



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CAMPUS DE QUIXADÁ**  
**CURSO DE GRADUAÇÃO EM ENGENHARIA DE SOFTWARE**

**EWALDO DE CARVALHO BEZERRA JUNIOR**

**GUIA PARA AUXILIAR O GERENCIAMENTO DE DÉBITOS TÉCNICOS EM  
PROJETOS DE SOFTWARE**

**QUIXADÁ**

**2023**

EWALDO DE CARVALHO BEZERRA JUNIOR

GUIA PARA AUXILIAR O GERENCIAMENTO DE DÉBITOS TÉCNICOS EM PROJETOS  
DE SOFTWARE

Trabalho de Conclusão de Curso apresentado ao  
Curso de Graduação em Engenharia de Software  
do Campus de Quixadá da Universidade Federal  
do Ceará, como requisito parcial à obtenção do  
grau de bacharel em Engenharia de Software.

Orientadora: Prof<sup>a</sup>. Ma. Antonia Diana  
Braga Nogueira

QUIXADÁ

2023

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Sistema de Bibliotecas  
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

B469g Bezerra Junior, Ewaldo de Carvalho.  
Guia para auxiliar o gerenciamento de débitos técnicos em projetos de software / Ewaldo de Carvalho  
Bezerra Junior. – 2023.  
98 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá,  
Curso de Engenharia de Software, Quixadá, 2023.  
Orientação: Profa. Ma. Antonia Diana Braga Nogueira.

1. Dívida tecnológica. 2. Projetos de software. 3. Engenharia de software. I. Título.

CDD 005.1

---

EWALDO DE CARVALHO BEZERRA JUNIOR

GUIA PARA AUXILIAR O GERENCIAMENTO DE DÉBITOS TÉCNICOS EM PROJETOS  
DE SOFTWARE

Trabalho de Conclusão de Curso apresentado ao  
Curso de Graduação em Engenharia de Software  
do Campus de Quixadá da Universidade Federal  
do Ceará, como requisito parcial à obtenção do  
grau de bacharel em Engenharia de Software.

Aprovada em: \_\_\_\_/\_\_\_\_/\_\_\_\_.

BANCA EXAMINADORA

---

Prof<sup>ª</sup>. Ma. Antonia Diana Braga  
Nogueira (Orientadora)  
Universidade Federal do Ceará (UFC)

---

Prof<sup>ª</sup>. Dra. Paulyne Matthews Jucá  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Jeferson Kenedy Vieira Morais  
Universidade Federal do Ceará (UFC)

Dedico este trabalho primeiramente a Deus, o maior responsável por me fazer chegar até aqui. À minha mãe, pelo amor incondicional e apoio constante. E a mim mesmo, por sempre acreditar no meu potencial e ter persistido dia após dia.

## AGRADECIMENTOS

A Deus pelo dom da existência e por ter me dado forças para concluir este curso e Nossa Senhora por todas as graças concebidas na minha vida.

À minha família, em especial a minha mãe Nadir, por sempre acreditar e ter me incentivado nos estudos.

À Prof<sup>a</sup>. Ma. Diana Braga, minha orientadora neste trabalho, minha primeira professora da graduação e que me identifiquei desde o primeiro momento, agradeço por todo o suporte e incentivo dado no desenvolvimento deste trabalho. Um verdadeiro exemplo de mãe, professora e orientadora.

A todos os professores do Campus de Quixadá, especialmente os professores participantes da banca examinadora, Prof<sup>a</sup>. Dra. Paulyne Jucá e Prof. Dr. Jeferson Kenedy, e os professores que ministraram disciplinas que fiz. Todos vocês são verdadeiros mestres que possuem o poder de transformar a sociedade através da educação.

Aos profissionais participantes da pesquisa, através das respostas no questionário, agradeço o tempo concedido e as contribuições.

Aos amigos que fiz durante o curso, dividimos momentos memoráveis de muito aprendizado, desafios e descontração que fizeram a experiência de morar em Quixadá bem melhor. Também aos colegas da turma de Engenharia de Software 2020.1 que contribuíram direta e indiretamente para o meu aprendizado. Aos colegas do INOVE que compartilharam comigo conhecimentos e vivências de inovação e empreendedorismo. Por fim, aos colegas bolsistas do estágio nos Projetos Dell, foram meses de crescimento profissional e desejo muito sucesso a todos.

Ao Doutorando em Engenharia Elétrica, Ednardo Moreira Rodrigues, e seu assistente, Alan Batista de Oliveira, aluno de graduação em Engenharia Elétrica, pela adequação do *template* utilizado neste trabalho para que o mesmo ficasse de acordo com as normas da biblioteca da Universidade Federal do Ceará (UFC).

“A humildade é o primeiro degrau para a sabedoria.”

(São Tomás de Aquino)

## RESUMO

Débitos técnicos referem-se às consequências de escolhas técnicas que, a curto prazo, proporcionam vantagens, como aceleração no desenvolvimento de software, mas acarretam custos e desafios futuros. Desse modo, a gestão eficiente de débitos técnicos em projetos de *software* é um desafio premente, demandando abordagens estratégicas e práticas específicas. Este estudo propõe um guia de boas práticas voltado para o gerenciamento destas dívidas técnicas, visando aprimorar a qualidade e o sucesso de projetos de desenvolvimento de *software*. A pesquisa compreende em um levantamento abrangente de informações sobre débitos técnicos, a consolidação de práticas de gerenciamento existentes e a estruturação do guia para auxiliar no gerenciamento de débitos técnicos. A metodologia adotada envolveu uma pesquisa quantitativa e qualitativa com profissionais de tecnologia, fundamentando-se em 75 respostas válidas. Os resultados da pesquisa revelaram a importância do gerenciamento de débitos técnicos, apontando para a necessidade de estabelecer processos e práticas recorrentes nos projetos. Além disso, proporcionaram subsídios para a elaboração do guia de boas práticas, consolidando as descobertas da pesquisa e recursos científicos pertinentes. O guia, resultado concreto deste trabalho, é um artefato estruturado com o propósito de formalizar atividades de gerenciamento de débitos técnicos e disseminar conhecimento sobre essas questões.

**Palavras-chave:** débitos técnicos; gerenciamento de débitos técnicos; gerência de projetos de *software*; qualidade de *software*; manutenção de *software*; boas práticas em engenharia de *software*.

## ABSTRACT

Technical debts refer to the consequences of technical choices that, in the short term, provide advantages such as accelerating software development but entail costs and future challenges. Thus, the efficient management of technical debts in software projects is a pressing challenge, requiring strategic approaches and specific practices. This study proposes a guide of best practices aimed at managing these technical debts, aiming to enhance the quality and success of software development projects. The research involves a comprehensive survey of information on technical debts, the consolidation of existing management practices, and the structuring of the guide to assist in the management of technical debts. The adopted methodology included a quantitative and qualitative survey with technology professionals, based on 75 valid responses. The research results revealed the importance of managing technical debts, emphasizing the need to establish recurring processes and practices in projects. Furthermore, they provided insights for the development of the guide of best practices, consolidating the findings of the research and relevant scientific resources. The guide, a tangible outcome of this work, is a structured artifact with the purpose of formalizing activities related to the management of technical debts and disseminating knowledge on these issues.

**Keywords:** technical debts; management of technical debts; software project management; software quality; software maintenance; best practices in software engineering.

## LISTA DE FIGURAS

Figura 1 – Débitos técnicos e atributos de qualidade . . . . .	16
Figura 2 – Relação entre as etapas de gerenciamento de débitos técnicos, custo, garantia e controle da qualidade de <i>software</i> . . . . .	17
Figura 3 – Processo de manutenção de <i>software</i> . . . . .	19
Figura 4 – <i>Framework</i> de gerenciamento de débitos técnicos . . . . .	23
Figura 5 – Práticas ágeis no desenvolvimento de <i>software</i> . . . . .	25
Figura 6 – Processos ágeis no desenvolvimento de <i>software</i> . . . . .	25
Figura 7 – Práticas ágeis no desenvolvimento de <i>software</i> . . . . .	27
Figura 8 – Visão geral dos resultados das atividades de gerenciamento de débitos técnicos	28
Figura 9 – Etapas do método de pesquisa . . . . .	31
Figura 10 – Grau de formação dos participantes . . . . .	34
Figura 11 – Faixa etária dos participantes . . . . .	36
Figura 12 – Tempo de experiência dos participantes . . . . .	36
Figura 13 – Tamanho da equipe dos participantes . . . . .	37
Figura 14 – Frequência de identificação dos débitos técnicos pelos participantes . . . . .	38
Figura 15 – Frequência de atividades de gerenciamento de débitos técnicos nos projetos dos participantes . . . . .	43

## LISTA DE TABELAS

Tabela 1 – Análise entre os trabalhos relacionados e este trabalho . . . . .	30
Tabela 2 – Curso de formação dos participantes . . . . .	35
Tabela 3 – Comunicação dos débitos técnicos nas equipes dos participantes . . . . .	39
Tabela 4 – Causas que geram os débitos técnicos de acordo com os participantes . . . . .	39
Tabela 5 – Atributos de qualidade afetados pelos débitos técnicos de acordo com os participantes . . . . .	40
Tabela 6 – Artefatos e práticas para organizar e rastrear débitos técnicos nas equipes dos participantes . . . . .	42
Tabela 7 – Benefícios de implementar boas práticas de gerenciamento de débitos técni- cos de acordo com os participantes . . . . .	46

## SUMÁRIO

1	INTRODUÇÃO . . . . .	13
2	FUNDAMENTAÇÃO TEÓRICA . . . . .	15
2.1	Débitos técnicos . . . . .	15
2.2	Qualidade de <i>software</i> . . . . .	16
2.3	Manutenção de <i>software</i> . . . . .	18
2.4	Gerência de projetos . . . . .	19
2.5	Boas práticas em Engenharia de <i>Software</i> . . . . .	20
3	TRABALHOS RELACIONADOS . . . . .	22
3.1	How do software development teams manage technical debt? – An empirical study . . . . .	22
3.2	Technical debt and agile software development practices and processes: An industry practitioner survey . . . . .	24
3.3	A framework to aid in decision making for technical debt management . . . . .	26
3.4	Perceptions of Technical Debt and its Management Activities - A Survey of <i>Software</i> Practitioners . . . . .	27
3.5	Análise Comparativa . . . . .	29
4	METODOLOGIA . . . . .	31
4.1	Projetar o questionário . . . . .	31
4.2	Validar a estrutura do questionário . . . . .	31
4.3	Aplicar o questionário . . . . .	32
4.4	Consolidar os dados coletados . . . . .	32
4.5	Analisar as informações consolidadas . . . . .	32
4.6	Elaborar um guia conforme as análises realizadas . . . . .	33
5	RESULTADOS . . . . .	34
5.1	Perfil do participante . . . . .	34
5.2	Perfil do ambiente de trabalho do participante . . . . .	37
5.3	Entendimento sobre débitos técnicos . . . . .	38
5.4	Entendimento sobre o gerenciamento de débitos técnicos . . . . .	42
5.5	Entendimento sobre a ausência de gerenciamento de débitos técnicos . . . . .	45
6	GUIA . . . . .	49

6.1	<b>Introdução</b> . . . . .	49
6.2	<b>O que são débitos técnicos?</b> . . . . .	49
6.3	<b>Por que gerenciar débitos técnicos?</b> . . . . .	49
6.4	<b>Como identificar os débitos técnicos?</b> . . . . .	50
6.5	<b>Como monitorar os débitos técnicos?</b> . . . . .	51
6.6	<b>Como priorizar os débitos técnicos?</b> . . . . .	52
6.7	<b>Como resolver os débitos técnicos?</b> . . . . .	53
6.8	<b>Como prevenir os débitos técnicos?</b> . . . . .	53
6.9	<b>Conclusão</b> . . . . .	55
7	<b>SÍNTESE</b> . . . . .	57
8	<b>CONSIDERAÇÕES FINAIS</b> . . . . .	59
8.1	<b>Limitações da pesquisa</b> . . . . .	59
8.2	<b>Principais contribuições</b> . . . . .	59
8.3	<b>Trabalhos futuros</b> . . . . .	60
	<b>REFERÊNCIAS</b> . . . . .	61
	<b>APÊNDICES</b> . . . . .	64
	<b>APÊNDICE A– TERMO DE CONSENTIMENTO LIVRE E ESCLA- RECIDO (TCLE)</b> . . . . .	64
	<b>APÊNDICE B– QUESTIONÁRIO PILOTO</b> . . . . .	66
	<b>APÊNDICE C– QUESTIONÁRIO FINAL</b> . . . . .	73
	<b>APÊNDICE D– JUSTIFICATIVA PARA AS QUESTÕES DO QUES- TIONÁRIO FINAL</b> . . . . .	83
	<b>APÊNDICE E– VERSÃO RESUMIDA DO GUIA</b> . . . . .	92

## 1 INTRODUÇÃO

Os débitos técnicos podem ser descritos como todos os atalhos que economizam dinheiro ou aceleram o progresso no presente, mas com o risco de potencialmente custar dinheiro ou desacelerar o progresso do projeto no futuro, prejudicando a qualidade, arquitetura e *design* do *software*. O gerenciamento adequado desses débitos pode ser complicado, já que o débito técnico pode vir de várias causas, tendo efeitos difíceis de prever e geralmente envolve uma aposta sobre o que acontecerá no futuro (ALLMAN, 2012).

Allman (2012) faz um comparativo com a dívida financeira para melhorar o entendimento sobre o termo. Na dívida financeira, a pessoa que faz o empréstimo deseja que ele seja reembolsado e geralmente tem que ser pago com juros. Desse modo, o cliente paga mais dinheiro do que recebeu. Caso o cliente não possa pagar de volta, haverá um custo muito alto. Diante disso, levanta-se três conceitos: reembolso, juros e alto custo. Quando caracterizado no contexto de projetos de *software*, priorizar metas a curto prazo, mudar o escopo do projeto constantemente e trabalhar com prazos apertados garantem o acúmulo de débitos técnicos, tornando a complexidade de resolução destes problemas cada vez maior, gerando um custo muito alto para resolver.

Brown *et al.* (2010) destaca que o gerenciamento eficaz dos débitos técnicos em projetos é percebido como crítico para alcançar e manter a qualidade do software. Se não forem gerenciados, surgirão problemas significativos a longo prazo, como aumento dos custos de manutenção. Assim, há a necessidade de monitorar as alterações no escopo e o impacto das alterações nos projetos de software.

Dahl (2021) descreve que os níveis de volatilidade, incerteza, complexidade e ambiguidade são altos demais no que se refere ao gerenciamento de projetos. Nesta perspectiva, quando associado tais conceitos com débitos técnicos, percebe-se que as constantes mudanças e a agilidade das informações favorecem a dificuldade em rastrear os atrasos causados por débitos técnicos. Portanto, entender, comunicar e gerenciar os débitos técnicos podem fazer uma enorme diferença no sucesso de um sistema a curto e longo prazo (ALLMAN, 2012).

Já Kruchten *et al.* (2012) ressalta que os débitos técnicos não são intrinsecamente ruins, podendo ser uma ferramenta útil para alcançar metas a curto prazo, permitindo que os projetos cumpram prazos apertados ou entreguem funcionalidades de forma mais rápida. No entanto, a falta de conscientização e estratégias para lidar com esses débitos pode levar a problemas graves a longo prazo. Portanto, é fundamental encontrar um equilíbrio entre a

necessidade de inovação e a manutenção da qualidade do *software*, estabelecendo processos eficazes de gerenciamento de débitos técnicos.

Neste contexto, o presente trabalho explora de maneira abrangente o gerenciamento de débitos técnicos em projetos de *software*, oferecendo um guia de boas práticas para abordar esse desafio, apresentando um conjunto de diretrizes que pode auxiliar as equipes de desenvolvimento de *software* na identificação, monitoramento, priorização, resolução e prevenção dos débitos técnicos. Espera-se que este guia contribua significativamente para a qualidade e o sucesso de projetos de *software*, ajudando as organizações a gerenciar de forma mais eficiente as complexidades inerentes ao desenvolvimento de *software* em um ambiente de constante volatilidade.

Assim, o objetivo geral deste trabalho é auxiliar no controle e na resolução de débitos técnicos. A iniciativa busca, assim, contribuir significativamente para a qualidade e o sucesso de projetos de desenvolvimento de *software*, fornecendo uma visão abrangente, por meio do guia, que oriente equipes na minimização dos impactos adversos causados por débitos técnicos.

Para atingir o objetivo central de desenvolver um guia para o gerenciamento de débitos técnicos em projetos de *software*, esta pesquisa delinea objetivos específicos interconectados. Primeiramente, visa-se realizar um levantamento abrangente de informações sobre débitos técnicos, buscando compreender seus diversos impactos, contextos e perspectivas. Em seguida, o trabalho propõe consolidar práticas de gerenciamento existentes, analisando experiências e estratégias adotadas em projetos de *software* para lidar de maneira eficiente com esses desafios específicos. Por fim, pretende-se estruturar um guia que, de forma clara e acessível, ofereça orientações precisas e aplicáveis, concentrando-se na minimização do impacto dos débitos técnicos nos projetos de desenvolvimento de *software*. Esses objetivos específicos formam uma base sólida para a elaboração do guia proposto, alinhando-se ao propósito maior de aprimorar a qualidade de projetos de *software*.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo pretende apresentar a base teórica utilizada na elaboração deste trabalho.

### 2.1 Débitos técnicos

Cunningham (2009) descreve o conceito de débito técnico comparando que alguns problemas com código são como dívidas financeiras. O autor usa o termo para justificar a refatoração em um projeto como algo extremamente necessário e, para isso, usou uma analogia financeira. A metáfora pode ser resumida da seguinte forma: ao pegar atalhos e entregar código que não é adequado para a programação de um determinado projeto naquele momento, a equipe de desenvolvimento incorre em débito técnico, diminuindo assim a produtividade da equipe. Essa perda de produtividade seria os juros desse débito técnico.

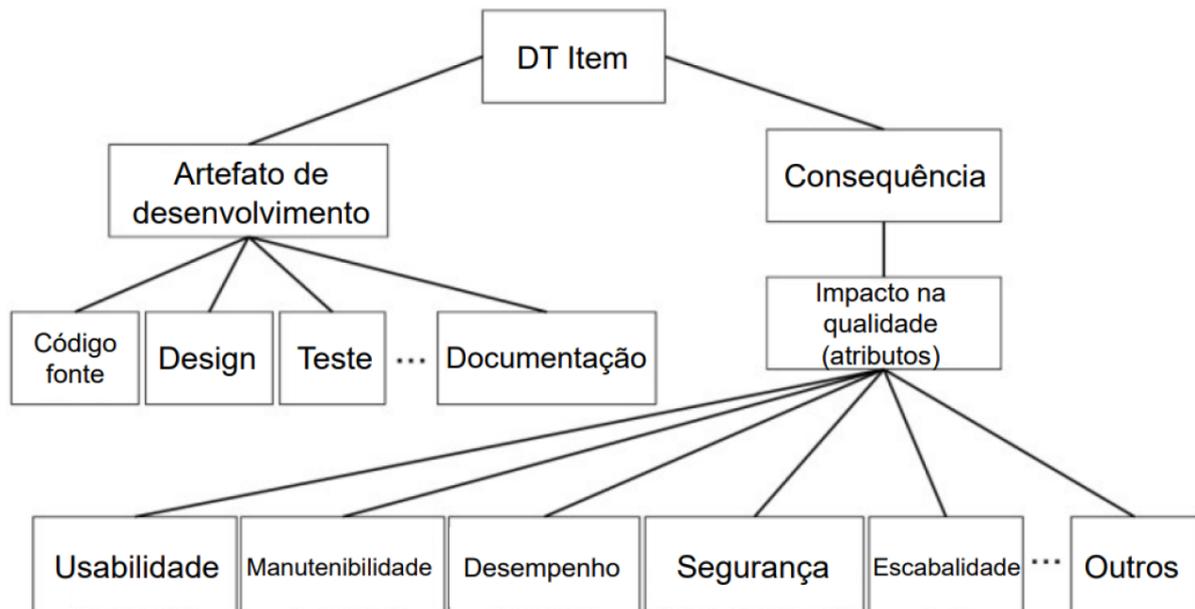
Fairbanks (2020) complementa os débitos técnicos no contexto de codificação, atrelando os conceitos descritos por Cunningham (2009). É descrito questões arquiteturais e padronizações que afetam o desempenho do *software*, utilizando o termo “código hacky”, que são códigos sem estruturação que prejudicam o desenvolvimento iterativo do *software*. Além disso, enfatiza que os débitos técnicos acontecem até nas melhores circunstâncias e utilizar ferramentas nem sempre são as melhores técnicas de mitigá-los, afirmando a necessidade de gerenciar os débitos técnicos ao invés de apenas dizer que os problemas vão surgir e vão piorar se forem ignorados.

