELSSON, J. Architecture knowledge management during system evolution: observations from practitioners. In: **Proceedings of the 2010 ICSE Workshop on Sharing and Reusing Architectural Knowledge**. [S.l.: s.n.], 2010. p. 52–59.

PAIVA, E. L. Conhecimento organizacional na área de produção e sua relação com o ambiente industrial. **RAM. Revista de Administração Mackenzie**, SciELO Brasil, v. 7, n. 4, p. 60–77, 2006.

PAULZEN, O. et al. A maturity model for quality improvement in knowledge management. **International Symposium on Empirical Software Engineering and Measurement Enabling Organizations and Society through Information Systems (ACIS)**, p. 243–253, 2002.

PEDRAZA-GARCIA, G.; ASTUDILLO, H.; CORREAL, D. An approach for software knowledge sharing based on architectural decisions. In: IEEE. **2016 XLII Latin American Computing Conference (CLEI)**. [S.l.], 2016. p. 1–10.

PETERSEN, K. et al. Action research as a model for industry-academia collaboration in the software engineering context. In: **Proceedings of the 2014 international workshop on Long-term industrial collaboration on software engineering**. [S.l.: s.n.], 2014. p. 55–62.

POHL, K.; SIKORA, E. Structuring the co-design of requirements and architecture. In: SPRINGER. **Requirements Engineering: Foundation for Software Quality: 13th International Working Conference, REFSQ 2007, Trondheim, Norway, June 11-12, 2007. Proceedings 13**. [S.l.], 2007. p. 48–62.

RABELO, J. de H. et al. Knowledge management and organizational culture in a software organization—a case study. In: IEEE. **8th International Workshop on Cooperative and Human Aspects of Software Engineering**. [S.l.], 2015. p. 89–92.

RAZAVIAN, M.; PAECH, B.; TANG, A. The vision of on-demand architectural knowledge systems as a decision-making companion. **Journal of Systems and Software**, Elsevier, v. 198, p. 111560, 2023.

RICHARDS, M. **Software architecture patterns**. [S.l.]: O'Reilly Media, Incorporated 1005 Gravenstein Highway North, Sebastopol, CA . . . , 2015.

SANGAIAH, A. K. et al. An integrated fuzzy dematel, topsis, and electre approach for evaluating knowledge transfer effectiveness with reference to gsd project outcome. **Neural Computing and Applications**, Springer, v. 28, p. 111–123, 2017.

SHAW, M.; GARLAN, D. **Software architecture: perspectives on an emerging discipline**. [S.l.]: Prentice-Hall, Inc., 1996.

SHULL, F.; CARVER, J.; TRAVASSOS, G. H. An empirical methodology for introducing software processes. **ACM SIGSOFT Software Engineering Notes**, ACM New York, NY, USA, v. 26, n. 5, p. 288–296, 2001.

SILVA, L. G.; FINGER, A.; SILVA, W. Um estudo exploratório sobre a aceitação do ux curve aplicado a ambientes virtuais de aprendizagem. In: SBC. **Anais da VI Escola Regional de Engenharia de Software**. [S.l.], 2022. p. 31–40.

SILVA, W.; STEINMACHER, I.; CONTE, T. Students' and instructors' perceptions of five different active learning strategies used to teach software modeling. **IEEE Access**, IEEE, v. 7, p. 184063–184077, 2019.

SINGH, S. K. Role of leadership in knowledge management: a study. **Journal of knowledge management**, Emerald Group Publishing Limited, v. 12, n. 4, p. 3–15, 2008.

SOFTWARE, J. **Jira Software Cloud Developer**. 2024. Available from Internet: <<https://developer.atlassian.com/cloud/jira/software/rest>>.

SOLMS, F. What is software architecture? In: **Proceedings of the south african institute for computer scientists and information technologists conference**. [S.l.: s.n.], 2012. p. 363–373.

SOMMERVILLE, I. Software engineering (ed.). **America: Pearson Education Inc**, 2015.