Buschmann (2011) apresenta questões sobre pagar ou não o débito técnico. O autor analisa que os débitos técnicos são pautadas em prazos difíceis para cumprir e metas de custo rígidas, de modo que fortalecem a consolidação de um *design* rápido e mal feito e estilo de codificação resultante de fatores sem robustez. Buschmann (2011) afirma que, quando os débitos técnicos são intencionais, os líderes do projeto conseguem tomar mais controle nas decisões, trabalhando de forma proativa e não reativa. Ademais, o autor destaca a importância de lidar com soluções que estejam no nível de habilidades da equipe de desenvolvimento, sendo papel dos arquitetos o dever de esboçar e avaliar alternativas que garantam custo-benefício.

Stochel *et al.* (2020) apontam quatro estratégias para auxiliar nas práticas relacionadas às formas de lidar com os débitos técnicos, que são: manter a compreensão metafórica do que são os débitos técnicos; esclarecimento explícito, introdução de definições e taxonomias; uso de termos para reduzir ambiguidades e definir o campo de investigação do débito técnico; e seleção

explícita da área de interesse que o débito técnico está relacionado. A partir destas estratégias, é possível compreender a necessidade de manter as informações consolidadas acerca das áreas que os débitos técnicos se relacionam a fim de facilitar o processo de rastreabilidade. Na Figura 1 os autores apresentam a correlação entre os débitos técnicos e os atributos de qualidade que os relacionam.

Figura 1 – Débitos técnicos e atributos de qualidade



Fonte: Stochel *et al.* (2020) adaptado pelo autor.

## 2.2 Qualidade de *software*

Valente (2020) descreve que qualidade é um objetivo recorrente em produtos de engenharia. Fabricantes de automóveis, celulares, computadores, empresas de construção civil, etc., todos almejam e dizem que possuem produtos de qualidade, e dentro do contexto de *software*, este conceito não seria diferente. O autor também menciona que, para garantir qualidade de *software*, diversas estratégias podem ser usadas, como métricas, que podem ser eficientes para acompanhar o desenvolvimento de um produto de *software*, incluindo métricas de código-fonte e métricas de processo, por exemplo.

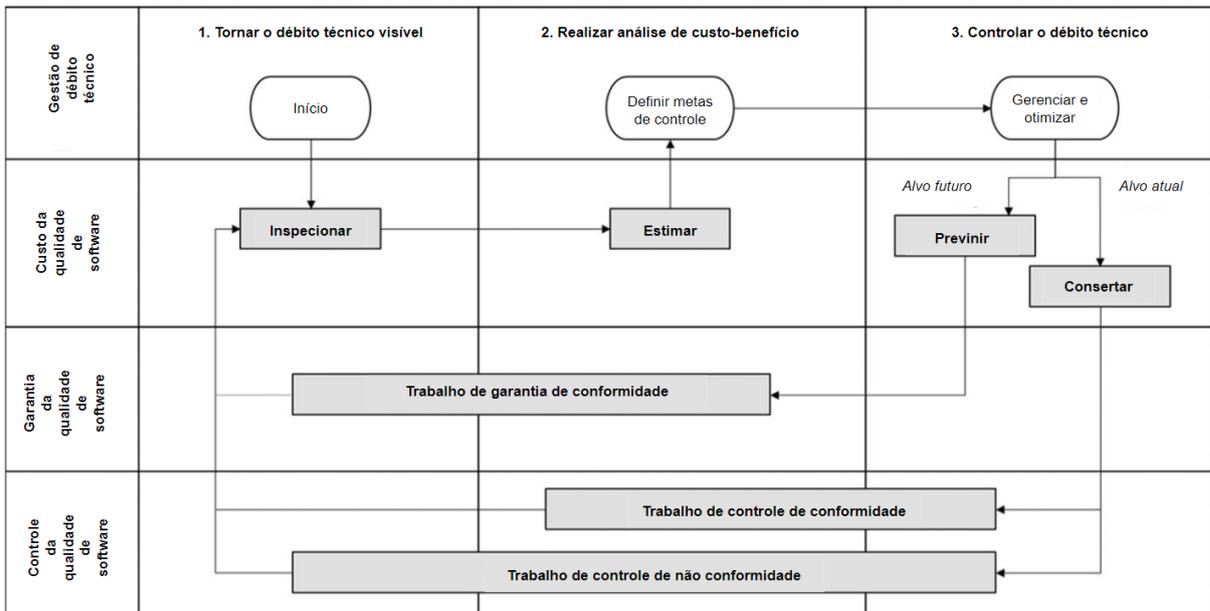
Consoante aos conceitos de qualidade, Valente (2020) destaca as duas dimensões de qualidade de *software*: qualidade externa e qualidade interna. A qualidade externa considera fatores que podem ser aferidos sem analisar o código e a qualidade interna considera propriedades e características relacionadas com a implementação de um sistema. Desse modo, a fim de fortalecer

os conceitos e métricas dessas qualidades, em NBR ISO/IEC 9126-1 (2003), é tratado sobre a qualidade de produtos em Engenharia de *Software*, apontando atributos de qualidade referentes a funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade. NBR ISO/IEC 9126-1 (2003) destaca que, tais atributos permitem visualizar mais facilmente esta qualidade e, por consequência, definir uma forma de medir a qualidade, possibilitando uma avaliação mais objetiva e uniforme.

A partir dos conceitos de qualidade de *software* citados, Ramasubbu e Kemerer (2019) descrevem que práticas estabelecidas de garantia e controle de qualidade podem ser utilizadas para associar efetivamente defeitos de *software* a decisões específicas de implantação tomadas pelos desenvolvedores, fazendo com que estas associações de débitos técnicos fiquem visíveis para a equipe, de modo a facilitar a quantificação do que é principal destas dívidas.

Além disso, Ramasubbu e Kemerer (2019) também expõem que os débitos técnicos são fatores críticos que podem afetar negativamente a qualidade do *software* e aumentar os custos de manutenção a longo prazo. Na Figura 2, é ilustrado um fluxo de práticas relacionadas aos débitos técnicos e processos de qualidade de *software*, destacando atividades de inspeção, estimativas, prevenções e consertos a fim de garantir a conformidade com a qualidade de *software*.

Figura 2 – Relação entre as etapas de gerenciamento de débitos técnicos, custo, garantia e controle da qualidade de *software*



Fonte: Ramasubbu e Kemerer (2019) adaptado pelo autor.

### 2.3 Manutenção de *software*

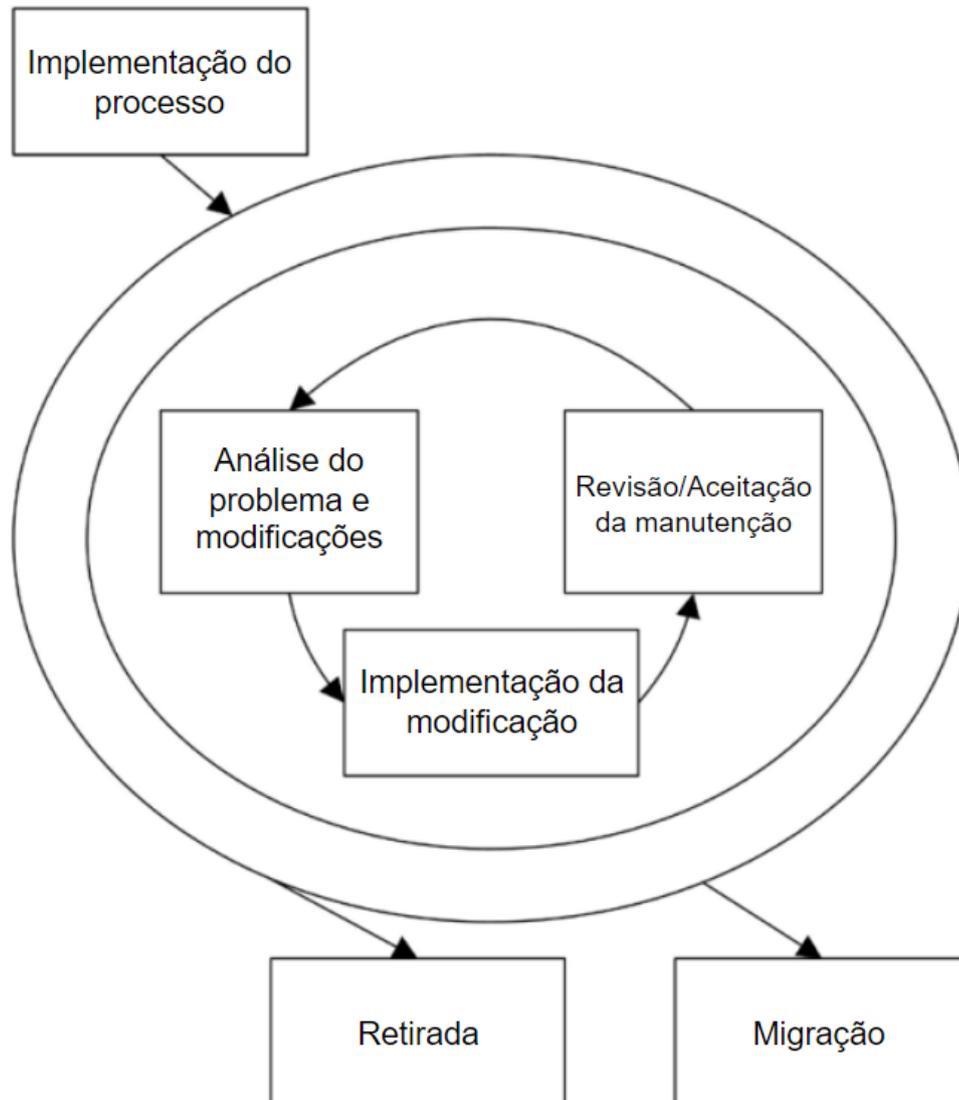
Sbrocco e Macedo (2012) descrevem que o *software* inevitavelmente vai sofrer modificações, e que uma modificação acontece quando: erros são encontrados; quando o *software* precisa ser adaptado para acomodar mudanças no seu ambiente externo; ou quando o cliente deseja melhorias funcionais ou de desempenho. Contextualizando, Valente (2020) destaca os tipos de manutenção que podem ser realizadas: corretiva, preventiva, adaptativa, refatoração e evolutiva.

- Corretiva: pretende corrigir *bugs* reportados por usuários ou outros desenvolvedores.
- Preventiva: visa corrigir *bugs* latentes no código, que ainda não causaram falhas junto aos usuários do sistema.
- Adaptativa: procura adaptar um sistema a uma mudança em seu ambiente, incluindo tecnologia, legislação, regras de integração com outros sistemas ou demandas de novos clientes.
- Refatoração: são modificações realizadas em um *software* mantendo seu comportamento e focado exclusivamente na melhoria do código ou projeto.
- Evolutiva: realizada para incluir uma nova funcionalidade ou introduzir aperfeiçoamentos relevantes em funcionalidades existentes.

Bourque e Fairley (2014) destacam que a manutenção de *software* é parte integrante do ciclo de vida do *software*. No entanto, não recebe o mesmo grau de atenção que as outras fases receberam. Historicamente, o autor afirma que o desenvolvimento de *software* teve um perfil muito mais elevado do que a manutenção de *software* na maioria das organizações, contudo, a necessidade de manter os *softwares* em operação pelo maior tempo possível tem dado respaldo às práticas de manutenção. Na Figura 3, é ilustrado como Bourque e Fairley (2014) descreve o processo de manutenção de *software*.

Desse modo, Almeida (2019) explicita que a manutenção de *software* envolve a identificação e correção dos débitos técnicos, garantindo que o *software* permaneça funcional, seguro e sustentável ao longo do tempo. Além disso, o autor destaca que a priorização dos débitos técnicos é um aspecto crítico da manutenção de *software*, pois envolve a identificação e a resolução de problemas e imperfeições no *software* a fim de reconhecer a importância de considerar fatores de negócios e métricas ao tomar decisões.

Figura 3 – Processo de manutenção de *software*



Fonte: Bourque e Fairley (2014) adaptado pelo autor.

## 2.4 Gerência de projetos

O PMI (2017) descreve que projeto é um esforço temporário empreendido para criar um produto, serviço ou resultado único. Já quanto ao gerenciamento de projetos, Carvalho (2018) afirma que inclui práticas de planejar, organizar, supervisionar e controlar todos os aspectos de um projeto, por meio de um processo contínuo para alcançar seus objetivos, no qual se integram com ferramentas, habilidades e conhecimentos que visem atender as necessidades e expectativas das partes interessadas. Além disso, afirma a singularidade de cada projeto, de modo que, são marcados por incertezas e complexidades únicas e inovadoras.

Carvalho (2018) complementa sobre a sobrevivência de uma empresa, requer de seus executivos ações rápidas, coerentes e não desperdiçar nenhum tipo de oportunidade. Para manter

uma empresa ativa, é importante muito conhecimento, esforço e escolha de um conjunto de práticas gerenciais que conduzam a empresa com eficiência e eficácia a fim de atingir resultados relevantes.

Nesse sentido, Carvalho (2018) ressalta o papel das equipes no âmbito dos projetos, pois devem ser proativas em buscar resultados por meio de uma orientação voltada para tarefas e atividades. Nesse aspecto, destaca a necessidade de planejar ações que visam o desenvolvimento e comprometimento com a agenda do projeto, o orçamento, a gestão dos riscos e com a qualidade. Para que isso aconteça, a equipe precisa desempenhar um espírito de colaboração e dedicação com os requisitos de gerenciamento do projeto, que referem-se às condições, capacidades, recursos e restrições necessários para planejar, executar, monitorar e concluir um projeto com sucesso.

A fim de entender os conceitos de gerenciamento de projetos com os débitos técnicos, é relevante investigar os motivos cruciais que fomentam a consolidação dos débitos técnicos. Em Oliveira *et al.* (2015) é explicado que os débitos técnicos se apoiam em três principais pilares, que são: identificação dos débitos técnicos, medição dos débitos técnicos e monitoramento dos débitos técnicos. De modo que, quando é atrelado estes pilares com os processos do gerenciamento de projetos, que são às atividades inter-relacionadas e coordenadas realizadas para atingir os objetivos de um projeto, torna-se pertinente encontrar uma concordância de atividades para rastrear e controlar as atividades nos projetos. Dessa forma, mostra-se fundamental associar os conceitos da gestão de projetos para construir uma estrutura sólida de gerenciamento de débitos técnicos.

## **2.5 Boas práticas em Engenharia de *Software***

Talluri e Haddad (2014) define boas práticas em Engenharia de *Software* como um conjunto de práticas de gerenciamento ou técnicas que demonstram consistentemente melhorar um ou mais dos seguintes aspectos em um projeto de *software* no que se refere a um ambiente de agilidade.

Talluri e Haddad (2014) elenca principais melhores práticas para o contexto de projetos de *software*, que são elas:

1. Equipes auto-organizadas: melhoria da competência, colaboração, motivação e confiança dos membros.
2. Reuniões de revisão: aprimoramento do desempenho, reconhecimento de sucessos e

identificação de melhorias.

3. Desenvolvimento orientado a testes (TDD): garantia de código bem projetado e testável.
4. Integração contínua: minimização de problemas de integração e melhoria da qualidade do *software*.
5. Documentação de requisitos: identificação de problemas, facilitação de acordos entre clientes e equipe de desenvolvimento e rastreamento de mudanças.
6. Envolvimento do cliente: melhoria na visão do cliente sobre o projeto e identificação de mudanças nos requisitos.

Marijan e Sen (2020) aborda o desafio de alinhar a pesquisa em engenharia de *software* com a prática na indústria e a importância de superar as barreiras que muitas vezes existem entre essas duas comunidades. Além disso, o autor também destaca a necessidade de tornar a pesquisa mais relevante e útil para a indústria, adaptando-a às necessidades práticas e ao contexto do setor, enfatizando a importância de uma comunicação eficaz e da colaboração entre as partes interessadas na pesquisa em engenharia de *software* e prática industrial para superar esses desafios e alcançar sinergias mais fortes.

Por fim, Frezza (2016) enfatiza a importância de não apenas ensinar habilidades técnicas em engenharia de *software*, mas também garantir a valorização e aplicação dessas habilidades em contextos reais, essa ênfase na valorização é vista como uma boa prática para melhorar a eficácia dos processos. Dessa forma, ao atrelar com os estudos de Marijan e Sen (2020) percebe-se que trazer as percepções acadêmicas teóricas para o contexto profissional é fundamental para construção de uma base sólida de práticas eficientes, o autor destaca principalmente sobre tornar as pesquisas acessíveis à indústria, a fim de facilitar a integração e a colaboração contínua.

### 3 TRABALHOS RELACIONADOS

Nas seções a seguir, serão apresentados quatro trabalhos relacionados com este trabalho e na última seção será apresentado um quadro com as principais características desses trabalhos de forma que será possível visualizar melhor quais são as diferenças e os pontos em comum com este trabalho.

#### 3.1 How do software development teams manage technical debt? – An empirical study

Em Yli-Huumo *et al.* (2016) é apresentado como os débitos técnicos são gerenciados. Uma grande organização de desenvolvimento de *software* é investigada para reunir evidências empíricas relacionadas ao conceito de gerenciamento de débito técnico. Além disso, os autores retratam que gerenciá-los é como uma parte essencial do desenvolvimento sustentável do *software*, apresentando que as empresas precisam encontrar abordagens corretas para lidar com os débitos técnicos para que o *software* seja mantido no futuro.

Este trabalho identifica a falta de evidências empíricas sobre gerenciamento de débitos técnicos em ambientes de desenvolvimento de *software* reais. Desse modo, faz uso de uma abordagem de estudo de caso para coletar dados qualitativos, entender como funciona na realidade e compreender como as equipes conduzem o gerenciamento de débitos técnicos.

Para entender o gerenciamento de débitos técnicos, Yli-Huumo *et al.* (2016) estudaram oito equipes (vinte e cinco pessoas ao todo) de desenvolvimento de *software* em uma organização fornecedora de soluções tecnológicas. De modo que, foi definido como um estudo de caso exploratório interpretativo por meio de entrevistas semiestruturadas, visando compreender a interpretação dos participantes em seu contexto. O procedimento metodológico foi dividido em cinco etapas: projetar o estudo de caso; preparar a coleta de dados; coletar as evidências; analisar os dados coletados; gerar relatórios. Além disso, a estruturação dos procedimentos ficou separada em duas rodadas porque os autores fizeram em anos diferentes e o projeto da entrevista precisou ser modificado.

Como resultados da pesquisa, os autores as conclusões dos oito casos de uso, categorizando as principais atividades referentes aos débitos técnicos após o processo de análise de dados: reembolso de débitos técnicos, identificação de débitos técnicos, medição de débitos técnicos, monitoramento de débitos técnicos, priorização de débitos técnicos, comunicação de débitos técnicos, prevenção de débitos técnicos e documentação de débitos técnicos.

Após analisar as categorias nos casos de uso, os autores construíram um framework com base em níveis de maturidade para entender a habilidade da equipe de desenvolvimento nas atividades de gerenciamento de débitos técnicos. De modo que, é possível visualizar o perfil do quão maduras são as equipes com os débitos técnicos, a fim de observar que existem vários desafios de gerenciamento de débitos técnicos e que as equipes de desenvolvimento de *software* precisam entender e reconhecer tais práticas. Na Figura 4, é apresentado o framework construído por Yli-Huumo *et al.* (2016), estruturado nas colunas com os níveis e atividades de gerenciamento de débitos técnicos, já nas linhas com o contexto das atividades presentes em cada nível.

Figura 4 – *Framework* de gerenciamento de débitos técnicos

DT: débito técnico	Atividades de gerenciamento de débito técnico							
Níveis de gerenciamento de DT	Reembolso do DT	Prevenção do DT	Documentação/re apresentação de DT	Identificação do DT	Medição do DT	Monitoramento do DT	Comunicação do DT	Priorização do DT
Organizado (Nível 3)	Reembolso contínuo com porcentagem mensal atribuída das tarefas de desenvolvimento.	Práticas de prevenção obrigatórias utilizadas pela equipe. Prática contínua durante o desenvolvimento.	A documentação é uma prática obrigatória no desenvolvimento. Os problemas são documentados em uma lista de pendências de DT separada.	Identificação contínua realizada manualmente e/ou com ferramentas durante o desenvolvimento.	Medição contínua durante o desenvolvimento. Análise de dados (vários dados usados (por exemplo, qualidade, desempenho)). Auxílio com ferramentas.	Monitoramento contínuo durante o desenvolvimento com vários dados (por exemplo, qualidade, desempenho). Ferramentas usadas para dar suporte.	Discussões/reuniões contínuas sobre questões de DT com todas as partes interessadas envolvidas.	Priorização continuamente durante o desenvolvimento. A priorização segue um método ou modelo específico.
Receptível (Nível 2)	Reembolso durante tarefas normais de desenvolvimento e tarefas de reembolso previamente identificadas. Reembolso realizado com base nas necessidades atuais.	Práticas opcionais de prevenção. Não é obrigatório usar, mas recomendado. Conduzido com base nas restrições de tempo atuais.	A documentação é uma prática opcional, mas recomendada. Problemas documentados em um backlog de desenvolvimento geral sem identificador de DT.	Identificação opcional durante o desenvolvimento normal. Conduzido com base nas restrições de tempo atuais.	A medição é uma prática opcional. Medição feita com dados simples (número de questões TD) do desenvolvimento, e os dados não necessariamente usados para outras atividades.	Monitoramento baseado em dados simples (número de problemas de DT). Realizado ocasionalmente.	Discussões/reuniões organizadas apenas com algumas partes interessadas.	Priorização baseada em palpites e estimativas aproximadas com base em experiências anteriores. Priorização feita de forma simples sem nenhum modelo específico.
Desorganizado (Nível 1)	Reembolso não realizado ou apenas quando não é mais possível evitar o problema.	Prevenção não atribuída como parte das práticas de desenvolvimento. Realizado apenas ocasionalmente.	Documentação não faz parte do desenvolvimento. Os problemas são deixados nas próprias mentes e notas dos desenvolvedores.	Práticas de identificação não atribuídas como parte do desenvolvimento. Conduzido apenas quando ocorrem problemas.	A medição não faz parte das práticas de desenvolvimento.	Monitoramento não faz parte das práticas de desenvolvimento.	DT não é um tópico em discussões/reuniões e muitas vezes é tratado apenas em mesas de café.	Priorização não realizada e decisões feitas sem raciocínio ou discussões.
Responsabilidade de pela atividade	Equipe de desenvolvimento e arquiteto(s) de software	Equipe de desenvolvimento e arquiteto(s) de software	Equipe de desenvolvimento e arquiteto(s) de software	Equipe de desenvolvimento e arquiteto(s) de software	Arquiteto(s) de software e líder da equipe	Arquiteto(s) de software e líder da equipe	Equipe de desenvolvimento, líder da equipe e arquiteto(s) de software	Arquiteto(s) de software e líder da equipe
Práticas / ferramentas pela atividade	Refatorar, redesenhar e reescrever	Padrões de codificação, revisões de código e definição de pronto	Backlog/lista de DT, prática de documentação, ferramenta de gerenciamento de projetos (JIRA, Wiki)	Reserva de tempo para inspeção manual do código. Utilização de ferramentas de análise de código (SonarQube, CheckStyle, FindBugs).	Dados de ferramentas de medição (SonarQube) e dados de ferramentas de gerenciamento de projetos (JIRA, Wiki).	Ferramentas de monitoramento (SonarQube). Ferramentas de gerenciamento de projetos (JIRA, Wiki).	Encontros específicos de DT, DT incluídos em tópicos de discussão.	Modelo de custo/benefício, classificação de emissão

Fonte: Yli-Huumo *et al.* (2016) adaptado pelo autor.

### 3.2 Technical debt and agile software development practices and processes: An industry practitioner survey

Em Holvitie *et al.* (2018) é apresentado a dinamicidade dos ambientes de desenvolvimento de *software*, propiciando o acúmulo de débitos técnicos. Mesmo que este fenômeno tenha aumentado no contexto acadêmico, os autores afirmam que o uso do conceito na indústria de *software* permanece não estudado. O conhecimento sobre os efeitos das práticas ou processos de desenvolvimento de *software* sobre os débitos técnicos é potencialmente importante para os profissionais por ser necessário tomar decisões mediante o cenário de agilidade usado nos projetos.

O objetivo deste trabalho é entender as percepções e experiências dos profissionais para mapear os efeitos do uso das metodologias ágeis no gerenciamento de débitos técnicos, e partir disso, é explorado o quanto os profissionais têm conhecimento sobre os débitos técnicos, como os débitos técnicos se manifestam nos processos de *software*, e os efeitos de práticas e processos comuns dos métodos ágeis nos débitos técnicos.

A fim de validar o cenário dos débitos técnicos na indústria de projetos de *software*, os autores fizeram um estudo exploratório, no qual, projetaram, testaram e executaram um questionário de pesquisa multinacional com 184 respostas de profissionais no Brasil, Finlândia e Nova Zelândia, o questionário continha 37 perguntas, 35 delas eram fechadas e as 2 abertas eram usadas para solicitar informações mais detalhadas. Holvitie *et al.* (2018) visam responder tais questões de pesquisa a partir deste questionário:

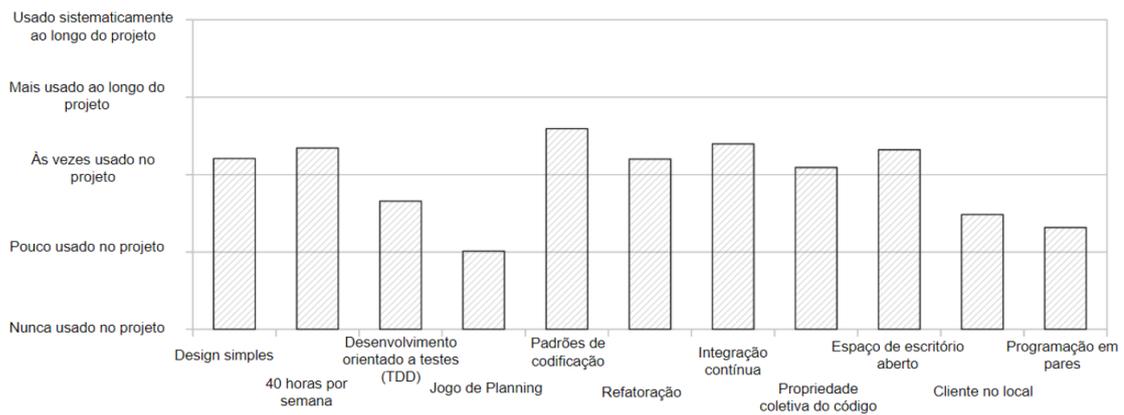
- a) Existem diferenças nas percepções das várias partes interessadas sobre o débito técnico e os antecedentes das partes interessadas afetam tais percepções?
- b) Quais são os impulsionadores percebidos do débito técnico e quais fases de desenvolvimento de *software* o débito técnico afeta?
- c) As práticas e processos de métodos ágeis são percebidos como tendo um efeito sobre o débito técnico ou seu gerenciamento e quais práticas são consideradas tendo o efeito mais significativo?

Como resultados da pesquisa, após a análise dos dados coletados no questionário, os profissionais da indústria demonstraram alta conformidade (ou podem se relacionar) com as definições dadas para débitos técnicos e os principais cargos deles são nas áreas de desenvolvimento, design e testes. Os autores também perceberam que os débitos técnicos são em sua grande maioria considerados pertinentes ao serem relatados em situações referentes à infraestrutura do

desenvolvimento, condução do desenvolvimento e em situações imprevistas. Outros resultados também são expostos e retratam que as principais causas concretas de débitos técnicos são de arquitetura, estrutura, testes e documentação inadequados.

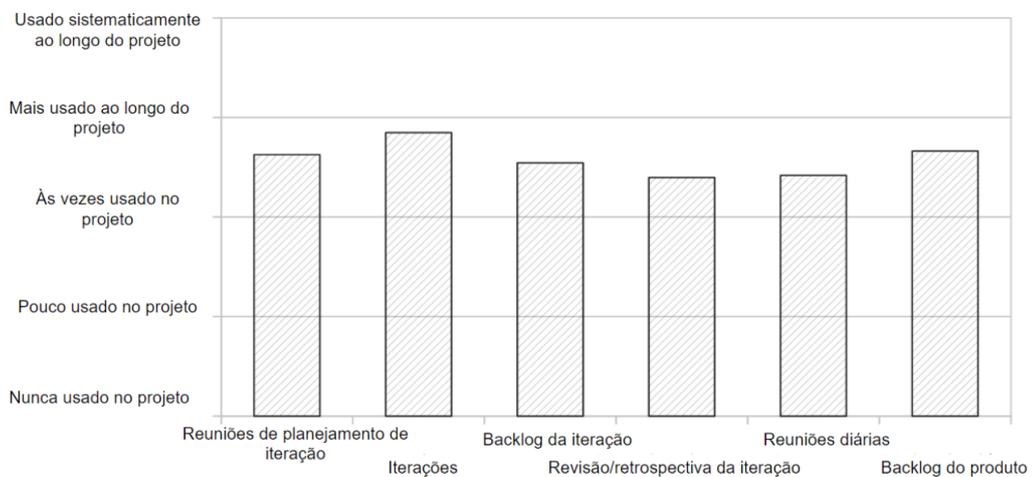
Para concluir as percepções ágeis dentro dos projetos, os autores puderam analisar que as práticas ágeis de padronização de código e integração contínua são as mais utilizadas para gerenciar os débitos técnicos, como ilustrado no gráfico da Figura 5. Além disso, nos processos ágeis, o papel das iterações (*sprints*) e estruturação do *backlog* (atividades realizadas nas iterações) são usadas nos projetos para enfatizar a priorização e as melhorias incrementais no desenvolvimento, estes resultados estão ilustrados na Figura 6. E no contexto de débitos técnicos, tais práticas e processos auxiliam no monitoramento e refinamento contínuo do desenvolvimento do *software*.

Figura 5 – Práticas ágeis no desenvolvimento de *software*



Fonte: Holvitie *et al.* (2018) adaptado pelo autor.

Figura 6 – Processos ágeis no desenvolvimento de *software*



Fonte: Holvitie *et al.* (2018) adaptado pelo autor.

### 3.3 A framework to aid in decision making for technical debt management

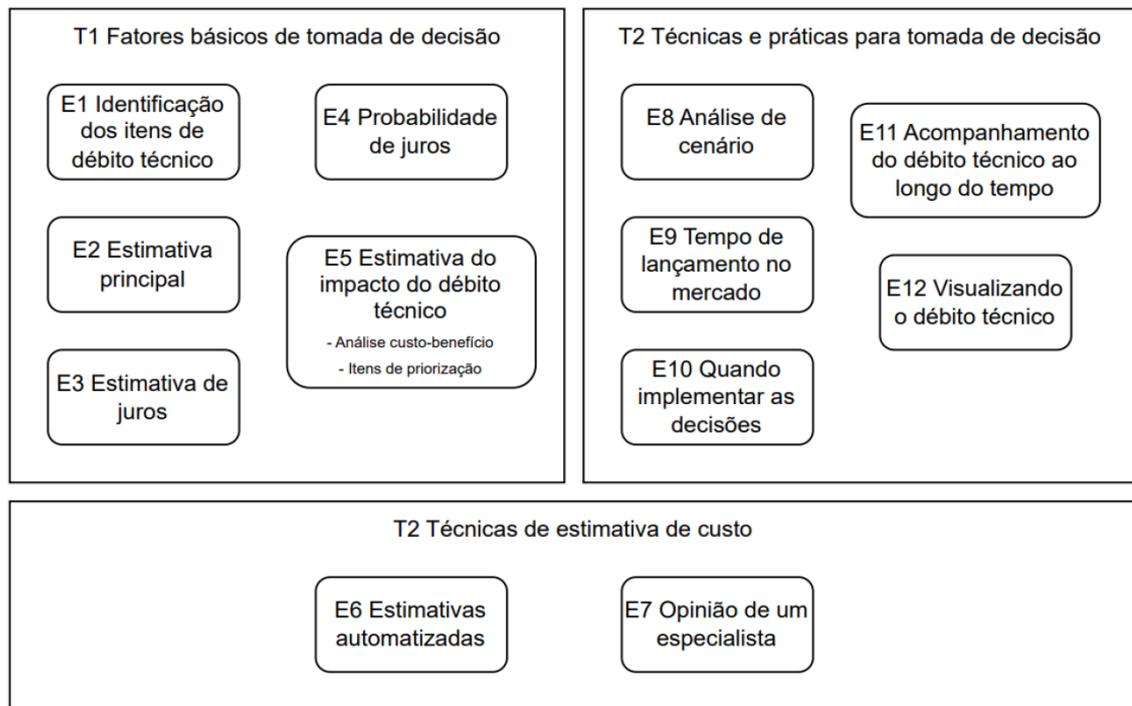
Em Fernández-Sánchez *et al.* (2015) é descrito o impacto negativo dos débitos técnicos descontrolados no desenvolvimento de *software*, causando efeitos na moral dos desenvolvedores, na velocidade da equipe e na qualidade dos produtos. O objetivo do trabalho é abordar os primeiros passos na definição de uma estrutura que auxilie na tomada de decisão para o gerenciamento de débitos técnicos. E por meio disso, são levantadas duas questões de pesquisa:

- a) Quais elementos são considerados para gerenciar os débitos técnicos na tomada de decisão em um projeto de *software*?
- b) Quais desses elementos são considerados no ponto de vista das várias partes interessadas?

Para identificar estes elementos presentes nas questões de pesquisa, os autores realizaram um mapeamento sistemático a fim de fornecer uma visão ampla para identificar os elementos que devem ser considerados para gerenciar eficientemente os débitos técnicos. O número total de trabalhos obtidos foi de 795 e, após um processo de seleção, 51 trabalhos foram definidos. Como resultados da pesquisa, Fernández-Sánchez *et al.* (2015) mostram os elementos identificados e seus respectivos tipos. Foram levantados três tipos: fatores básicos de tomada de decisão, técnicas de estimativa de custo, e práticas e técnicas para tomada de decisão.

Na Figura 7, é apresentado quadrantes que mostram os elementos identificados, e, no trabalho, é descrito especificamente cada um destes elementos. Por exemplo, o elemento “identificação de itens de débito técnico” descreve que se faz necessário incluir e estabelecer uma lista de más práticas que criam débitos técnicos, bem como analisar os tipos e fontes dos débitos técnicos. Para entender o ponto de vista das partes interessadas, foram definidas três categorias: engenharia (*software*), gestão técnica e gestão organizacional de negócios. O ponto de vista preponderante foi o da engenharia, que envolve times de *design*, desenvolvimento, testes e arquitetura, pois a maioria dos trabalhos mapeados eram focados nos fatores básicos de tomada de decisão.

Figura 7 – Práticas ágeis no desenvolvimento de *software*



Fonte: Fernández-Sánchez *et al.* (2015) adaptado pelo autor.

### 3.4 Perceptions of Technical Debt and its Management Activities - A Survey of Software Practitioners

Em Albuquerque *et al.* (2022) é apresentado uma análise abrangente das atividades de gerenciamento de débitos técnicos. O estudo desempenhado no trabalho foca em compreender as percepções dos profissionais de *software* em relação aos percalços que surgem a partir dos débitos técnicos, e a partir disso, foi desenvolvido uma pesquisa com funcionários de organizações multinacionais. Além disso, é mencionado os principais tipos de débitos técnicos que pertinentes no contexto do ciclo de vida do projeto: débitos técnicos em requisitos, documentações, arquitetura, *design*, código, teste e infraestrutura. Ademais, dentro das atividades de gerenciamento de débitos técnicos, os autores destacam as práticas de identificação, medição, priorização, prevenção, monitoramento, reembolso, documentação e comunicação como cruciais para evitar a ocorrência de débitos técnicos.

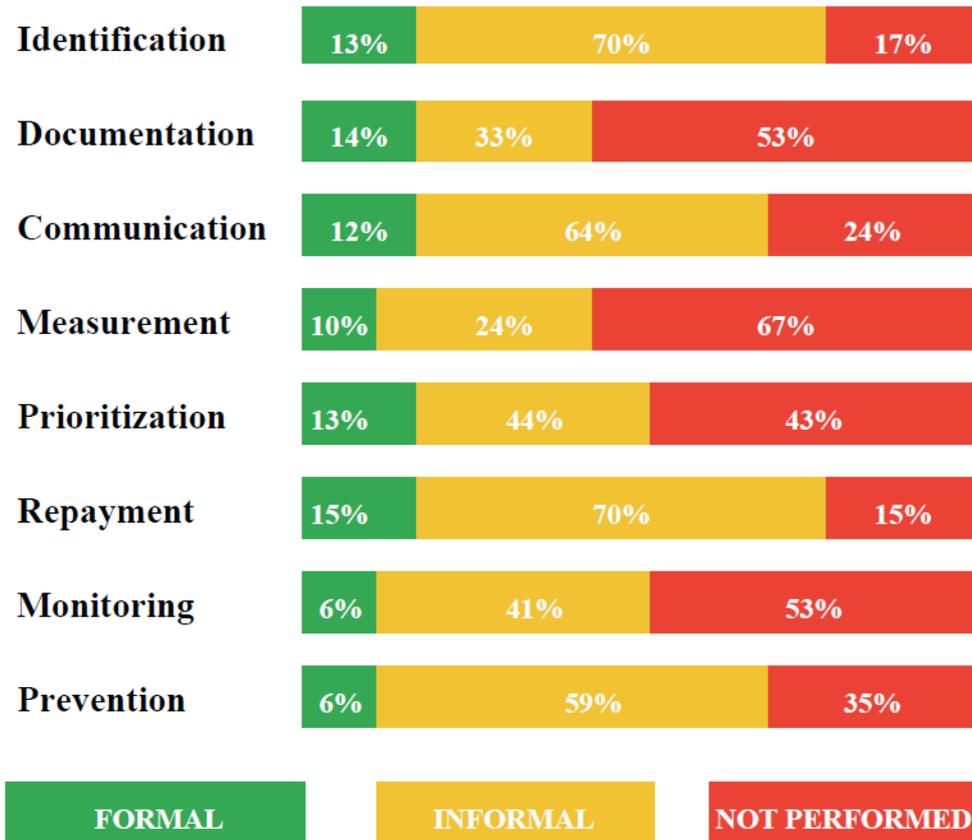
Nesse sentido, para auxiliar no desenvolvimento da pesquisa, os autores estipularam questões de pesquisa relacionadas ao consenso dos profissionais de *software* acerca dos aspectos gerais dos débitos técnicos, as atividades de gerenciamento de débitos técnicos relevantes nos projetos e as soluções adotadas para gerenciar estas atividades. Por se tratar de um *survey*, foram feitas etapas de construção do questionário, teste piloto, coleta de dados e análise destes dados. E

como resultados, conseguiram 120 respostas de cinco países diferentes (Brasil, Estados Unidos, Canadá, Suécia e Índia).