SOUZA, W.; SILVA, W.; GUEDES, G. Technical report and dataset concerning the systematic literature review extension on knowledge management diagnosis in software organizations. In: . Zenodo, 2024. Available from Internet: <<https://doi.org/10.5281/zenodo.10814854>>.

SOUZA, W.; SILVA, W.; GUEDES, G. T. A. Knowledge management diagnosis in software organizations: A systematic literature review extension. In: **Proceedings of the 20th Brazilian Symposium on Information Systems**. [S.l.: s.n.], 2024. p. 1–10.

SPOLADORE, D.; PESSOT, E. An evaluation of agile ontology engineering methodologies for the digital transformation of companies. **Computers in Industry**, Elsevier, v. 140, p. 103690, 2022.

SPOTIFY. **Backstage**. 2024. Available from Internet: <<https://backstage.io/docs/overview/what-is-backstage/>>.

Spotify Open Source Team. **Backstage Documentation**. [S.l.], 2025. Acesso em: 27 fev. 2025. Available from Internet: <<https://backstage.spotify.com/docs/portal/core-features-and-plugins/>>.

STARON, M. **Action research in software engineering**. [S.l.]: Springer, 2020.

STRINGER, E. T. **Action research**. Thousand Oaks, CA: SAGE Publications, 2014.

SZULANSKI, G. The process of knowledge transfer: A diachronic analysis of stickiness. **Organizational behavior and human decision processes**, Elsevier, v. 82, n. 1, p. 9–27, 2000.

TAHERI, L. et al. Identifying knowledge components in software requirement elicitation. In: IEEE. **2014 IEEE International Conference on Industrial Engineering and Engineering Management**. [S.l.], 2014. p. 286–291.

TAHERI, L. et al. A knowledge audit model to assess the knowledge in requirement elicitation process. In: IEEE. **2015 9th Malaysian Software Engineering Conference (MySEC)**. [S.l.], 2015. p. 106–111.

TANG, A. et al. A comparative study of architecture knowledge management tools. **Journal of Systems and Software**, Elsevier, v. 83, n. 3, p. 352–370, 2010.

THIOLLENT, M. **Metodologia da Pesquisa-Ação**. 3ª. ed. São Paulo, Brasil: Cortez Editora, 1997.

THIOLLENT, M. **Metodologia da pesquisa-ação**. 18. ed. São Paulo: Cortez, 2011.

TODOROVA, N.; MILLS, A. M. Why do people share?: A study of intrinsic and extrinsic motivation to share knowledge in organisations. **International Journal of Knowledge Management (IJKM)**, IGI Global, v. 14, n. 3, p. 1–20, 2018.

TOMLIN, W. C.; TOMLIN, W. C. What's a persona? **UX Optimization: Combining Behavioral UX and Usability Testing Data to Optimize Websites**, Springer, p. 11–18, 2018.

TRIPP, D. **Action Research: A Methodological Introduction**. Brasília: Education and Research Development, 2005. ISBN 978-85-7034-432-5.

TUZOVSKY, A.; YAMPOLSKY, V.; CHIRIKOV, S. Ontological knowledge management system development. In: IEEE. **Proceedings. The 9th Russian-Korean International Symposium on Science and Technology, 2005. KORUS 2005**. [S.l.], 2005. p. 718–720.

UNKELOS-SHPIGEL, N.; HADAR, I.; LEVY, M. Enhancing deployment requirements' traceability via knowledge management audit (s). In: **SEKE**. [S.l.: s.n.], 2013. p. 574–577.

VENKATESH, V.; DAVIS, F. D. A theoretical extension of the technology acceptance model: Four longitudinal field studies. **Management science, INFORMS**, v. 46, n. 2, p. 186–204, 2000.

VERONEZ, M. et al. Um catálogo de requisitos não-funcionais e padrões arquiteturais para microsserviços. Universidade Estadual do Oeste do Paraná, 2022.

VLIET, H. V. Software architecture knowledge management. In: IEEE. **19th Australian conference on software engineering (aswec 2008)**. [S.l.], 2008. p. 24–31.