A partir da pesquisa, foi constatado que quase 80% dos participantes afirmaram ter uma compreensão sobre débitos técnicos e em relação à percepção de débitos técnicos, 72% dos participantes forneceram exemplos válidos afetados, como código, arquitetura, requisitos, testes e *design*. Um resultado relevante da pesquisa é que apenas 11% dos participantes adotam um processo formal para lidar com o gerenciamento de débitos técnicos.

Na Figura 8, é possível perceber que as atividades são de caráter mais informal ou nem chegam a ser realizadas em boa parte dos projetos. Os autores expõem que para atividades de identificação, a maioria dos profissionais faz manuais de inspeção, para documentação usam listas ou *backlogs*. Ademais, destacam que a comunicação flui por meio de reuniões, a medição acontece por rastreadores de problemas e para priorizar fazem trabalho com análises estáticas. E por fim, para reembolsar o débito técnico é feito em sua grande maioria a partir de refatorações, o monitoramento manual se destaca como uma prática contínua e as prevenções são principalmente baseadas em revisões de código.

Figura 8 – Visão geral dos resultados das atividades de gerenciamento de débitos técnicos



Fonte: Albuquerque *et al.* (2022)

### 3.5 Análise Comparativa

Esta seção apresenta uma análise comparativa entre os trabalhos relacionados que foram apresentados neste capítulo com o trabalho proposto. Na Tabela 1, está disposta uma comparação a fim de resumir a análise.

As semelhanças encontradas entre este trabalho e o trabalho de Yli-Huumo *et al.* (2016) foram que os autores visaram entender como os débitos técnicos são tratados em ambientes reais de desenvolvimento de *software* e a estruturação de categorias para entender melhor os débitos técnicos, se tratando da identificação, prevenção, priorização, entre outros. A principal diferença é a metodologia, os autores fizeram estudo de caso e a partir destes casos geraram um *framework* conforme os níveis de gerenciamento de débitos técnicos, enquanto no trabalho proposto é executado um formulário para a coleta de dados e a geração de um guia.

A principal semelhança encontrada entre este trabalho e o de Holvitie *et al.* (2018) foi a metodologia aplicada. Os autores fizeram um questionário para entender a influência dos débitos técnicos nos projetos de *software*, e as respostas coletadas foram relevantes para entender as causas concretas dos débitos técnicos. Já, como diferença, no presente trabalho não é tratado diretamente sobre processos e práticas ágeis como foi neste trabalho relacionado, pois esta não é a prioridade do estudo, apesar de que, os métodos ágeis atualmente estão bem presentes dos projetos de *software* e faz-se relevante entender como eles se correlacionam com os débitos técnicos. No trabalho proposto é abordado questões gerais sobre agilidade, no entanto, focado em entender a influência, meios de mitigação e priorização dos débitos técnicos.

A principal semelhança entre este trabalho e o trabalho de Fernández-Sánchez *et al.* (2015) foi a consolidação de uma estrutura que auxilie na tomada de decisões para gerenciar os débitos técnicos, baseada em elementos para estimar, identificar e mapear os débitos técnicos. A principal diferença é a divergência da metodologia. Neste trabalho relacionado foi realizado um mapeamento sistemático e validado tais perspectivas sobre o gerenciamento de débitos técnicos por meio da base científica, enquanto no trabalho proposto é realizado formulários para coleta de dados e a geração de um guia com base nas perspectivas do mercado de trabalho. No entanto, este trabalho faz-se fundamental para embasamento teórico acerca dos débitos técnicos, sendo usado como base de idealização na construção do presente trabalho.

A principal semelhança entre este trabalho e o trabalho de Albuquerque *et al.* (2022) é a metodologia utilizada, em ambos os trabalhos foram realizados um *survey* focado no contexto da indústria de desenvolvimento de *software*. Além disso, os trabalhos visam estudos sobre

o gerenciamento de débitos técnicos, focados em atividades fundamentais para rastrear estas questões. A principal diferença são as contribuições dos trabalhos, enquanto em Albuquerque *et al.* (2022) são construídas conclusões a partir dos resultados obtidos no questionário, neste presente trabalho é desenvolvido um artefato, o guia, a partir das percepções obtidas na pesquisa.

Na Tabela 1 encontra-se um resumo das principais características dos trabalhos relacionados citados neste capítulo e também os aspectos pertinentes a este trabalho.

Tabela 1 – Análise entre os trabalhos relacionados e este trabalho

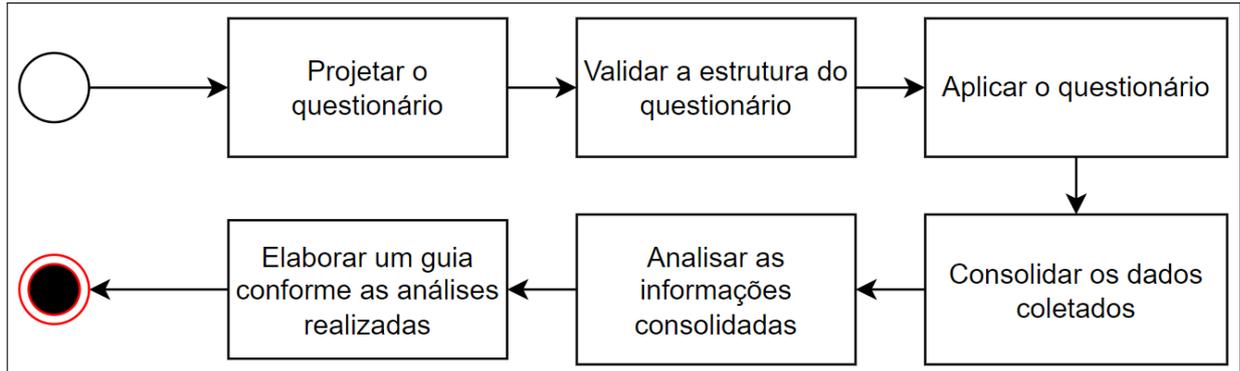
<b>Trabalho</b>	<b>Metodologia utilizada na pesquisa</b>	<b>Artefato gerado para profissionais em projetos de software</b>	<b>Analisa ambientes reais de projetos de software</b>	<b>Estrutura os débitos técnicos em atividades de gerenciamento</b>
(YLI-HUUMO <i>et al.</i> , 2016)	Estudo de caso	<i>Framework</i> baseado no nível de organização das equipes	Sim	Sim
(HOLVITIE <i>et al.</i> , 2018)	<i>Survey</i>	Não foi gerado	Sim	Não
(FERNÁNDEZ-SÁNCHEZ <i>et al.</i> , 2015)	Mapeamento sistemático	<i>Framework</i> baseado na tomada de decisão	Não	Sim
(ALBUQUERQUE <i>et al.</i> , 2022)	<i>Survey</i>	Não foi gerado	Sim	Sim
Este trabalho	<i>Survey</i>	Guia	Sim	Sim

Fonte: Elaborado pelo autor.

## 4 METODOLOGIA

Neste capítulo, são apresentados os passos necessários para a execução deste trabalho a fim de cumprir os objetivos estabelecidos. A Figura 9 ilustra as fases da metodologia, nas seções seguintes, cada fase do método é apresentada.

Figura 9 – Etapas do método de pesquisa



Fonte: Elaborado pelo autor

### 4.1 Projetar o questionário

A primeira atividade deste trabalho foi realizar a construção do questionário utilizado para identificar como os projetos de *software* lidam com os débitos técnicos e entender como comprometem no contexto dos projetos. A partir disso, esta primeira etapa definiu as perguntas que compuseram o questionário, das quais foram tanto quantitativas quanto qualitativas a respeito de débitos técnicos. Foi utilizada a plataforma Google Forms<sup>1</sup>, que possui uma estrutura bem consolidada de construção de formulários, além de gerar visualizações relevantes para análise. Ademais, para melhorar a captação dos dados, foi usado a Escala Likert em alguns questionamentos. A versão inicial do questionário está no Apêndice B.

### 4.2 Validar a estrutura do questionário

Nesta etapa, foi realizada uma validação do questionário, focada em testá-lo para evitar que seja aplicado com possíveis falhas, inconsistências nas perguntas e enviesamento de respostas. Desse modo, o questionário foi aplicado em um grupo de 5 participantes que fazem parte do perfil da pesquisa, a fim de que sejam feitas críticas construtivas capazes de

<sup>1</sup> Disponível em <https://docs.google.com/forms/u/0/>

refinar o formulário para diminuir ameaças à validade que possam ocorrer. Além disso, ajustes foram sugeridos a partir das análises da banca na defesa do Trabalho de Conclusão de Curso I e com pesquisas em outros trabalhos referenciais. Como resultados deste teste piloto, algumas perguntas abertas foram retiradas para evitar possível repetição em respostas e consolidação dessa subjetividade de perguntas qualitativas por meio da escala Likert a fim de objetificação.

### **4.3 Aplicar o questionário**

Após validado, o questionário foi aplicado, disponibilizado e divulgado para participantes que se encaixaram no perfil, participantes que trabalham na área de desenvolvimento e gerenciamento de projetos/produtos, por meio de redes sociais utilizadas por profissionais, como o LinkedIn<sup>2</sup>, grupos no Discord, e Telegram, além de ter sido encaminhado para acadêmicos da Universidade Federal do Ceará que atuam no mercado de trabalho na área. O questionário passou um 45 dias aberto, e isso possibilitou a coleta de 75 respostas válidas.

### **4.4 Consolidar os dados coletados**

Após aplicado, foi necessário consolidar os dados coletados. O Google Forms auxiliou no processo de consolidação ao gerar gráficos e planilhas capazes de integrar as informações e facilitar na visualização das respostas coletadas. Como o questionário era composto por perguntas quantitativas e qualitativas, em algumas perguntas abertas existiram respostas vazias, inconsistentes, e podem não fazer sentido com a proposta da pesquisa, então foi necessário fazer a extração destes dados e consolidar as informações mais relevantes.

### **4.5 Analisar as informações consolidadas**

Após consolidar, foi realizada uma análise dessas informações. Nesta etapa ocorreu uma análise quantitativa das respostas obtidas, verificando a suficiência dos dados coletados para a pesquisa, visualização estatística e gráfica dos resultados. Além disso, a utilização da escala Likert foi fundamental para avaliar algumas perspectivas. E, em relação às perguntas qualitativas, com base nos estudos de Leitão e Prates (2017), foi realizada uma categorização das respostas para visualizar as ocorrências de alguns assuntos destacados pelos participantes e garantir a visibilidade dos mesmos. Após as duas análises, foi realizada a triangulação das informações,

---

<sup>2</sup> Disponível em <https://www.linkedin.com/>

a fim de relacionar as respostas abertas e fechadas. O processo de análise foi importante para sintetizar os dados mais relevantes para a próxima etapa da pesquisa, feito o uso de planilhas e documentos para filtrar as informações que foram mais eficientes de compreender sobre como os débitos técnicos são tratados nos projetos de *software*, como lidam com essas questões e atividades de mitigação no *software*.

#### **4.6 Elaborar um guia conforme as análises realizadas**

Enquanto as informações coletadas iam sendo analisadas e constatadas as perspectivas dos débitos técnicos nos projetos de *software*, foi iniciada a concepção do guia baseado nestas constatações. Foram levadas em consideração as respostas dos participantes, principalmente por meio da triangularização entre as informações quantitativas e as categorias das qualitativas que mais se repetiram. Nesse sentido, o objetivo do guia é orientar no processo de identificação, monitoramento, priorização, resolução e prevenção dos débitos técnicos. O intuito do guia é tentar ser simples, assertivo e direcionado a profissionais em diferentes níveis, oferecendo suporte na tomada de decisões para garantir a rastreabilidade e previsibilidade dos débitos técnicos, minimizando riscos que possam impactar o progresso do projeto. Além disso, no intuito de disseminar os resultados obtidos na pesquisa, o presente guia será postado no LinkedIn, visando a propagação dos conhecimentos alcançados na pesquisa. No Apêndice E, está ilustrado esta versão do guia.

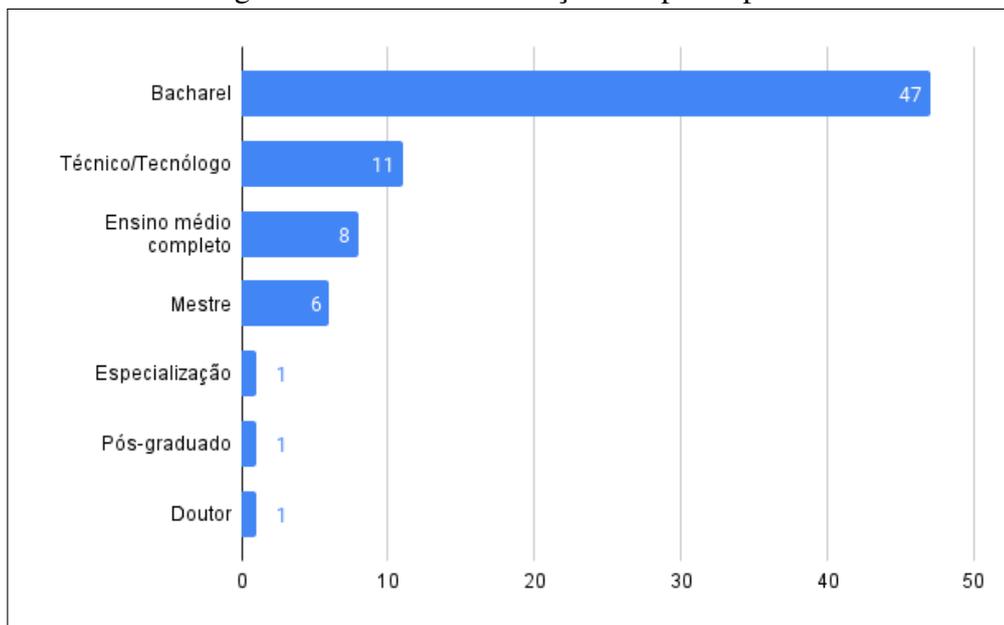
## 5 RESULTADOS

Este capítulo apresenta os resultados obtidos através do questionário enviado aos participantes dessa pesquisa. Os resultados foram organizados nas seguintes seções: perfil do participante, perfil do ambiente de trabalho do participante, e entendimento sobre débitos técnicos. Além disso, os participantes seguiram dois fluxos de distintos de perguntas: entendimento sobre o gerenciamento de débitos técnicos caso seja gerenciado no contexto do projeto do participante, e entendimento sobre a ausência de gerenciamento de débitos técnicos caso não seja. A consolidação dos resultados feitos no formulário foram armazenados em uma planilha no Google Planilhas<sup>1</sup> com as respostas extraídas dos 75 participantes válidos e a versão prévia dos gráficos antes de serem refinados e apresentados posteriormente nesta seção.

### 5.1 Perfil do participante

Sobre o grau de formação dos participantes, pode-se observar que 47 dos participantes são bacharéis, 11 são técnicos/tecnólogos, 8 possuem o ensino médio completo, 6 são mestres, um possui especialização, um é pós-graduado e um é doutor. A Figura 10 sintetiza esses dados.

Figura 10 – Grau de formação dos participantes



Fonte: Elaborado pelo autor.

Em relação ao curso de formação dos 75 participantes, pode-se observar que não há

<sup>1</sup> Disponível em <https://bit.ly/planilha-forms>

uma variedade muito discrepante fora do contexto de tecnologia, apesar de algumas ocorrências. Os mais citados foram Engenharia de *Software* (16), Ciência da Computação (15), Sistemas de Informação (14), Análise e Desenvolvimento de Sistemas (9), participantes que não possuem ou não estão fazendo algum curso (5), é possível inferir que os mesmos sinalizaram possuir o ensino médio completo e Engenharia da Computação (5). Já entre os menos citados, pode-se destacar Sistemas e Mídias Digitais (5), *Design Digital*, Técnico em Edificações, Técnico em Informática, Matemática Industrial, Engenharia de Produção e Administração, com apenas um participante cada curso. Na Tabela 2, estes dados estão consolidados.

Tabela 2 – Curso de formação dos participantes

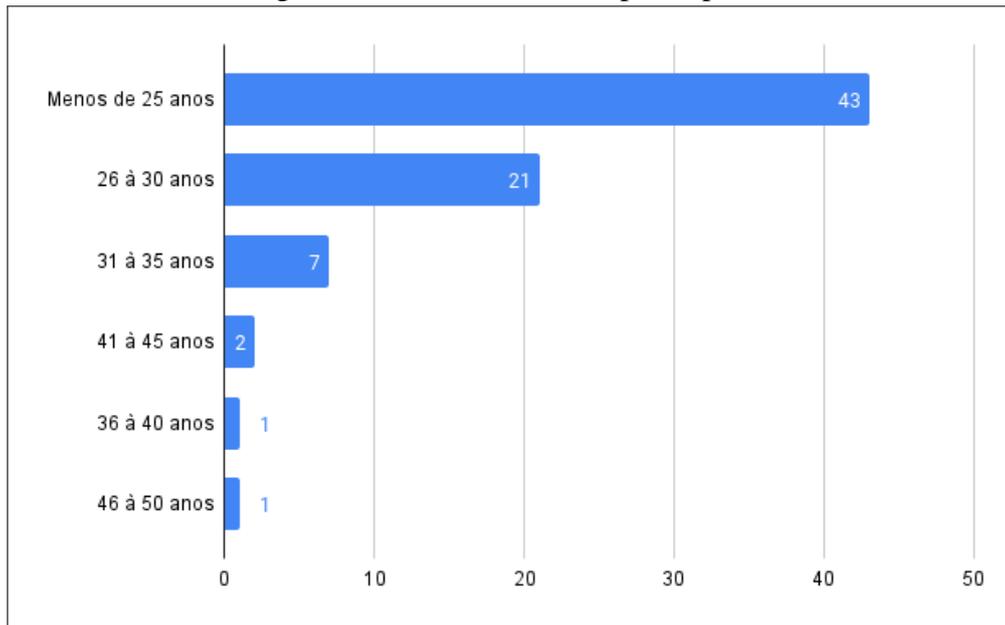
<b>Curso</b>	<b>Valor (percentual)</b>
Engenharia de <i>Software</i>	16 (21,3%)
Ciência da Computação	15 (20%)
Sistemas de Informação	14 (18,7%)
Análise e Desenvolvimento de Sistemas	9 (12%)
Não possui	5 (6,7%)
Engenharia da Computação	5 (6,7%)
Sistemas e Mídias Digitais	5 (6,7%)
<i>Design Digital</i>	1 (1,3%)
Técnico em Edificações	1 (1,3%)
Técnico em Informática	1 (1,3%)
Matemática Industrial	1 (1,3%)
Engenharia de Produção	1 (1,3%)
Administração	1 (1,3%)

Fonte: Elaborado pelo autor.