WAHEED, S. et al. Improving knowledge sharing in distributed software development. **International Journal of Advanced Computer Science and Applications**, Science and Information (SAI) Organization Limited, v. 10, n. 6, 2019.

WANG, S.; NOE, R. A. Knowledge sharing: A review and directions for future research. **Human Resource Management Review**, Elsevier, v. 17, n. 2, p. 172–185, 2007.

WEINREICH, R.; GROHER, I. Software architecture knowledge management approaches and their support for knowledge management activities: A systematic literature review. **Information and Software Technology**, Elsevier, v. 80, p. 265–286, 2016.

WU, Y.; SENOO, D.; MAGNIER-WATANABE, R. Diagnosis for organizational knowledge creation: an ontological shift seci model. **Journal of Knowledge Management**, Emerald Group Publishing Limited, v. 14, n. 6, p. 791–810, 2010.

APÊNDICE A — QUESTIONÁRIO 1

Tabela 12 – Respostas Qualitativas 1

| |
|--|
| Na sua opinião quais pontos deveriam ser melhor comunicados dentro da organização? |
| 1 - [Deveriam ser comunicados] Qual o objetivo e [o porquê de] como as decisões [que] foram tomadas sobre as definições estruturais [do código] e metas estabelecidas [foram repassadas para as equipes ou interessados]. |
| 2 - [Deveriam ter como objetivo] Incentivar a colaboração entre as equipes de desenvolvimento e promover o compartilhamento ativo de conhecimento por meio de workshops. |
| 3 - Hoje é clara a falta de comunicação entre arquitetura e desenvolvedores, muitas coisas são criadas/atualizadas sem nenhum comunicado prévio. Outro ponto é que a arquitetura deveria ser uma espécie de facilitadora do trabalho, porém, a realidade é que eles fazem mais o trabalho de barrar ideias e iniciativas de melhoria partindo dos times enquanto eles não propõem nada para o mesmo fim. Sinto como se eles fossem o juiz, júri e executor, novas ideias não são bem vindas, enquanto, por parte deles não há ideia alguma, nova ou antiga. Então hoje para mim é muito obscuro o real papel da equipe de arquitetura dentro da empresa, não vejo facilidade alguma sendo criada, pelo contrário, vejo dificuldades! |
| 4 - [Deveriam trazer melhorias na] A comunicação entre os times de outras áreas deveria ser mais fácil |
| 5 - Não tenho uma resposta |
| 6 - Não tenho uma resposta |
| *Obtivemos apenas 6 respostas e 10 abstenções em relação ao questionamento |

Tabela 13 – Respostas Qualitativas 2

| |
|---|
| Na sua opinião quais informações são importantes de se ter um fácil acesso dentro do time, para desenvolvedores experientes no projeto e novos desenvolvedores que chegam ao projeto, trazendo uma melhor comunicação e melhoria no tempo gasto de desenvolvimento? |
| *(1) Transferência de conhecimento |
| 2 - Uma documentação detalhada do projeto, incluindo arquitetura de software, design/interfaces e onde se encontram as documentações de negócio. Isso ajuda os desenvolvedores a entenderem o contexto do projeto |
| 4 - Documentação técnica e de fluxos básicos |
| Uma página com documentações gerais sobre os projetos (uma página no drive com várias pastas como RFCs, evidências de testes realizados, apresentações de review etc.); - Documento de boas vindas contendo informações iniciais sobre o projeto: Quais acessos solicitar, quais apps configurar, vídeos explicativos sobre o negócio, qual 'arch' [modelo de arquitetura] é usado...; - Dashboards e informações que mostrem a saúde das aplicações; - Informações de acesso às máquinas, banco de dados, git; Acredito que vale a pena refletir sobre quais informações usamos com mais frequência para deixá-las em um local de fácil acesso. Existem também aquelas informações mais complexas, que acredito, ser uma boa prática deixá-las salvas em algum lugar para que o time acesse no dia-a-dia |
| *(2) Onboarding de novos integrantes às equipes |
| 4 - Documentação técnica e de fluxos básicos |
| 5 - Toda a documentação do projeto, técnica e de negócio para que novo integrante consiga ficar por dentro o mais rápido possível |
| *(3) Padrões organizacionais para projetos |
| 3 - Acredito que falte um boilerplate/scaffold padronizado para que os times possam iniciar novos projetos de forma ágil e adequada, como também, um suporte maior da arquitetura com relação ao entendimento da arquitetura proposta. |
| 1 - Padrões de projeto, estrutura e metodologia de trabalho são pontos que sempre ficam nebulosos da empresa |
| *Obtivemos apenas 8 respostas e 8 abstenções em relação ao questionamento |

APÊNDICE B — QUESTIONÁRIO 2

Tabela 14 – (KT1) Em se tratando do compartilhamento de conhecimento pelo time de arquitetura para com os times de produto e os times de produto entre si, o Backstage® se mostra promissor? Fale sua opinião e sugestões a respeito desse tema. (parte A)

| |
|--|
| 1. Atende porém é necessário melhorar o engajamento com treinamentos e workshop a respeito |
| 4 - Ainda processando essa informação. |
| 7 - Sem sugestão. |
| 11 - Conheço muito pouco do backstage. |
| 16 - Ainda estou avaliando para tomar alguns conclusões. |
| 17 - Não tenho sugestão no momento. |
| 18 - <em branco> |
| 19 - Nada. |
| 24 - No momento não possuo muita opinião sobre isso |
| 2. Atende sendo um motivador para o compartilhamento |
| 2 - A ideia de ter um único local para o compartilhamento é muito vantajosa pois encurtamos a barreira da força de vontade dos colaboradores com os materiais. Só de pensar em ter que usar vários lugares para atualizar uma informação já desmotiva a pessoa de querer compartilhar a informação. |
| 4 - Creio que com um único lugar que concentre boa parte das informações fica mais simples de compartilhar o conhecimento. |
| 7 - O backstage ajuda a melhorar a comunicação entre o time de arquitetura e os de desenvolvimento por ter a documentação unificada e visibilidade dos serviços como APIs para que os devs possam acessar com facilidade. |
| 10 - Acredito que a centralização de conhecimento permite uma melhoria no desenvolvimento dos times |
| 12 - Extrema importância nós do time de produto andarmos lado a lado com a arquitetura, do meu ponto de vista esse time contém os melhores profissionais de TI da TOTVS e usá-los como referência para o crescimento é algo surreal, essa foi a minha experiência quando precisei solicitar apoio desse time. |
| 15 - Já enfrentei diversas situações onde não compreendia completamente ações que precisavam ser tomadas que vieram da arquitetura. Acredito que o backstage pode ser a solução ideal para esses problemas, pois ele proporciona alinhamento total, garantindo que todos tenham o mesmo nível de informação e conhecimento necessário para lidar eficazmente com as transformações em curso. |

Tabela 15 – (KT1) Em se tratando do compartilhamento de conhecimento pelo time de arquitetura para com os times de produto e os times de produto entre si, o Backstage® se mostra promissor? Fale sua opinião e sugestões a respeito desse tema.(parte B)

| |
|--|
| 16 - Se conseguir centralizar o conhecimento em um único lugar, facilita a disseminação. |
| 18 - Assim, fica muito mais fácil conhecer a arquitetura de um sistema ajudando na sua manutenção ou desenvolvimento. |
| 19 - Neste caso acredito que serviria como uma forma de publicar pequenas alterações, aquelas que não precisem de comunicados formais e/ou formalizar algo que já foi passado, deixando assim para uso como documentação. |
| 20 - Ele facilita a colaboração e a automação de processos. Melhora a eficiência e o compartilhamento do conhecimento entre os times. |
| 22 - A redução da distância entre squads sempre será benéfica para o desenvolvimento de um produto. No que diz respeito à aproximação das equipes de desenvolvedores com a equipe de arquitetura, se torna essencial estreitar o relacionamento para melhor entendimento das soluções propostas. |
| 25 - É uma forma de concentrar informações importantes, documentações, etc e pode ser útil para todo o time. |
| 3. Atende como o esperado |
| 3 - Entendo que sim. |
| 6 - Sim |
| 9 - Concordo |
| 14 - a princípio sim |
| 4. Opinião não declarada |
| 17 - <nda> |
| 21 - Nada a comentar |
| 23 - Nada a declarar |
| 26 - <nda> |