Sobre o cargo atual dos participantes, 63 deles trabalham com desenvolvimento de *software* e 12 trabalham no gerenciamento de projetos/produtos. Desse modo, pode-se perceber que profissionais desenvolvedores foram a maioria, representando 84%, e àqueles que trabalham com gestão são os 16%. Esta discrepância de área de atuação é importante para analisar de forma mais precisa as percepções de desenvolvedores em relação aos débitos técnicos.

No que diz respeito a faixa etária dos participantes, pode-se observar que 43 possuem menos de 25 anos, outros 31 possuem entre 26 e 30 anos, 7 possuem entre 31 e 35 anos, 2 possuem entre 41 e 45 anos, apenas um possui entre 36 e 40 anos e também apenas um possui entre 46 e 50 anos. A Figura 11 sintetiza esses dados.

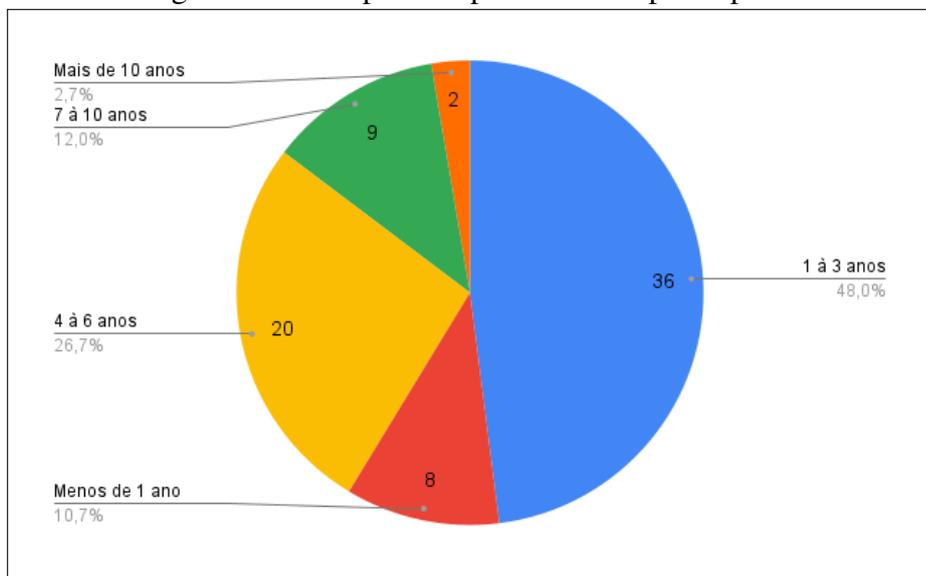
Figura 11 – Faixa etária dos participantes



Fonte: Elaborado pelo autor.

A Figura 12 apresenta o tempo de experiência dos participantes na área de Tecnologia da Informação. Observa-se que 36 participantes possuem entre 1 e 3 anos de experiência, 20 possuem entre 4 e 6 anos de experiência, 9 possuem entre 7 e 10 anos de experiência, 8 possuem menos de 1 ano de experiência e apenas 2 possuem mais de 10 anos de experiência. Ao analisar essas informações, percebe-se que 48% dos participantes só possuem entre 1 e 3 anos de experiência, isso pode ser um indicativo de imaturidade técnica dos profissionais pelo pouco tempo de experiência.

Figura 12 – Tempo de experiência dos participantes

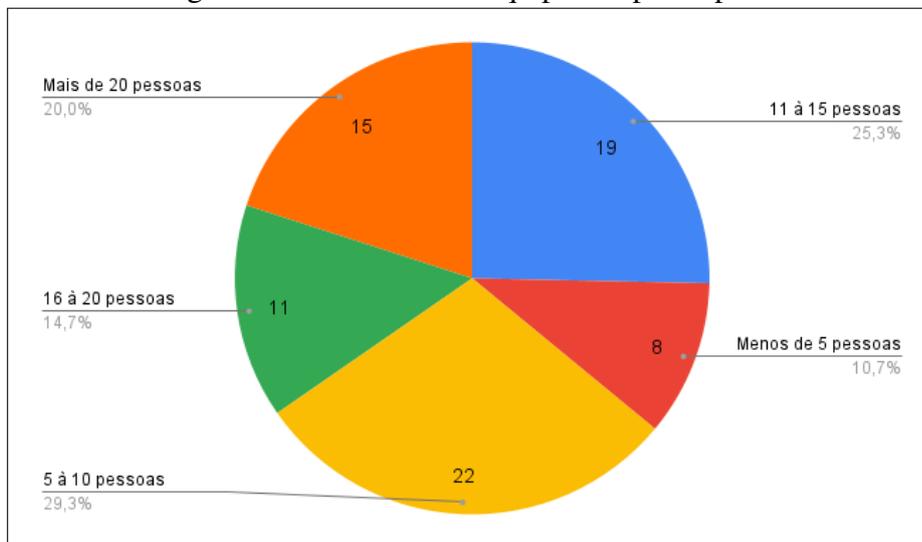


Fonte: Elaborado pelo autor.

## 5.2 Perfil do ambiente de trabalho do participante

Sobre o tamanho da equipe dos participantes, 22 possuem um time de 5 a 10 pessoas, 19 possuem de 11 a 15 pessoas, 15 possuem mais de 20 pessoas, 11 possuem de 16 a 20 pessoas e 8 possuem menos de 5 pessoas. Destes, pode-se perceber que os times entre 5 a 10 pessoas representam 29,3%, 11 a 15 pessoas 25,3% e mais de 20 pessoas com 20%, de modo que, observa-se a presença de equipes significativamente maiores, podendo indicar como os profissionais atuando no contexto de trabalho. A Figura 13 sintetiza esses dados.

Figura 13 – Tamanho da equipe dos participantes



Fonte: Elaborado pelo autor.

Em relação ao formato de trabalho, 56 (74,7%) participantes trabalham de forma remota, 10 (13,3%) trabalham de forma presencial e 9 (12%) trabalham de forma híbrida. O regime de trabalho é crucial para entender o contexto de rotinas diárias do profissional e em relação à interação e comunicação com os outros profissionais.

No que se refere a metodologia de gerenciamento de projetos, 74 (98,7%) participantes utilizam a metodologia ágil, e apenas um (1,3%) participante respondeu que faz uso da metodologia preditiva em seu projeto, dessa forma, de modo quase que unânime, a agilidade é instaurada nas atividades dos profissionais.

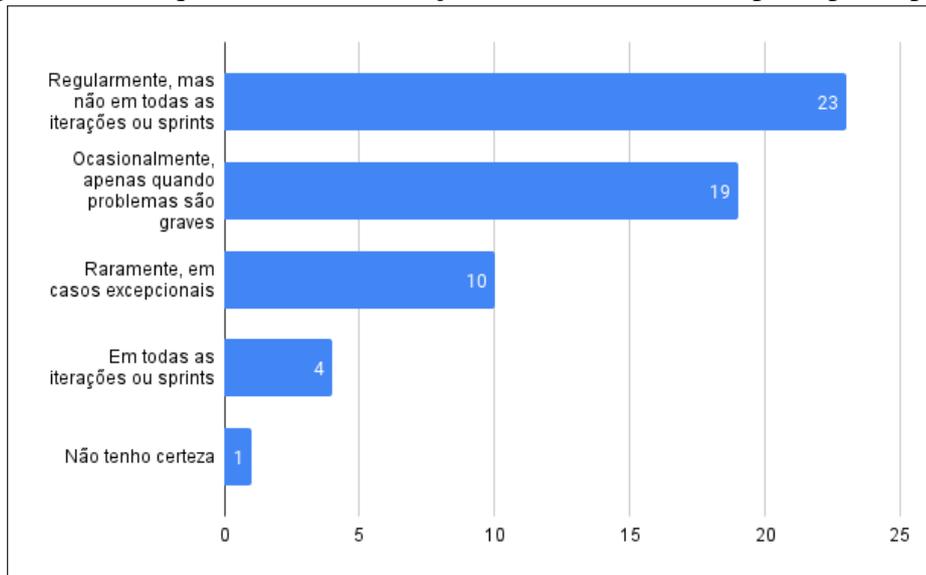
Por fim, foi realizada uma contextualização sobre o que são débitos técnicos aos participantes, dos 75 participantes, 57 (76%) deles tem familiaridade com o conceito e 18 (24%) disseram não ter familiaridade. Desses que responderam não, finalizaram o questionário neste momento, já que as perguntas seguintes eram de caráter específico sobre os débitos técnicos. Além disso, analisar que há um número representativo de participantes que não

possuem familiaridade com o conceito, é um tanto quanto importante para entender o papel dos débitos técnicos nos projetos.

### 5.3 Entendimento sobre débitos técnicos

Nesta seção, é iniciado perguntas voltadas aos débitos técnicos. A primeira é sobre a frequência que os débitos técnicos são identificados no contexto de trabalho dos participantes. Desse modo, 23 participantes identificam regularmente, mas não tem todas as iterações; 19 identificam ocasionalmente, apenas quando problemas são graves, 10 identificam raramente, em casos excepcionais; 4 identificam em todas as iterações ou sprints; e apenas um não tem certeza sobre a identificação. Com isso, pode-se perceber que há uma observação regular das questões relacionadas a débitos técnicos. Em paralelo, a quantidade de participantes que identificam ocasionalmente (19) também é significativa na análise de monitoramento dos débitos técnicos. A Figura 14 sintetiza esses dados.

Figura 14 – Frequência de identificação dos débitos técnicos pelos participantes



Fonte: Elaborado pelo autor.

No que se refere a comunicação dos débitos técnicos utilizada pelos participantes, 48 dos participantes fazem reuniões regulares com a equipe, 27 se comunicam de forma escrita por meio de e-mail, *chat*, por exemplo, e 29 utilizam ferramentas de rastreamento de problemas ou tarefas, como o *Jira*, *Trello*, *Github Issues*. Vale ressaltar que, os participantes podiam marcar mais de uma opção nesta pergunta. A Tabela 3 apresenta esses dados.

Tabela 3 – Comunicação dos débitos técnicos nas equipes dos participantes

<b>Comunicação</b>	<b>Valor (percentual)</b>
Em reuniões de equipe regulares	48 (77,4%)
Utilizando ferramentas de rastreamento de problemas ou tarefas ( <i>Jira, Trello, Github Issues, etc.</i> )	29 (46,8%)
Através de comunicação escrita (e-mail, <i>chat</i> , etc.)	27 (43,5%)

Fonte: Elaborado pelo autor.

Sobre as causas que acarretam o surgimento de débitos técnicos, pode-se observar uma variedade de causas, dentre elas: escopo mal definido (40), pressão para cumprir os prazos (40), documentação inadequada (39), falta de comunicação (35), mudança técnica (31) e falta de visão geral do projeto (30). Também vale mencionar que os participantes salientam a falta de conscientização da equipe, falta de suporte da alta gestão, resistência dos usuários e falta de recursos. Vale lembrar que, os participantes podiam responder mais uma causa. A Tabela 4 apresenta esses dados.

Tabela 4 – Causas que geram os débitos técnicos de acordo com os participantes

<b>Causa</b>	<b>Valor (percentual)</b>
Escopo mal definido	40 (64,5%)
Pressão para cumprir prazos	40 (64,5%)
Documentação inadequada	39 (62,9%)
Falta de comunicação	35 (56,5%)
Mudança técnica	31 (50%)
Falta de visão geral do projeto	30 (48,4%)
Falta de entendimento das dependências de atividades	28 (45,2%)
Falta de tempo	23 (37,1%)
Falta de conscientização da equipe	15 (24,2%)
Falta de suporte da alta gestão	12 (19,4%)
Usuários resistentes	8 (12,9%)
Falta de recursos	5 (8,1%)
Falta de gerenciamento de dívida técnica	1 (1,6%)
Clientes repriorizando entregas e requisitos, fazendo com que algumas partes do sistema sejam entregues às pressas ou deixadas de lado	1 (1,6%)
Muitos desenvolvedores já passaram pelo projeto, o que causa algumas inconsistências no código, na implementação ou na arquitetura do projeto	1 (1,6%)
Requisitos mal definidos	1 (1,6%)
Rotatividade de desenvolvedores	1 (1,6%)

Fonte: Elaborado pelo autor.

Em relação aos atributos de qualidade atingidos pelos débitos técnicos, 53 participantes afirmam que afeta a manutenibilidade, 44 afirmam prejudicar a confiabilidade, 41 afirmam dos danos na eficiência, 33 afirmam prejudicar as funcionalidades, 13 afirmam afetar a usabilidade e 9 afirmam afetar a portabilidade. Nesse sentido, pode-se observar que questões relacionadas principalmente à manutenção (85,5%), confiança (71%), eficiência (66,1%) e funcionamento (53,2%) do *software* são os atributos de qualidade diretamente afetados conforme os participantes. Vale lembrar que, os participantes podiam responder mais de um atributo. A Tabela 5 apresenta esses dados.

Tabela 5 – Atributos de qualidade afetados pelos débitos técnicos de acordo com os participantes

Atributo de qualidade	Valor (percentual)
Manutenibilidade	53 (85,5%)
Confiabilidade	44 (71%)
Eficiência	41 (66,1%)
Funcionalidade	33 (53,2%)
Usabilidade	13 (21%)
Portabilidade	9 (14,5%)

Fonte: Elaborado pelo autor.

Em relação à importância de cultura de qualidade e boas práticas no tratamento de débitos técnicos, 44 (77,2%) dos participantes responderam como "muito importante" e 13 (22,8%) responderam como "importante". Como descrito na metodologia da pesquisa, foi utilizada a escala Likert a fim de quantificar dados subjetivos e auxiliar na medição de percepções. Com isso, a escala foi baseada de 1 a 5, no qual 1 significa "não é importante", 2 é "pouco importante", 3 é "tanto faz", 4 é "importante" e 5 é "muito importante". De forma estatística, ao realizar uma média a partir das respostas dos participantes, pode-se concluir 4,77, indicando a alta importância à pergunta feita.

Foi perguntado aos participantes qual(is) é(são) o(s) principal(is) desafio(s) ao lidar com débitos técnicos por meio de uma pergunta qualitativa. A partir disso, foi utilizada a técnica de categorização das respostas para encontrar padrões existentes e relacionar o contexto da pesquisa. Desse modo, as seguintes categorias foram definidas:

- **Impacto na manutenibilidade:** 11 participantes destacaram que o principal desafio é a impacto na manutenibilidade do *software*. Eles mencionam que, ao lidar com débitos técnicos, a capacidade de manter o software e fazer alterações futuras é comprometida. Isso pode levar a problemas como dificuldade em adicionar novas funcionalidades, aumento de

erros e maior dificuldade na manutenção.

- **Integração com o escopo do projeto:** 10 participantes mencionaram que encaixar os débitos técnicos no escopo e cronograma do projeto é um desafio significativo. Isso pode ocorrer devido à pressão por cumprir prazos, mudanças técnicas ou a falta de recursos.
- **Complexidade na correção:** 8 participantes mencionaram que a complexidade em corrigir os débitos técnicos é um desafio, isso pode ocorrer porque a correção de um problema gerado por débitos técnicos pode gerar problemas em outras partes do código, resultando em erros em cascata.
- **Identificação dos débitos técnicos:** 8 participantes mencionam que esses débitos podem estar escondidos em várias partes do código ou do processo de desenvolvimento, e a equipe de desenvolvimento precisa ser capaz de reconhecer quando um débito técnico está sendo criado ou já existe.
- **Falta de rastreabilidade:** 7 participantes destacam a falta de rastreabilidade dos débitos técnicos como um desafio. Isso significa que os participantes têm dificuldade em identificar e rastrear os débitos técnicos, o que dificulta o tratamento oportuno.
- **Tomada de decisão e priorização:** 5 participantes mencionaram que a tomada de decisão e a priorização dos débitos técnicos são desafios iminentes, eles mencionaram a importância de definir prioridades e tratar os débitos técnicos com eficiência.

Vale destacar que, a ocorrência dessas categorizações são pertinentes para contextualizar as principais questões e desafios ao lidar com débitos técnicos. 13 participantes levantaram pontos sobre

Além disso, faz-se relevante destacar algumas respostas pertinentes dos participantes para fomentar o embasamento desses dados qualitativos:

- "A falta de rastreabilidade dos débitos técnicos para poderem ser tratados em tempo oportuno."
- "Explicar para quem define as prioridades de *tasks*, o quão importante é uma implementação bem feita."
- "Equilibrar a pressão por entregas rápidas com a necessidade de investir tempo na refatoração e manutenção do código, a fim de evitar acumulação excessiva".

Sobre a importância de gerenciar os débitos técnicos, 40 (70,2%) dos participantes responderam como "muito importante" e 17 (29,8%) responderam como "importante". Com isso, a escala Likert foi baseada de 1 a 5, no qual 1 significa "não é importante", 2 é "pouco

importante", 3 é "tanto faz", 4 é "importante" e 5 é "muito importante". De forma estatística, ao realizar uma média a partir das respostas dos participantes, pode-se concluir 4,7, indicando a alta importância à pergunta feita.

Ao fim da seção, faz-se relevante entender se há gerenciamento ou não dos débitos técnicos, dos 57 participantes, 35 (61,4%) responderam que os débitos técnicos são gerenciados, 19 (33,3%) responderam que não é gerenciado e 3 (5,3%) responderam que não sabem se é gerenciado. E a partir desta pergunta, surgem dois novos fluxos de respostas para os participantes, presença e ausência de gerenciamento de débitos técnicos.

#### 5.4 Entendimento sobre o gerenciamento de débitos técnicos

No início da seção é perguntado sobre os artefatos utilizados nas equipes dos participantes para organizar e gerenciar os débitos técnicos, 25 fazem revisões de código, 23 utilizam *backlog* de débitos técnicos, 22 utilizam histórias ou tarefas de refatoração, 20 utilizam registros de débitos técnicos, 16 utilizam documentação de arquitetura e *design*, 5 utilizam indicadores de saúde do projeto e 4 utilizam relatórios de avaliação de débitos técnicos. Vale lembrar que, os participantes podiam responder mais de um artefato. A Tabela 6 sumariza os resultados obtidos.