Tabela 16 – (KT2) Em relação a geração de conhecimento entre os times, na sua opinião, o backstage facilita esse processo? O que poderia ser melhorado ou acrescentado para que isso ocorra? (parte A)

| |
|--|
| 1. Sim, facilita a geração de conhecimento através da documentação gerada |
| 1 - Sim, uma maneira mais rápida de documentar. |
| 3 - Os eventos publicados, recebidos, quais exchanges está usando. |
| 4 - Acho que o backstage já é promissor para a geração de conhecimento entre os times já que podemos ter uma visão ampla de cada um dos produtos ali concentrados. |
| 5 - Sim, facilita, pois centraliza informações importantes. |
| 6 - Sim. Sem sugestão. |
| 8 - A documentação de todos os produtos em um único lugar vai facilitar muito o entendimento dos processos de cada produto, fora que vai ser uma mão na roda, quando precisamos integrar os produtos. |
| 9 - Concordo. |
| 10 - Acredito que facilita. Acho que deve ser mais divulgado. |
| 12 - Acredito que quanto mais padrões seguirmos na documentação mais fácil para pessoas externas ao produto encontrarem o que procuram, nesse caso o backstage faz um trabalho mais coerente do que o TDN. |
| 14 - acredito que sim. |
| 4. Dúvidas em relação à sua real funcionalidade para o que é proposto |
| 16 - Não sei se seria a melhor ferramenta, mas pode auxiliar. |
| 19 - Não vejo esse tipo de oportunidade dentro do backstage. Hoje ele faz o que é esperado. Acredito que a oportunidade maior está na disseminação do mesmo. |
| 5. Sem opinião definida |
| 17 - <nada> |
| 26 - <nada> |

Tabela 17 – (KT2) Em relação a geração de conhecimento entre os times, na sua opinião, o backstage facilita esse processo? O que poderia ser melhorado ou acrescentado para que isso ocorra? (parte B)

| |
|--|
| 15 - Acredito que é essencial estabelecer processos que permitam a participação de todos na geração de conhecimento. Isso não se restringe apenas ao desenvolvimento de produtos e ferramentas existentes, mas também à criação de conteúdos que promovam a evolução desses produtos e possam beneficiar outros desenvolvedores. |
| 18 - Sim facilita o processo porque todo o conhecimento fica documentado e centralizado em um lugar não dependendo de uma só pessoa. |
| 20 - Facilita, pois é um lugar onde as informações estão organizadas. Uma possibilidade de definir um dashboard com cursos ou materiais que são obrigatórios para cada nível de cargo ou posição. |
| 21 - Sim, facilita. |
| 22 - Entendo que sempre que houver um local único com as informações necessárias, isso irá favorecer a democratização do conhecimento. |
| 23 - Sim facilita e cancela as ilhas de conhecimento, sobre as melhorias eu não tenho o que falar. |
| 25 - Sim. |
| 2. Facilita, porém há a necessidade de mais treinamento |
| 2 - Acredito que em termos de ferramentas estamos bem servidos, o que falta é o engajamento justamente para que possamos ir melhorando ela. Óbvio que só chegaremos a configuração ideal conforme a gente vai usando, e, algumas coisa que pensamos no início como boas para se mostrar, assim como outras ideias que ficaram de fora podem entrar para compor a visão compartilhada na ferramenta. |
| 7 - Facilita sim. Organizar um mini workshop/reunião quando algo maior for adicionado ao backstage para garantir que todos os membros das equipes saibam como utilizar. |
| 24 - Facilita o processo certamente. Teria que ter uma validação para atualizar os documentos ao entregar as issues para garantir que seja sempre atualizado. |
| 3. Não agrega valor para a geração de conhecimento |
| 11 - eu acredito que não agrega muito na geração de conhecimento, pois se trata de um lugar onde posso consultar o que cada time usa, se quiser me aprofundar a respeito de algo não sei se seria o caso de encontrar no backstage, mas de uma forma geral, para buscar coisas referentes a estrutura, padrões e afins é uma ferramenta muito boa. |
| 13 - Infelizmente o Backstage não possui uma solução integrada para criação/edição de documentos. Embora a proposta do Backstage seja utilizar mais o DocAsCode, poderia ser utilizado o https://decapcms.org/ para prover uma edição integrada de documentos. Além disso, o backstage não suporta comentários e subscrição (watch) nas documentações e entidades, o que poderia facilitar bastante a comunicação. |

Tabela 18 – (KT3) O backstage pode facilitar o processo de onboarding de novos integrantes ao time, com links úteis e informações essenciais a respeito do produto? (Sonar, Repositório, API's, etc) (parte A)

| |
|--|
| 3 - Com certeza, é um único local para se buscar o conhecimento. |
| 4 - entendo que sim. |
| 6 - Acredito fortemente que sim. |
| 7 - Facilita bastante. |
| 10 - Concordo. |
| 11 - Sim. |
| 12 - Sim, o backstage iria facilitar muito o onboarding de novos integrantes. |
| 14 - Com certeza. O backstage deveria ser a home de todo desenvolvedor. Aba pinada no browser! |
| 15 - Acredito que sim. |
| 17 - Concordo. |
| 22 - Queria ter visto por primeiro o backstage durante o meu processo de onboarding. |
| 2. Sim, por centralizar o conhecimento |
| 5 - Com certeza, por ser um centralizador. |
| 8 - Sim, os novos integrantes conseguem ter autonomia ao encontrar todas as informações importantes no backstage, sem precisar abrir vários links para acessar o que precisam. |
| 9 - Totalmente, acredito que a maior dificuldade dos novos e até mesmo antigos integrantes é localizar os artefatos utilizados no dia a dia. Tanto ferramentas, quanto links úteis e até mesmo o processo "padrão" de desenvolvimento utilizado pelo time etc. |
| 13 - Sim, as informações centralizadas evitam que os novos membros precisem se preocupar em lembrar onde buscar conhecimento. |
| 16 - Sim, quando entrei na T***s, tive que descobrir muitas coisas por conta própria, pois as informações não estavam centralizadas em um único local. Ter acesso centralizado a conhecimentos e informações será incrível, não apenas para o onboarding, mas também para o dia a dia dos desenvolvedores. |
| 19 - Pode pois centraliza a maioria das informações em um só lugar. |
| 20 - Demais. Todas as infos centralizadas seria um mundo perfeito. |
| 23 - Sim. Estando tudo documentado em um único local facilita bastante |
| 24 - Sim, agrupando essas informações fica mais fácil localizar e compartilhar. |
| 3. Sim, porém precisa ser gerado uma trilha de aprendizado |
| 21 - Pode, através de uma jornada de cursos e materiais definidos segundo a posição do colaborador. |
| 4. Sem opinião definida |
| 18, 25, 26 - <nada> |

Tabela 19 – (KT3) O backstage pode facilitar o processo de onboarding de novos integrantes ao time, com links úteis e informações essenciais a respeito do produto? (Sonar, Repositório, API's, etc) (parte B)

| |
|--|
| 1. Sim, realmente facilita o onboarding |
| 1 - Sim. |
| 2 - Sim, uma ótima ferramenta para onboard. |