Tabela 6 – Artefatos e práticas para organizar e rastrear débitos técnicos nas equipes dos participantes

Artefato	Valor (percentual)
Revisões de Código	25 (69,4%)
<i>Backlog</i> de Débitos Técnicos	23 (63,9%)
Histórias ou Tarefas de Refatoração	22 (61,1%)
Registro de Débitos Técnicos	20 (55,6%)
Documentação de Arquitetura e <i>Design</i>	16 (44,4%)
Indicadores de saúde do projeto	5 (13,9%)
Relatórios de Avaliação de Débitos Técnicos	4 (11,1%)

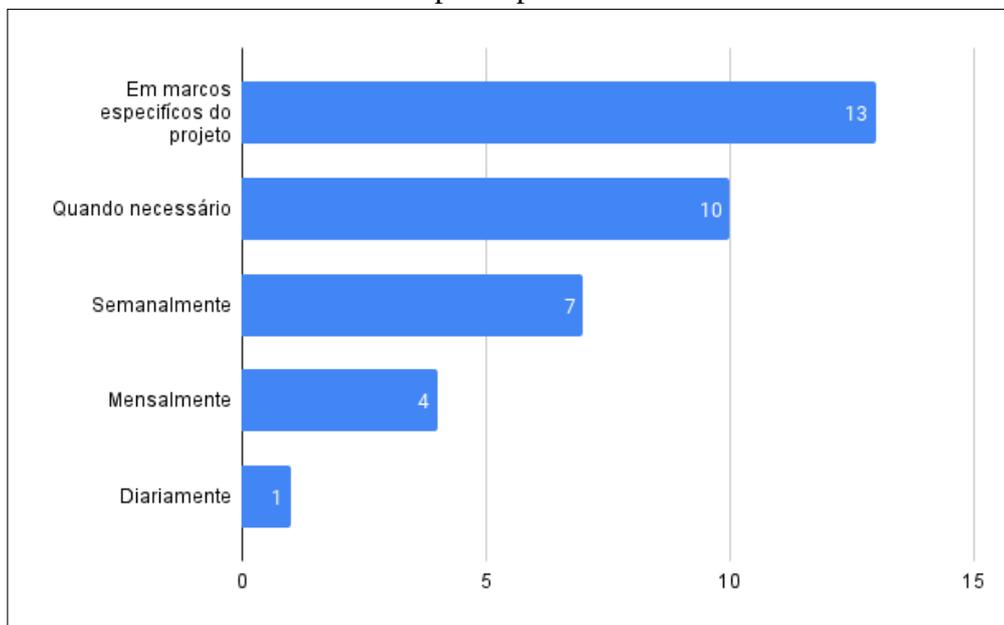
Fonte: Elaborado pelo autor.

Sobre o nível de envolvimento dos clientes ou usuários quando mencionado os débitos técnicos no contexto de trabalho dos participantes. 16 (45,7%) responderam que os clientes são pouco envolvidos, 8 (22,9%) dos participantes responderam que não são envolvidos, 5 (14,3%) dos participantes responderam que são envolvidos, 4 (11,4%) dos participantes responderam que para os clientes tanto faz e 2 (5,7%) dos participantes responderam que são muito envolvidos. Com isso, a escala Likert foi baseada de 1 a 5, no qual 1 significa "não são envolvidos", 2 é "pouco envolvidos", 3 é "tanto faz", 4 é "envolvidos" e 5 é "muito envolvidos".

De forma estatística, ao realizar uma média a partir das respostas dos participantes, pode-se concluir 2,34, indicando o pouco engajamento dos clientes ou usuários em questões referentes aos débitos técnicos.

Em relação à frequência que ocorrem atividades de gerenciamento de débitos técnicos nas equipes dos participantes. 13 afirmam que ocorrem em marcos específicos do projeto, 10 responderam que acontece quando necessário, 7 responderam que ocorrem semanalmente, 4 responderam que ocorrem mensalmente e apenas um respondeu que ocorrem diariamente. Nesse sentido, pode-se perceber a ocorrência é datada principalmente pela organização de marcos no projeto e em contextos de necessidade iminente. A Figura 15 sumariza os resultados obtidos.

Figura 15 – Frequência de atividades de gerenciamento de débitos técnicos nos projetos dos participantes



Fonte: Elaborado pelo autor.

Foi questionado aos participantes o que eles consideram como boas práticas para o gerenciamento de débitos técnicos. Nesta pergunta qualitativa, foi utilizada a técnica de categorização das respostas para encontrar padrões existentes e relacionar o contexto da pesquisa. Desse modo, as seguintes categorias foram definidas:

- **Monitoramento constante:** 10 participantes mencionaram a importância do monitoramento constante dos débitos técnicos, destacando a necessidade de estar ciente da presença desses débitos e estimar quando podem ser resolvidos.
- **Cultura de manutenção:** 7 participantes mencionaram a importância de manter uma cultura de manutenção do *software*, enfatizando a necessidade de revisões de código,

feedbacks e avaliação constante.

- **Refatoração:** 6 participantes destacaram a refatoração e reestruturação dos códigos como boas práticas para gerenciar débitos técnicos, indicando a necessidade de melhorar a qualidade do código existente.
- **Documentação:** 5 participantes ressaltaram a importância da documentação dos débitos técnicos, enfatizando a necessidade de documentar esses débitos e procurar resolvê-los o mais rápido possível.
- **Visibilidade e priorização:** 2 participantes destacaram a necessidade de ter visibilidade dos impactos dos débitos técnicos, priorizar os itens do backlog e dedicar tempo para evoluções técnicas como uma boa prática.
- **Reservar parte da *Sprint*:** 2 participantes sugeriram a alocação de parte das *sprints* para a resolução de débitos técnicos, garantindo que haja tempo dedicado a essas questões em cada ciclo de desenvolvimento.

Além disso, faz-se relevante destacar algumas respostas dos participantes para ajudar no embasamento desses dados qualitativos:

- "Sempre reservar parte do sprint para resolução desses débitos."
- "Identificação proativa, refatoração contínua e revisões de código."
- "Documentar quando ocorrem, o que pode ser feito para resolver (alinhamento da gestão com o time de desenvolvimento) e estruturar com o cliente as consequências destes débitos técnicos."

Por fim, foi perguntado sobre informações de melhorias ou sugestões que os participantes proporia para aprimorar o gerenciamento de débitos técnicos nos seus projetos. Por se tratar de uma pergunta qualitativa, as categorias definidas foram:

- **Reestruturação dos projetos:** 13 participantes propuseram a reestruturação dos projetos como uma melhoria, indicando que a qualidade do código e da arquitetura precisa ser revisada e aprimorada.
- **Melhorar comunicação:** 12 participantes destacaram a comunicação eficaz, sugerindo que uma melhor comunicação entre as equipes é essencial para resolver débitos técnicos.
- **Treinamento e conscientização:** 8 participantes propuseram treinamento e conscientização, destacando a necessidade de capacitar a equipe e os clientes sobre a importância dos débitos técnicos e como lidar com eles.

Algumas respostas dos participantes:

- "Canais de comunicação da própria da empresa, aonde será repassado conhecimentos, melhores práticas de desenvolvimento de *software* e estruturação do projeto."
- "Avaliação de riscos. Deixar débitos técnicos em um projeto pode ser muito custoso futuramente, portanto, deve ser fazer uma análise de escopo a cada escopo de tempo para verificar os débitos e conseguir uma manutenção de *software* coesa."
- "Conscientização e equilíbrio. Encontrar um equilíbrio entre atender às demandas de entrega e manter a qualidade técnica, fortalecer a comunicação."

### 5.5 Entendimento sobre a ausência de gerenciamento de débitos técnicos

De início, foi perguntado em relação à existência de processos eficazes para evitar os débitos técnicos, 12 (54,5%) participantes responderam que não existem, 9 (40,9%) responderam que não tem certeza e apenas um (4,5%) afirmou que existem processos nos seus projetos. Pode-se observar que este indicativo aponta desalinhamento de formas a fim de lidar com a iminência de débitos técnicos.

Posteriormente, foi realizada uma pergunta qualitativa que visa saber quais são as práticas adotadas pelos projetos dos participantes em relação ao monitoramento e rastreabilidade dos débitos técnicos. Dessa forma, foram criadas as categorizações a partir das respostas:

- **Foco em entregas de recursos:** 7 participantes mencionaram que o projeto prioriza a entrega de novas funcionalidades em detrimento da qualidade, assumindo a qualidade como uma dívida técnica que será tratada posteriormente.
- **Inexistência de prática:** 7 participantes informaram que não existe nenhuma prática, pelo menos por parte do conhecimento deles.
- **Comunicação e reuniões:** 5 participantes relataram práticas que envolvem comunicação diária, reuniões ou discussões informais para tratar de débitos técnicos.
- **Práticas de testes:** 3 participantes mencionaram que as principais práticas de monitoramento e rastreabilidade estão relacionadas aos testes, como testes unitários e testes de exploratórios.

Algumas respostas dos participantes:

- "Quando achamos algum débito técnico, é criado uma atividade de baixa prioridade, que raramente entra em uma *sprint*."
- "Quando aparece o problema, o time simplesmente vai 'apagar o incêndio' e segue nesse ritmo."

- "Eventualmente em reuniões quando surge algum problema e é falado de maneira informal."

Em relação aos benefícios ou vantagens de implementar boas práticas de gerenciamento de débitos técnicos, 21 participantes responderam sobre o aumento da qualidade do *software*, 21 responderam sobre a redução de custos de manutenção, 20 sobre a diminuição de problemas de escalabilidade, 18 sobre o aumento da produtividade da equipe, 10 responderam que há aumento da satisfação do cliente e um afirmou sobre aumento da saúde dos projetos e equipes durante o desenvolvimento. Vale lembrar que, os participantes podiam responder mais de um benefício. A Tabela 7 sumariza os resultados obtidos.

Tabela 7 – Benefícios de implementar boas práticas de gerenciamento de débitos técnicos de acordo com os participantes

Benefícios	Valor (percentual)
Maior qualidade do <i>software</i>	21 (95,4%)
Redução de custos de manutenção	21 (95,4%)
Menos problemas de escalabilidade	20 (90,9%)
Aumento da produtividade da equipe	18 (81,8%)
Maior satisfação do cliente	10 (45,4%)
Projetos e equipes mais saudáveis durante o desenvolvimento	1 (4,5%)

Fonte: Elaborado pelo autor.

Posteriormente, foi perguntado aos participantes se em seus projetos já enfrentaram consequências negativas decorrentes da ausência do gerenciamento de débitos técnicos. Por se tratar de uma pergunta qualitativa, foram definidos tais categorias de respostas:

- **Impacto nas entregas:** 5 participantes mencionaram que a ausência de gerenciamento de débitos técnicos afetou as entregas do projeto, causando atrasos e problemas nas *releases*.
- **Qualidade do código:** 3 participantes destacaram que a ausência de gerenciamento de débitos técnicos resultou em problemas de qualidade do código, incluindo erros em produção e alto custo para refatoração.
- **Prejuízos ao cliente:** 2 participantes relataram que a ausência de gerenciamento de débitos técnicos causou prejuízos ao cliente e insatisfação, resultando em instabilidade da gestão.
- **Impacto na manutenção:** 4 participantes mencionaram que o gerenciamento inadequado de débitos técnicos dificultou a manutenção do sistema e a introdução de novas funcionalidades.
- **Reprojeção do funcionamento:** 2 participantes indicaram que a ausência de geren-

ciamento de débitos técnicos levou à reprojeção do funcionamento de determinadas funcionalidades.

- **Desorganização e bugs:** 2 participantes mencionaram que a falta de gerenciamento adequado resultou em desorganização e geração de *bugs* no projeto, destacando o gerenciamento das tarefas dos projetos.

Algumas respostas dos participantes:

- "Sem dúvida, problemas que afetam o ambiente de produção e causam prejuízo no sistema."
- "A aplicação subiu para produção cheia de erros e tivemos um alto gasto para refatorar."
- "O maior problema está em dar manutenção no sistema ou trabalhar novas funcionalidades."

Em seguida, foi perguntado aos participantes se há alguma justificativa ou razão específica para a ausência do gerenciamento de débitos técnicos nos seus projetos. Por se tratar de uma pergunta qualitativa, foram definidos tais categorias de respostas:

- **Ausência de processos:** 8 participantes indicaram que a ausência de processos específicos impacta na falta de diretrizes claras, práticas e procedimentos formalizados para identificar, avaliar e abordar débitos técnicos.
- **Falta de compreensão e conhecimento:** 4 participantes apontaram a falta de compreensão e conhecimento sobre débitos técnicos como uma das razões para a ausência de gerenciamento. Isso inclui falta de entendimento da gestão e da equipe de desenvolvimento.
- **Falta de planejamento:** 3 participantes apontaram que a falta de planejamento pode incluir a ausência de consideração durante a fase de planejamento do projeto sobre como lidar com possíveis débitos técnicos que podem surgir ao longo do ciclo de vida do *software*.
- **Falta de tempo e recursos:** 2 participantes mencionaram que a falta de tempo e recursos é uma justificativa para a ausência de gerenciamento de débitos técnicos, devido a prazos apertados, falta de mão de obra ou recursos insuficientes.
- **Gestão com pouca competência técnica:** 2 participantes relataram que a falta de competência técnica na alta gestão, de modo que, quando a liderança não compreende completamente as implicações técnicas e os benefícios associados à gestão proativa de débitos técnicos, pode haver uma falta de suporte e direção para implementar estratégias eficazes.

Algumas respostas dos participantes:

- "Não tem incentivo, talvez falte algum treinamento."
- "Gestão não se importa muito, e internamente o time de desenvolvimento não conhece muito bem de práticas que possam ajudar."
- "Falta de importância do CTO."

Por fim, foi perguntado aos participantes sobre a relevância de um artefato capaz de auxiliar no processo de gerenciamento de débitos técnicos. 12 (54,5%) dos participantes responderam que seria muito relevante, 9 (40,9%) participantes sinalizaram como relevante e apenas um (4,5%) participante respondeu que seria pouco relevante. Desse modo, como a pergunta segue a padronização da escala Likert, sendo baseada de 1 a 5, no qual 1 significa "não é relevante", 2 é "pouco relevante", 3 é "tanto faz", 4 é "relevante" e 5 é "muito relevante". De forma estatística, ao realizar uma média a partir das respostas dos participantes, pode-se concluir 4,48, indicando uma relevância significativa à pergunta feita.

## 6 GUIA

Nesta seção, é apresentado o guia construído após as análises feitas por meio do questionário, os trabalhos relacionados e referenciais teóricos do presente trabalho.

### 6.1 Introdução

O guia pretende fornecer orientações e boas práticas às equipes de desenvolvimento de *software* para o gerenciamento eficaz de débitos técnicos. Débitos técnicos fazem parte do desenvolvimento de *software*, e a falta de gerenciamento adequado pode levar a consequências negativas. Este guia oferece uma estrutura abrangente para identificação, monitoramento, priorização, resolução e prevenção de débitos técnicos.

### 6.2 O que são débitos técnicos?

Débitos técnicos referem-se a decisões ou ações tomadas durante o desenvolvimento de *software* que priorizam soluções de curto prazo em detrimento de práticas adequadas de desenvolvimento. Isso resulta em problemas técnicos, instabilidade e risco aumentado no projeto.

#### **Exemplos de débitos técnicos:**

- Código-fonte mal estruturado e desorganizado;
- Falta de documentação significativa do código;
- Vulnerabilidades de segurança não tratadas;
- Falta de testes adequados;
- Soluções de curto prazo que não consideram escalabilidade;
- Uso excessivo de soluções "gambiarras" para resolver problemas.

### 6.3 Por que gerenciar débitos técnicos?

O gerenciamento de débitos técnicos é vital para manter a saúde do projeto. Através do reconhecimento, registro, priorização e resolução eficiente dos débitos técnicos, a equipe pode evitar consequências negativas e garantir um desenvolvimento de *software* mais eficaz e sustentável. Além da importância de alinhar as expectativas sobre o que são débitos técnicos e por que devem ser tratados com seriedade. **Impactos da ausência de gerenciamento dos débitos técnicos:**

- **Deterioração da qualidade do *software*:** débitos técnicos resultam em código mal estruturado e desorganizado, o que prejudica a qualidade do *software*. A falta de manutenção adequada pode levar a sistemas ineficientes e difíceis de entender;
- **Dificuldade em adicionar novas funcionalidades:** sistemas com débitos técnicos tornam-se menos flexíveis e ágeis. Adicionar novas funcionalidades requer esforço adicional devido à complexidade técnica existente;
- **Aumento de erros e falhas:** débitos técnicos frequentemente contribuem para erros e falhas. Isso pode prejudicar a confiabilidade do *software* e causar problemas para os usuários finais;
- **Dificuldades na manutenção:** A manutenção de sistemas com débitos técnicos torna-se um desafio. Corrigir problemas existentes consome tempo e recursos, o que afeta a produtividade da equipe;
- **Atrasos e custos adicionais no projeto:** O acúmulo de débitos técnicos pode levar a atrasos nas entregas e custos extras, à medida que a equipe precisa resolver problemas técnicos que poderiam ter sido evitados.

#### 6.4 Como identificar os débitos técnicos?

Identificar e reconhecer os débitos técnicos é o primeiro passo no processo de gerenciamento eficaz. Débitos técnicos podem ser sutis e muitas vezes se escondem profundamente no código e nos processos de desenvolvimento. Reconhecer esses débitos enquanto o desenvolvimento está em andamento é fundamental para evitar problemas futuros.

Para reconhecer débitos técnicos, é importante que a equipe esteja atenta a sinais de alerta. Isso pode incluir a identificação de práticas de codificação questionáveis, a percepção de que um atalho está sendo tomado para cumprir prazos ou a observação de código que parece frágil e propenso a falhas. O desenvolvimento de uma cultura de responsabilidade técnica e qualidade é o primeiro passo para a identificação de débitos técnicos. **Principais sinais de débitos técnicos:**

- Código-fonte desorganizado ou mal estruturado;
- Falta de documentação adequada;
- Altas taxas de erro e falhas frequentes;
- Reclamações constantes dos usuários;
- Dificuldade em estender ou modificar o sistema;

– Lentidão na entrega de novas funcionalidades.

Incentive a equipe a ser proativa na busca e identificação de possíveis débitos técnicos, estejam eles no código, na arquitetura ou nos processos de desenvolvimento. Além disso, crie um ambiente onde o time se sinta à vontade para relatar preocupações e compartilhar suas percepções sobre possíveis débitos técnicos.

A identificação de débitos técnicos não deve ser responsabilidade exclusiva dos desenvolvedores, tente envolver toda a equipe no processo. Testadores, arquitetos, gerentes de projeto, *designers* e outras funções podem contribuir significativamente para a identificação de débitos técnicos. Eles podem trazer perspectivas diferentes e identificar problemas que podem passar despercebidos pelos desenvolvedores.

A comunicação transparente e aberta entre os membros da equipe é crucial para identificar débitos técnicos de maneira eficaz. Estabelecer um ambiente propício à expressão de preocupações e desafios contribui para a detecção precoce dessas questões. Uma comunicação aberta fomenta também em uma cultura de aprendizado contínuo. A equipe se beneficia ao compartilhar experiências e lições aprendidas, enriquecendo o entendimento sobre práticas eficazes de gerenciamento de débitos técnicos.

Promova a colaboração entre as diferentes funções da equipe, incentivando discussões regulares sobre possíveis débitos técnicos e como eles podem ser evitados ou tratados. Isso pode ser feito por meio de reuniões de revisão de código, retrospectivas de projeto ou outras práticas ágeis. Além disso, crie canais de comunicação onde os membros da equipe possam relatar possíveis débitos técnicos de forma eficaz.

## **6.5 Como monitorar os débitos técnicos?**

O monitoramento contínuo de débitos técnicos é essencial para evitar que se acumulem e se transformem em problemas que afetam a qualidade e a eficiência do desenvolvimento de *software*. Sem um processo de monitoramento eficaz, é fácil que os débitos técnicos sejam negligenciados e se acumulem com o tempo.