Tabela 20 – (KT4) Ao observar o Backstage como um centralizador de conhecimento organizacional, você acredita que ele se mostra eficiente? Nesse sentido, quais são as informações que você considera como úteis para a centralização do conhecimento organizacional que pode ocorrer entre a arquitetura e entre equipes?(parte A)

| |
|--|
| 1. Se mostra como uma ferramenta eficiente |
| 5 - Sim, muito eficiente. |
| 6 - Sim. |
| 9 - Concordo. |
| 14 - as informação que já estão planejadas para terem é um bom começo, agora é preciso colocar em uso para ver se vai funcionar no dia a dia. |
| 16 - Se conseguir centralizar todas as informações necessárias em um único lugar seria ótimo. |
| 22 - Eu acredito muito nisso. |
| 1.1. APIs e Swagger são importantes |
| 1 - Sim, as APIs (swagger). |
| 1.2. Diagrama de classe e componente |
| 2 - Mapa de relacionamento (“Diagrama de classe”) pra mim é um dos pontos mais legal pois mostra de maneira visual e simplificada as ligações entre os componentes. |
| 1.3. APIs, documentação, estrutura da arquitetura, repositórios e monitoramento |
| 4 - Sim. Documentação, API's, repositórios, monitoradores. |
| 7 - Sim. É interessante ter disponível os padrões de codificação e exemplos de uso, repositórios de código, anotações/atas de reunião e histórico de versões das ferramentas de software. |
| 8 - Sim, acho toda informação relacionadas aos produtos e processo de desenvolvimento útil, exemplo: Acordos de arquitetura (event storming), stack de desenvolvimento back e front, padrões de arquitetura exemplo: hexagonal aplicando conceitos de DDD, patterns utilizados atualmente no produto em questão, comunicação entre os serviços exemplo: eventos (CloudAMQP). E toda parte do fluxo DEVOPs qual metodologia o time utiliza (agil Kanbam) as cadencias que temos no time (diaria, refinamento, reabastecimento, review e retrospectiva) etc. |
| 10 - Documentação de projetos, padrões, convenções de formas de desenvolvimento. |
| 12 - Sim, entre arquitetura e equipes alguns temas poderiam ser centralizados, tais como definições de stack de trabalho(IDE, plugins, Frameworks), decisões estruturais dos projetos, boas práticas, documentação a respeito de produtos mantidos pela arquitetura(por exemplo as extensões e SDK da Google). |

Tabela 21 – (KT4) Ao observar o Backstage como um centralizador de conhecimento organizacional, você acredita que ele se mostra eficiente? Nesse sentido, quais são as informações que você considera como úteis para a centralização do conhecimento organizacional que pode ocorrer entre a arquitetura e entre equipes? (parte B)

| |
|--|
| 13 - A ferramenta é eficiente e possui a abrangência suficiente para atender os desafios da gestão de conhecimentos com completude. Porém, para exponenciar seu funcionamento, seu uso deve estar alinhado a um processo bem desenhado. Principalmente seria útil se, além das informações da arquitetura, estivessem centralizadas informações da infraestrutura e plataforma t***sa**s lá! |
| 15 - Informações sobre migrações realizadas nos produtos, informações sobre tomadas de decisões de ferramentas e processos, criação de padrões e diretrizes bem definidas entre os times para que cada equipe tome decisões corretas de como prosseguir com um desenvolvimento ou processo específico sem que cada um faça de uma forma. |
| 18 - Sim é eficiente, as informações mais necessárias seriam as relativas à arquitetura e ferramentas usadas, e também algumas informações sobre processos e regras de negócio que facilitariam a compreensão de determinadas funcionalidades do sistema. |
| 19 - Tudo que for definido pela arquitetura e conversado com os devs pode entrar no backstage, desde que seja bem organizado. O intuito é guardar as informações para que possamos consultar em momentos de dúvidas. |
| 23 - Sim. As informações seriam acordos do time, configs de IDE e ambiente de projeto. |
| 24 - São muitos pontos para documentar e isso é facilitado por ser totalmente centralizado, porém os desafios normais de documentação seguem. Como documentos realmente úteis e atualizados. |
| 25 - Informações dos repositórios e status do sistema. |
| 2. Não sei responder |
| 3, 17, 21, 26 - Não sei ainda. |
| 3. Não vejo como eficiente sem testar melhor |
| 11 - Em como está hoje não vejo que ele é completamente eficiente, porém precisaria analisar implementações futuras, mas claramente é uma ferramenta que pode se tornar eficiente sim. |
| 20 - Preciso de mais tempo de uso para chegar a uma conclusão. |

Tabela 22 – (KT5) O que você mudaria dentro do Backstage e/ou na apresentação dos artefatos disponibilizados em cada produto? Cite os pontos positivos, negativos e limitações que você percebeu na utilização do backstage. Sugestões são sempre bem vindas! (parte A)

| |
|---|
| 1. Desconsideradas |
| 4 - Ainda processando essa informação. |
| 7 - Sem sugestão. |
| 11 - Conheço muito pouco do backstage. |
| 16 - Ainda estou avaliando para tomar alguns conclusões. |
| 17 - Não tenho sugestão no momento. |
| 18 - <em branco> |
| 19 - Nada |
| 24 - No momento não possuo muita opinião sobre isso. |
| 2. Opiniões gerais |
| 5 - A única coisa que não me agrada muito no backstage é a nível de layout, acho que [é] um layout mais simplório |
| 10 - Muito top |
| 21 - Integração como extensão de browser |
| 1 - Um pouco trabalhoso de manter |
| 6 - Não mudaria nada, só tentaria alimentar cada vez mais a ferramenta. |
| 9 - No momento nada, ele entrega bem o que se propõe, talvez com o uso contínuo da ferramenta a gente possa sugerir melhorias, mas pelo que foi demonstrado o backstage vai ser uma ótima ferramenta. |
| 3. Centralizador de informações |
| 8 - Acho legal a possibilidade de poder personalizar a tela inicial com os widgets e também a facilidade de encontrar o contato de cada membro das equipes de acordo com a área da pessoa. |
| 15 - Ponto positivo é ele ser um centralizador de informações. |
| 20 - O site como um todo é muito bom. Contém várias informações em um só lugar e já adianta muita coisa do dia-a-dia. A princípio, acho que traria como oportunidade a centralização de pessoas-time-APIs. O intuito é ajudar em momentos de estudos de problemas que podem acontecer, facilitando encontrar o vínculo entre pessoas, times e APIs. Ex: Determinado problema pode vir do time X. Vou olhar a API deles e chamar o Pedro* para conversarmos sobre (* Apenas para fins de exemplo.) |
| 22 - Ponto positivo: centralização das informações, ponto negativo: interface poderia ser mais bonita. |
| 23 - Entendo que centralizar a informação sobre os projetos simplifica a adaptação de quem está iniciando, evita a redundância de estratégias de acesso a documentação do produto, e simplifica consultas sempre que forem necessárias. |
| 25 - Gostei bastante da forma de apresentar as informações. Ainda achei um pouco difícil de encontrar dados mas acredito que seja pela falta de costume. |

Tabela 23 – (KT5) O que você mudaria dentro do Backstage e/ou na apresentação dos artefatos disponibilizados em cada produto? Cite os pontos positivos, negativos e limitações que você percebeu na utilização do backstage. Sugestões são sempre bem vindas! (parte C)

| |
|--|
| 4. Funcionalidades já existentes |
| 2 - Automatizar a atualização de conteúdos como swagger, async api, etc |
| 12 - Acredito que seria interessante ter uma aba direita para buscar as equipes ou membros. |
| 13 - Em um mundo ideal seria possível realizar uma integração com o T**, endo em vista que a T***S criou esse espaço e deprecia-lo totalmente levaria a perda de muitos conteúdos importantes. |
| 14 - A home poderia já trazer as 'infos' dos componentes que estou relacionado, sobre meu time, mensagens do time de arquitetura, notificações de atualização de SDKs dos componentes que mantenho. Como uma dashboard |
| 5. Liberação de acesso aos usuários |
| 3 - O permissionamento precisa ser melhorado. |
| 6. Foco no monitoramento da aplicação |
| 26 - Criação de uma sessão para monitoramento da aplicação, com mais detalhes, alertas, dashboards etc |