A equipe deve entender que monitorar débitos técnicos é um esforço constante. Desse modo, deve estar constantemente atenta aos débitos técnicos já identificados, a detecção precoce de novos débitos técnicos e a tomada de medidas proativas. A rastreabilidade deve ocorrer não apenas durante o desenvolvimento, mas também durante a manutenção e a evolução do *software*. O monitoramento contínuo permite a detecção prévia de novos débitos técnicos e a

tomada de medidas proativas. **Algumas práticas de rastreabilidade:**

- **Documentação de código:** manter uma documentação clara e atualizada é essencial para rastrear débitos técnicos. Documentar áreas de código problemáticas e detalhar soluções propostas pode ajudar na identificação e tratamento;
- **Registros de erros e correções:** manter um registro de erros e correções é valioso para identificar áreas críticas com débitos técnicos. Isso inclui relatórios de *bugs*, solicitações de correção e registros de problemas conhecidos;
- **Avaliação de desempenho:** o desempenho do sistema deve ser monitorado continuamente. Decréscimos significativos de desempenho podem indicar a presença de débitos técnicos.
- **Revisões de código:** são oportunidades para identificar e discutir possíveis débitos técnicos. A equipe deve estar atenta a práticas de programação inadequadas durante as revisões;
- **Métricas de qualidade de código:** acompanhar métricas de qualidade de código, como complexidade ciclomática, cobertura de código e índices de duplicação, podem ajudar a identificar áreas problemáticas;
- **Feedback contínuo:** encorajar os membros da equipe a fornecer *feedback* sobre áreas problemáticas ou práticas que podem resultar em débitos técnicos.

## 6.6 Como priorizar os débitos técnicos?

A priorização é uma etapa crucial no gerenciamento de débitos técnicos. Não é realista ou eficaz abordar todos os débitos técnicos de uma vez, é necessário definir critérios claros para a priorização e estabelecer estratégias para gerenciar essas prioridades. A priorização deve ser baseada em critérios sólidos e alinhados com os objetivos do projeto e as necessidades da organização. **Alguns critérios comuns para a priorização de débitos técnicos incluem:**

- **Impacto no cliente:** avaliar o impacto dos débitos técnicos nas operações do cliente e na experiência do usuário;
- **Impacto na qualidade:** considerar como os débitos técnicos afetam a qualidade do *software*;
- **Risco:** avaliar o risco de deixar um débito técnico não resolvido;
- **Custo de resolução:** estimar o custo e o esforço necessários para resolver um débito técnico;
- **Valor de negócio:** avaliar se a resolução do débito técnico agregará valor ao negócio.

Certifique-se de que a equipe compreenda e esteja alinhada com os critérios de

priorização definidos. Isso ajudará a evitar conflitos e garantir que os débitos técnicos mais críticos sejam tratados primeiro.

### 6.7 Como resolver os débitos técnicos?

Resolver débitos técnicos é o passo subsequente à priorização. É aqui que a equipe deve encontrar estratégias eficazes para lidar com os débitos técnicos. Lidar com débitos técnicos durante o desenvolvimento pode ser desafiador, mas é uma prática crucial e estratégica. **Para resolver de maneira eficaz:**

- **Crie um plano de ação:** projete um plano claro para resolver cada débito técnico, incluindo atribuição de responsabilidades e prazos. Isso ajuda na execução eficaz;
- **Analise o impacto:** avalie como a resolução de um débito técnico afetará outras partes do sistema. Isso é fundamental para evitar erros em cascata;
- **Aloque tempo para refatoração:** reserve tempo nas iterações para lidar com débitos técnicos à medida que são identificados. Isso evita que se acumulem ao longo do tempo;
- **Evite soluções de curto prazo:** desvie de soluções rápidas que possam criar mais débitos técnicos. Em vez disso, busque abordagens que resolvam o problema fundamental;
- **Implemente testes:** criar testes automatizados ajuda a identificar e evitar a reintrodução de débitos técnicos;
- **Colabore com a equipe:** envolva a equipe no processo de resolução de débitos técnicos. Discussões e revisões de código são momentos ideais para identificar e resolver esses problemas.

### 6.8 Como prevenir os débitos técnicos?

A importância da prevenção e adoção de boas práticas em projetos de software para gerenciar débitos técnicos reside na garantia da qualidade, sustentabilidade e eficiência do software ao longo do tempo. A prevenção busca evitar a acumulação de dívidas técnicas desde as fases iniciais do desenvolvimento, promovendo decisões e práticas que minimizam a ocorrência de problemas futuros. Isso não apenas reduz os custos associados à correção posterior, mas também contribui para a manutenção de uma arquitetura sólida e flexível.

A instauração de boas práticas em todos os estágios do ciclo de vida do software é essencial para mitigar riscos e facilitar a identificação precoce de potenciais débitos técnicos. Ao

incorporar práticas eficientes de desenvolvimento, gestão de projetos e manutenção, as equipes conseguem gerenciar proativamente e abordar as dívidas técnicas antes que se tornem onerosas. Isso resulta em sistemas mais robustos, com menor probabilidade de falhas, maior facilidade de manutenção e maior satisfação do usuário. **Boas práticas essenciais para garantir a saúde do projeto:**

- **Definição de padrões de codificação:** incentive a escrita de código limpo e organizado. Estabelecer padrões de codificação sólidos é essencial para manter a consistência e a qualidade do código, isso inclui convenções de nomenclatura, estrutura de diretórios, funções curtas e bem definidas, e a eliminação de código não utilizado ou redundante. Certifique-se de que toda a equipe esteja ciente desses padrões e os siga de perto. Além disso, promova a atualização regular desses padrões à medida que a equipe ganha experiência e o projeto evolui, código limpo é mais fácil de entender, manter e estender;
- **Documentação adequada:** a documentação é essencial para garantir que o código seja compreensível para outros desenvolvedores. Certifique-se de documentar o código concisamente, explicando sua funcionalidade, uso e quaisquer decisões de *design* importantes. Além disso, mantenha a documentação atualizada à medida que o código evolui. Isso é fundamental para facilitar a manutenção e o entendimento do sistema;
- **Revisão de código contínua:** a revisão de código não é apenas uma prática de prevenção, mas também uma parte vital das boas práticas de desenvolvimento. Realize revisões de código regularmente, mesmo após a implementação, para identificar e corrigir possíveis débitos técnicos. Além disso, promova uma cultura de revisão de código entre os membros da equipe, incentivando a colaboração e a troca de conhecimento;
- **Mentoria e treinamento:** promova um ambiente de aprendizado dentro da equipe. Estabeleça um sistema de mentoria onde membros mais experientes possam orientar aqueles com menos experiência. Além disso, ofereça oportunidades de treinamento e desenvolvimento para que todos os membros da equipe possam aprimorar suas habilidades. O investimento em educação e desenvolvimento técnico é essencial para manter a equipe atualizada com as melhores práticas de desenvolvimento;
- **Iteração contínua:** mantenha um ciclo de desenvolvimento contínuo, isso inclui a melhoria constante do código existente, identificação e resolução de débitos técnicos à medida que surgem, e a busca constante pela otimização e eficiência no processo de desenvolvimento. Incentive a equipe a adotar uma mentalidade de melhoria contínua e a abraçar a

- ideia de que a qualidade do código é uma responsabilidade de todos;
- **Desenvolvimento orientado a testes:** implemente testes unitários, testes de integração e testes de aceitação desde o início do desenvolvimento, ajudando a identificar problemas de forma precoce. Os testes garantem que as alterações no código não introduzam erros inesperados e são uma parte crucial da prevenção de débitos técnicos. Incentive a equipe a escrever testes robustos e execute-os regularmente como parte do processo de desenvolvimento;
  - **Gestão eficiente de dependências:** gerencie cuidadosamente as dependências do sistema. Evite dependências excessivas e mantenha uma visão clara das relações entre os componentes, isso reduz a complexidade e minimiza os riscos associados a mudanças em bibliotecas ou *frameworks*.
  - **Design modular e desacoplado:** adote uma abordagem de *design* modular e desacoplado para o sistema. Componentes bem definidos e desacoplados reduzem a propagação de impactos negativos, facilitando a manutenção e minimizando a criação de débitos técnicos.
  - **Revisão de requisitos:** um dos principais pontos de partida para a prevenção de débitos técnicos é garantir que os requisitos do projeto sejam claros e bem definidos desde o início. Requisitos vagos ou mal compreendidos podem levar a decisões de desenvolvimento precipitadas que resultam em débitos técnicos. É fundamental que a equipe de desenvolvimento e partes interessadas estejam alinhados em relação aos requisitos do projeto;
  - **Gerenciamento de projetos adequado:** além das práticas de desenvolvimento, é importante ter um gerenciamento de projetos eficaz, isso inclui a definição clara de metas, prazos realistas e alocar recursos adequados. Uma boa gestão de projetos ajuda a evitar a pressão por entregas rápidas que podem levar a escolhas técnicas ruins.

Estas boas práticas e estratégias de prevenção não apenas ajudam a evitar a criação de débitos técnicos, mas também contribuem para a construção de sistemas de *software* mais confiáveis, sustentáveis e fáceis de manter. A implementação dessas práticas deve ser parte integrante da cultura de desenvolvimento da equipe.

## 6.9 Conclusão

Em suma, o gerenciamento de débitos técnicos é uma parte fundamental do processo de desenvolvimento de *software*. Não se trata apenas de corrigir problemas técnicos, mas de adotar uma abordagem proativa para prevenir e resolver questões que podem prejudicar

a qualidade, a eficiência e a sustentabilidade de um projeto. Este guia oferece orientações abrangentes para enfrentar os desafios relacionados aos débitos técnicos de forma sistemática e eficaz.

Reconhecer a importância da identificação e monitoramento contínuos de débitos técnicos é o primeiro passo para o sucesso. Envolver toda a equipe no processo de detecção ajuda a criar uma cultura de responsabilidade técnica e compartilhada. Além disso, priorizar e resolver débitos técnicos de acordo com critérios sólidos é essencial para garantir que as questões mais críticas sejam tratadas prontamente.

A prevenção é outra faceta crucial do gerenciamento de débitos técnicos. Implementar boas práticas de codificação, realizar revisões regulares de código e manter documentação adequada são estratégias preventivas eficazes. A integração de atividades relacionadas a débitos técnicos nas iterações regulares assegura que essas questões sejam abordadas de maneira contínua e sustentável.

Em última análise, a importância do gerenciamento de débitos técnicos se reflete na qualidade do *software* entregue e na satisfação do cliente. Ignorar débitos técnicos pode levar a consequências negativas, como atrasos nas entregas, problemas em produção e custos mais elevados. Portanto, adotar uma abordagem proativa para gerenciar débitos técnicos é essencial para o sucesso de qualquer projeto de desenvolvimento de *software*.

Lembre-se de que o gerenciamento de débitos técnicos não é responsabilidade exclusiva de um único membro da equipe, mas sim um esforço conjunto de todos os envolvidos no desenvolvimento de *software*. Ao seguir as práticas recomendadas neste guia, sua equipe estará melhor preparada para enfrentar os desafios técnicos com confiança, resultando em *software* de alta qualidade e projetos bem-sucedidos a longo prazo.

Este presente guia é fundamentado a partir de um Trabalho de Conclusão de Curso, de modo que, foi construído por meio de uma pesquisa realizada com profissionais da área de Tecnologia da Informação, mas especificamente desenvolvedores e gerentes de projetos/produtos. Além disso, o material científico utilizado como referencial no Trabalho de Conclusão de Curso foi imprescindível na sustentação de uma base teórica robusta e embasada para o guia.

## 7 SÍNTESE

Na análise do entendimento dos débitos técnicos, a pesquisa destacou a riqueza na diversidade das formações acadêmicas e áreas de atuação dos participantes. Isso não apenas ressalta a abrangência do tema, mas também evidencia a necessidade de uma compreensão holística para lidar com os desafios relacionados a débitos técnicos. A inclusão tanto de desenvolvedores quanto de gerentes de projetos proporcionou uma visão abrangente dos obstáculos e práticas associados a esses débitos.

A investigação detalhada da faixa etária e experiência dos participantes revelou nuances importantes. A influência significativa da maturidade técnica nas abordagens para resolução de débitos técnicos sugere a importância de estratégias personalizadas, enquanto a variedade de contextos organizacionais, notadamente no cenário de trabalho remoto e na predominância da metodologia ágil, destaca a necessidade de adaptação constante no desenvolvimento de *software*.

A pesquisa revela que 76% dos participantes estão familiarizados com débitos técnicos, indicando uma consciência contínua sobre o tema, mas um número significativo de participantes (18) sequer tinham familiaridade com o conceito de débitos técnicos, e isto deve ser tópico de alerta e análise no que se refere ao conhecimento técnico. A comunicação sobre débitos técnicos ocorre principalmente por meio de reuniões regulares (77,4%), comunicação escrita (43,5%) e ferramentas de rastreamento (46,8%). As causas dos débitos técnicos incluem escopo mal definido (64,5%), pressão por prazos (64,5%) e documentação inadequada (62,9%), impactando principalmente a manutenibilidade (85,5%), confiabilidade (71%) e eficiência (66,1%) do *software*.

A cultura de qualidade e boas práticas é considerada "muito importante" por 77,2% dos participantes. Desafios identificados incluem impacto na manutenibilidade, integração com o escopo do projeto, falta de rastreabilidade, complexidade na correção, identificação e tomada de decisão. A gestão de débitos técnicos é considerada "muito importante" por 70,2% dos participantes. Na análise qualitativa dos desafios ao lidar com débitos técnicos, emergem categorias como impacto na manutenibilidade, integração com o escopo do projeto, falta de rastreabilidade, complexidade na correção, identificação e tomada de decisão. Esses desafios ressoam com as complexidades inerentes ao tratamento de débitos técnicos, fornecendo visões valiosas para estratégias de mitigação.

No âmbito do gerenciamento de débitos técnicos, a pesquisa trouxe à luz as práticas e artefatos mais adotadas, como revisões de código, *backlog* e histórias de refatoração. Essas

práticas integradas ao ciclo de desenvolvimento não apenas refletem a preferência dos participantes, mas também indicam a importância de uma abordagem contínua e incorporada ao processo de desenvolvimento para lidar com débitos técnicos.

As sugestões de melhoria oferecidas pelos participantes adicionaram camadas de compreensão às práticas eficazes. Estratégias como monitoramento constante, uma cultura de manutenção e investimentos em treinamento foram destacadas, indicando que a gestão eficaz de débitos técnicos não é apenas uma questão de ferramentas, mas também de processos e cultura organizacional.

Em contraste, a análise da ausência de gerenciamento de débitos técnicos revelou desafios substanciais. A falta de processos eficazes e práticas específicas, conforme indicado por mais da metade dos participantes, enfatiza a necessidade urgente de uma abordagem mais proativa na gestão desses débitos.

Portanto, em síntese, os resultados da pesquisa iluminam a complexidade inerente ao gerenciamento de débitos técnicos, enfatizando a necessidade de uma abordagem multifacetada, com destaque para a conscientização contínua e práticas organizacionais que priorizem a qualidade do *software*. Essa compreensão profunda é essencial para promover o sucesso sustentável dos projetos de *software* em ambientes dinâmicos e diversificados.

## 8 CONSIDERAÇÕES FINAIS

Este trabalho realizou uma análise quantitativa e qualitativa das percepções de profissionais da área de Tecnologia da Informação de desenvolvimento de *software* e gerenciamento de projetos/produtos voltada a entender as percepções e contextualização dos débitos técnicos nos projetos de *software* a fim de compreender atividades de gerenciamento destes débitos técnicos. A partir da pesquisa com 75 respostas válidas, pode-se concluir a importância de gerenciar as dívidas técnicas, instaurar processos e práticas recorrentes nos projetos focado em identificação, monitoramento, priorização e prevenção.

A fim de disseminação da pesquisa, foi construído um guia para auxiliar no gerenciamento de débitos técnicos, objetivando a sinergia entre academia e mercado de trabalho acerca destes problemas. O guia é pautado nos resultados obtidos com a pesquisa e os recursos científicos utilizados neste presente trabalho, além disso, é um artefato estruturado no intuito de formalizar atividades de gerenciamento de débitos técnicos e fomento à difusão do conhecimento sobre questões referentes a estas dívidas técnicas. O guia está no Apêndice E.

### 8.1 Limitações da pesquisa

É importante destacar algumas limitações inerentes a esta pesquisa. Em primeiro lugar, a amostra, embora abrangente, é finita, representando uma parcela específica de profissionais e contextos de desenvolvimento de *software* (75 participantes). Isso pode influenciar a generalização dos resultados para além do escopo deste estudo. Além disso, a natureza autorrelatada das respostas pode introduzir vieses de resposta e subjetividade. As interpretações dos participantes sobre conceitos específicos, como débitos técnicos, podem variar, afetando a consistência das respostas. Essas limitações devem ser consideradas ao interpretar e aplicar os resultados desta pesquisa.

### 8.2 Principais contribuições

Apesar das limitações, esta pesquisa oferece contribuições significativas para o entendimento do gerenciamento de débitos técnicos no contexto de projetos de desenvolvimento de *software*. As principais contribuições incluem uma visão abrangente sobre a prevalência de débitos técnicos, as práticas de gerenciamento adotadas e os impactos percebidos. A análise das causas subjacentes, a avaliação das consequências da ausência de gerenciamento e a identificação

das práticas consideradas eficazes pelos profissionais fornecem percepções valiosas para a comunidade de desenvolvimento de *software*. Essas contribuições têm o potencial de orientar práticas e estratégias futuras no gerenciamento de débitos técnicos.

Além disso, destaca-se como uma contribuição relevante a elaboração do guia, resultante das pesquisas conduzidas e do presente trabalho, de modo que, representa uma ferramenta prática e aplicável que emerge da análise das práticas adotadas por profissionais. O guia visa fornecer tópicos claros e acionáveis para equipes de desenvolvimento lidarem proativamente com débitos técnicos, contribuindo assim para a melhoria contínua do *software*.

### **8.3 Trabalhos futuros**

Considerando a complexidade e a dinâmica em constante evolução do desenvolvimento de *software*, várias direções podem ser exploradas em pesquisas futuras. Em primeiro lugar, a investigação mais aprofundada sobre práticas específicas de gerenciamento de débitos técnicos pode oferecer percepções detalhadas sobre estratégias eficazes. Além disso, a exploração de soluções tecnológicas, como ferramentas automatizadas para detecção e mitigação de débitos técnicos, pode ser um campo promissor para pesquisas futuras. Além disso, a extensão desta pesquisa para diferentes tipos de projetos pode enriquecer ainda mais a compreensão dos desafios específicos enfrentados em diferentes contextos.

Adicionalmente, há uma oportunidade de pesquisa promissora na validação prática do guia em um contexto real de um projeto de *software*. A implementação e avaliação deste guia em situações do mundo real proporcionariam visões valiosas sobre sua eficácia, aceitação pelos profissionais e sua capacidade de contribuir para a melhoria do gerenciamento de débitos técnicos. Essa abordagem prática poderia validar e aprimorar ainda mais os tópicos abordados no guia, adaptando-os às demandas e complexidades específicas de diferentes projetos.

## REFERÊNCIAS

- ALBUQUERQUE, D.; GUIMARAES, E. T.; TONIN, G. S.; PERKUSICH, M. B.; ALMEIDA, H.; PERKUSICH, A. Perceptions of technical debt and its management activities - a survey of software practitioners. **Association for Computing Machinery**, p. 220–229, 2022.
- ALLMAN, E. Managing technical debt: Shortcuts that save money and time today can cost you down the road. **Queue**, v. 3, n. 3, p. 10–17, 2012.
- ALMEIDA, R. R. de. Business-driven technical debt prioritization. In: **2019 IEEE International Conference on Software Maintenance and Evolution (ICSME)**. Nova Iorque, Estados Unidos: [S. n.], 2019. p. 605–609.
- BESKER, T.; GHANBARI, H.; MARTINI, A.; BOSCH, J. The influence of technical debt on software developer morale. **Journal of Systems and Software**, v. 167, 2020.
- BOURQUE, P.; FAIRLEY, R. **Guide to the Software Engineering Body of Knowledge**. [S. l.]: IEEE Computer Society, 2014.
- BROWN, N.; CAI, Y.; GUI, Y.; ZAZMAN, R.; KIM, M.; KRUCHTEN, P.; LIM, E.; MACCORMACK, A.; NORD, R.; OZKAYA, I.; SANGWAN, R.; SEAMAN, C.; SULLIVAN, K.; ZAZWORKA, N. Managing technical debt in software-reliant systems. In: **Proceedings Of The Fse/Sdp Workshop On Future Of Software Engineering Research**. Nova Iorque, Estados Unidos: [S. n.], 2010. p. 47–52.
- BUSCHMANN, F. To pay or not to pay technical debt. **IEEE Software**, v. 28, p. 29–31, 2011.
- CARVALHO, M. M. **Fundamentos em Gestão de Projetos - Construindo Competências para Gerenciar Projetos**. São Paulo, Brasil: Editora: Grupo GEN, 2018.
- CUNNINGHAM, W. **Debt Metaphor**. 2009. Disponível em: <https://www.youtube.com/watch?v=pqeJFYwnkjE>. Acesso em: 29 maio 2023.
- DAHL, J. **Liderança Lean**. [S. l.]: Editora Alta Books, 2021.
- ERNST, N. A.; WILLIAMS, C.; KALLIAMVAKOU, E. The who, what, how of software engineering research: a socio-technical framework. **Empir Software Eng**, v. 25, p. 4097–4129, 2020.
- FAIRBANKS, G. Ur-technical debt. **IEEE Software**, v. 37, p. 95–98, 2020.
- FALESSI, D.; KRUCHTEN, P. Five reasons for including technical debt in the software engineering curriculum. In: **Proceedings of the 2015 European Conference on Software Architecture Workshops**. [S. l.]: Association for Computing Machinery, 2015.
- FERNÁNDEZ-SÁNCHEZ, C.; GARBAJOSA, J.; YAGUE, A. A framework to aid in decision making for technical debt management. In: **IEEE 7th International Workshop On Managing Technical Debt (Mtd)**. Nova Iorque, Estados Unidos: [S. n.], 2015. p. 69–76.
- FREZZA, S. T. Issues in student valuing of software engineering best practices. In: **2016 IEEE Frontiers in Education Conference (FIE)**. Nova Iorque, Estados Unidos: [S. n.], 2016. p. 1–4.

- GUO, Y.; SEAMAN, C.; GOMES, R.; CAVALCANTI, A.; TONIN, G.; SILVA, F. Q. B. D.; SANTOS, A. L. M.; SIEBRA, C. **Tracking technical debt — An exploratory case study**. 2011. 528-531 p.
- GUO, Y.; SEAMAN, C.; SILVA, F. Q. da. Costs and obstacles encountered in technical debt management – a case study. **Journal of Systems and Software**, v. 120, p. 156–169, 2016.
- GUO, Y.; SPINOLA, R. O.; SEAMAN, C. Exploring the costs of technical debt management - a case study. **Empirical Softw. Engg.**, Kluwer Academic Publishers, v. 21, p. 159–182, 2016.
- HOLVITIE, J.; LICORISH, S. A.; SPÍNOLA, R. O.; HYRYNSALMI, S.; MACDONELL, S. G.; MENDES, T. S.; BUCHAN, J. Technical debt and agile software development practices and processes: An industry practitioner survey. **Information and Software Technology**, v. 96, p. 141–160, 2018.
- KOLES, B.; NAGY, P. Virtual worlds as digital workplaces conceptualizing the affordances of virtual worlds to expand the social and professional spheres in organizations. **Organizational Psychology Review**, v. 4, p. 175–195, 2014.
- KRUCHTEN, P.; NORD, R. L.; OZKAYA, I. Technical debt: From metaphor to theory and practice. **IEEE Software**, v. 29, n. 6, p. 18–21, 2012.
- LAVAZZA, L.; MORASCA, S.; TOSI, D. **Technical Debt as an External Software Attribute**. 2018. 21-30 p.
- LEITÃO, C. F.; PRATES, R. O. A aplicação de métodos qualitativos em computação. In: **Jornadas de Atualização em Informática**. [S. l.]: SBC, 2017.
- MARIJAN, D.; SEN, S. Good practices in aligning software engineering research and industry practice. **SIGSOFT Softw. Eng. Notes**, Association for Computing Machinery, v. 44, n. 3, p. 65–67, 2020.
- MISHRA, A.; CAGILTAY, N.; KILIC, O. Software engineering education: Some important dimensions. **European Journal of Engineering Education**, v. 32, p. 349–361, 2007.
- NBR ISO/IEC 9126-1. **Engenharia de software - Qualidade de produto**. Rio de Janeiro, Brasil, 2003.
- OLIVEIRA, F.; GOLDMAN, A.; SANTOS, V. Managing technical debt in software projects using scrum: An action research. In: **Agile Conference**. Nova Iorque, Estados Unidos: [S. n.], 2015. p. 50–59.
- PMI. **Um guia do conhecimento em gerenciamento de projetos (Guia PMBOK)**. 6. ed. Newton Square, Pensilvânia, Estados Unidos: Project Management Institute, 2017.
- RAMASUBBU, N.; KEMERER, C. F. Integrating technical debt management and software quality management processes: A normative framework and field tests. **IEEE Transactions on Software Engineering**, p. 285–300, 2019.
- RIOS, N.; SPINOLA, R. O.; MENDONÇA, M.; SEAMAN, C. The most common causes and effects of technical debt: First results from a global family of industrial surveys. In: **Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement**. [S. l.]: Association for Computing Machinery, 2018.

- RIOS, N.; SPINOLA, R. O.; NETO de M. G. M.; SEAMAN, C. A study of factors that lead development teams to incur technical debt in software projects. In: **2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)**. [S. l.: s. n.], 2018. p. 429–436.
- ROCHA, J. C.; ZAPALOWSKI, V.; NUNES, I. Understanding technical debt at the code level from the perspective of software developers. In: **Proceedings of the XXXI Brazilian Symposium on Software Engineering**. [S. l.: s. n.], 2017. p. 64–73.
- SBROCCO, J. H. T. de C.; MACEDO, P. C. de. **Metodologias Ágeis - Engenharia de Software sob Medida**. [S. l.]: Editora Saraiva, 2012.
- STOCHEL, M. G.; CHOLDA, P.; WAWROWSKI, M. R. On coherence in technical debt research : Awareness of the risks stemming from the metaphorical origin and relevant remediation strategies. In: **Euromicro Conference On Software Engineering And Advanced Applications (Seaa)**. Nova Iorque, Estados Unidos: [S. n.], 2020. p. 367–375.
- TALLURI, M.; HADDAD, H. M. Best managerial practices in agile development. In: **Proceedings of the 2014 ACM Southeast Regional Conference**. New York, NY, USA: Association for Computing Machinery, 2014.
- VALENTE, M. T. **Engenharia de Software Moderna: Princípios e Práticas para Desenvolvimento de Software com Produtividade**. [S. l.]: Editora: Independente, 2020.
- VIJAYASARATHY, L.; BUTLER, C. W. Choice of software development methodologies: Do organizational, project, and team characteristics matter? **IEEE Software**, v. 33, n. 5, p. 86–94, 2016.
- YLI-HUUMO, J.; MAGLYAS, A.; SMOLANDER, K. How do software development teams manage technical debt? – an empirical study. **Journal of Systems and Software**, v. 120, p. 195–218, 2016.
- ZAZWORKA, N.; SHAW, M. A.; SHULL, F.; SEAMAN, C. Investigating the impact of design debt on software quality. In: **Proceedings of the 2nd Workshop on Managing Technical Debt**. [S. l.]: Association for Computing Machinery, 2011. p. 17–23.

## APÊNDICE A – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO (TCLE)

### Termo de Consentimento Livre e Esclarecido (TCLE)

Prezado(a) participante, você está sendo convidado(a) a participar de uma pesquisa de Trabalho de Conclusão de Curso (TCC) cujo objetivo é entender como os débitos técnicos impactam os projetos de *software*, os desafios apresentados no desenvolvimento do *software* e compreender o gerenciamento de débitos técnicos nos projetos de *software*. Vale lembrar que, esta pesquisa é destinada a profissionais das áreas relacionadas a arquitetura de *software*, desenvolvimento e qualidade de *software*.

**Confidencialidade e Anonimato:** Todas as informações fornecidas por você serão tratadas de forma estritamente confidencial. Não será coletado seu nome ou quaisquer informações pessoais, exceto se queira fornecê-las. Caso as forneça, as mesmas serão mantidas em sigilo e não serão divulgadas em nenhum relatório ou resultado da pesquisa. Além disso, as respostas serão tratadas de maneira agregada, garantindo o anonimato dos participantes.

**Participação Voluntária:** Sua participação neste estudo é voluntária. Você tem o direito de recusar-se a participar, de desistir a qualquer momento durante a pesquisa ou optar por não responder a determinadas perguntas.

**Benefícios:** Ao participar desta pesquisa, você estará contribuindo para uma pesquisa que busca compreender os efeitos dos débitos técnicos em projetos de *software*, além de contribuir para a construção de um artefato que auxilia no gerenciamento de débitos técnicos.

### Riscos:

- a) Toda pesquisa realizada com seres humanos apresenta riscos. No entanto, os riscos apresentados nesta pesquisa são mínimos, visto que o questionário utilizado não altera aspectos fisiológicos, psicológicos ou sociais dos participantes.
- b) Os dados fornecidos serão tratados com confidencialidade e não serão compartilhados com terceiros externos. Todas as medidas de segurança necessárias serão tomadas para proteger a privacidade e a integridade dos dados.
- c) Os dados coletados durante o estudo destinam-se estritamente às atividades da pesquisa. Desta forma, não serão utilizados para qualquer forma de avaliação profissional ou pessoal.
- d) O participante não terá de arcar com despesas, nem gastos, não havendo nenhuma forma possível para isso.

**Consentimento e Autorização:** Ao responder este termo de consentimento com

"Sim", você confirma que leu e compreendeu as informações acima e concorda em participar da pesquisa. Você autoriza o uso das informações fornecidas por você exclusivamente para fins de pesquisa acadêmica.

**Tempo de resposta:** Dependendo do contexto de respostas, para responder esse formulário você levará em média 5-8 minutos.

Caso você tenha alguma dúvida ou queira mais informações sobre o estudo, sinta-se à vontade para entrar em contato com o pesquisador responsável.

Atenciosamente,

Pesquisadores Responsáveis Ewaldo Junior ewaldojunior@alu.ufc.br

Profa. Diana Braga diana@ufc.br

Caso você concorde em participar da pesquisa, por favor, responda "Sim" para a pergunta logo abaixo.

Você aceita participar desta pesquisa após ler o termo de consentimento?

Sim  Não

## APÊNDICE B – QUESTIONÁRIO PILOTO

### PERFIL DO PARTICIPANTE

Nesta seção será coletado algumas informações sobre o seu perfil como profissional de tecnologia. Esses dados são importantes para compreender melhor o contexto em que os débitos técnicos são gerenciados em projetos de *software* e como as práticas variam de acordo com diferentes características dos profissionais.

1 - Qual seu grau de formação? (Caso o grau atual esteja em andamento, selecione o seu grau anterior)

- Técnico/Tecnólogo
- Bacharel
- Mestre
- Doutor
- Não possuo
- Outro.

2 - Qual seu curso de formação?

- Análise e Desenvolvimento de Sistemas
- Ciência da Computação
- Design* Digital
- Engenharia da Computação
- Engenharia de *Software*
- Redes de Computadores
- Sistemas de Informação
- Sistemas e Mídias Digitais
- Não possuo
- Outro.

3 - Seu cargo atual é em qual das seguintes áreas da Tecnologia de Informação?

- Ciência de dados
- Desenvolvimento

- Design*
- Gerência de projetos
- Infraestrutura
- Produtos / Requisitos
- Qualidade de *Software*
- Segurança
- Outro.

4 - Qual sua faixa etária?

- Menos de 25 anos
- entre 26 à 30 anos
- entre 31 à 35 anos
- entre 36 à 40 anos
- entre 41 à 45 anos
- entre 46 à 50 anos
- Mais de 50 anos

5 - Quanto tempo de experiência você possui na área de tecnologia?

- Menos de 1 ano
- 1 à 3 anos
- 4 à 6 anos
- 7 à 10 anos
- Mais de 10 anos

### **PERFIL DO AMBIENTE DE TRABALHO DO PARTICIPANTE**

O objetivo desta seção é de conhecer melhor o ambiente de trabalho em que o participante está inserido. Entender as características do ambiente em que os profissionais de tecnologia atuam é fundamental para analisar os fatores que influenciam a gestão de débitos técnicos e identificar possíveis desafios e oportunidades.

6 - Qual é o tamanho da equipe em que você trabalha?

- Menos de 5 pessoas
- 5 à 10 pessoas
- 11 à 15 pessoas
- 16 à 20 pessoas
- Mais de 20 pessoas

7 - Qual seu regime de trabalho?

- Presencial
- Remoto
- Híbrido

8 - Qual metodologia de gerenciamento de projetos de *software* você utiliza atualmente?

- Metodologia preditiva (Cascata)
- Metodologia ágil (*Scrum, Kanban e Extreme Programming*)
- Metodologia estratégica (Gestão do portfólio)
- Outro.

## **ENTENDIMENTO ACERCA DE DÉBITOS TÉCNICOS EM PROJETOS DE SOFTWARE**

Débitos técnicos / dívidas técnicas / *techdebt* em projetos de *software* são problemas ou deficiências encontrados no código-fonte, na arquitetura ou em outros aspectos técnicos de um *software*. Eles surgem devido a escolhas de projeto questionáveis, prazos apertados, falta de documentação adequada ou outras razões. Os débitos técnicos podem incluir código mal estruturado, falta de testes adequados, dependências desatualizadas, baixa manutenibilidade do *software*, entre outros. Esses débitos técnicos podem impactar negativamente a qualidade do *software*, levando a problemas como maior complexidade, dificuldade em adicionar novas funcionalidades, aumento de erros e maior dificuldade em realizar manutenção. Ignorar ou acumular débitos técnicos ao longo do tempo pode resultar fragilidade de manter a aplicação.

9 - Você está familiarizado com o conceito de débitos técnicos / dívidas técnicas /

*techdebt*?

Sim

Não

10 - Com que frequência você identifica débitos técnicos em projetos de *software*?

Nunca      Sempre

11 - Você já participou de projetos em que os débitos técnicos foram considerados e tratados como parte do processo de desenvolvimento?

Nunca      Sempre

12 - Quais são as principais causas que você acredita que geram débitos técnicos a partir do contexto do seu projeto? (Selecione as opções que você acredita que se aplique).

Documentação inadequada

Escopo mal definido

Falta de conscientização da equipe

Falta de recursos

Falta de suporte da alta gestão

Falta de tempo

Mudança técnica

Pressão para cumprir prazos

Usuários resistentes

Outro.

13 - Em sua opinião, quais são os atributos de qualidade mais afetados por débitos técnicos em projetos de *software*? (Selecione as opções que você acredita que se aplique).

Confiabilidade

Eficiência

Funcionalidade

Manutenibilidade

Portabilidade

Usabilidade

Outro.

14 - Para você, qual a importância de uma cultura de qualidade e boas práticas no tratamento de débitos técnicos em projetos de *software*?

Sem importância      Muito importante

15 - Qual(is) é(são) o(s) principal(is) desafio(s) ao lidar com débitos técnicos em projetos de *software*?

Resposta aberta.

16 - Como você classificaria a importância de gerenciar débitos técnicos em projetos de *software*?

Sem importância      Muito importante

17 - Os débitos técnicos são gerenciados no projeto que você trabalha?

Sim

Não

Não sei

## **ENTENDIMENTO ACERCA DO GERENCIAMENTO DE DÉBITOS TÉCNICOS**

Nesta seção busca-se compreender como o time do participante trabalha para gerenciar os débitos técnicos em seus projetos de *software*. Essa seção pretende explorar as práticas adotadas, a existência de processos formais, a utilização de artefatos específicos e a abordagem para priorização e resolução desses débitos.

18 - No seu projeto são utilizadas métricas ou indicadores para monitorar a quantidade ou a gravidade dos débitos técnicos ao longo do projeto?

Sim

Não

Não tenho certeza

19 - Qual é o nível de envolvimento das partes interessadas (clientes, usuários, gerentes, etc.) no processo de gerenciamento de débitos técnicos?

Não são envolvidos ( ) ( ) ( ) ( ) ( ) Muito envolvidos

20 - Com que frequência as atividades de gerenciamento de débitos técnicos são realizadas no seu projeto de *software*?

( ) Diariamente

( ) Semanalmente

( ) Mensalmente

( ) Em marcos específicos do projeto

( ) Quando necessário

( ) Não realizamos atividades voltadas para débitos técnicos, deixamos do jeito que fica

21 - Em seu projeto de *software* atual, como o time lida com a identificação e o gerenciamento de débitos técnicos?

Resposta aberta.

22 - O time possui algum artefato específico para organizar e rastrear débitos técnicos, como uma lista de pendências, backlog técnico, framework ou algo similar? Caso possua algum artefato, informe qual artefato é utilizado.

Resposta aberta.

23 - Que melhorias ou sugestões você proporia para aprimorar o gerenciamento de débitos técnicos no seu projeto?

Resposta aberta.

## **ENTENDIMENTO ACERCA DA AUSÊNCIA DE GERENCIAMENTO DE DÉBITOS TÉCNICOS**

Essa seção faz-se relevante compreender as razões pelas quais o time do participante não realiza o gerenciamento de débitos técnicos em seus projetos de *software*. Visando explorar

os processos adotados, a existência de justificativas específicas, a falta de recursos ou tempo adequados, bem como os possíveis impactos negativos decorrentes dessa ausência.

24 - Você acredita que existem processos eficazes para evitar a acumulação de débitos técnicos?

- Sim
- Não
- Não tenho certeza

25 - Quais são os principais processos ou práticas adotados pelo seu projeto em relação ao monitoramento e rastreabilidade dos débitos técnicos?

Resposta aberta.

26 - Na sua opinião, quais seriam os benefícios ou vantagens de implementar práticas de gerenciamento de débitos técnicos?

Resposta aberta.

27 - Seu projeto já enfrentou consequências negativas decorrentes da ausência do gerenciamento de débitos técnicos? Se sim, por favor, descreva.

Resposta aberta.

28 - Você acredita que a existência de um artefato capaz de auxiliar no processo de gerenciamento de débitos técnicos seria relevante para utilizar no seu projeto?

Não é relevante      Muito relevante

## APÊNDICE C – QUESTIONÁRIO FINAL

### PERFIL DO PARTICIPANTE

Nesta seção será coletado algumas informações sobre o seu perfil como profissional de tecnologia. Esses dados são importantes para compreender melhor o contexto em que os débitos técnicos são gerenciados em projetos de *software* e como as práticas variam de acordo com diferentes características dos profissionais.

1 - Qual seu grau de formação? Caso o grau atual esteja em andamento, selecione o seu grau anterior.

- Ensino médio completo
- Técnico/Tecnólogo
- Bacharel
- Mestre
- Doutor
- Outro

2 - Qual seu curso de formação?

- Análise e Desenvolvimento de Sistemas
- Ciência da Computação
- Design* Digital
- Engenharia da Computação
- Engenharia de *Software*
- Redes de Computadores
- Sistemas de Informação
- Sistemas e Mídias Digitais
- Não possuo
- Outro.

3 - Seu cargo atual é em qual da seguinte área da Tecnologia de Informação?

- Desenvolvimento de *software*
- Gerenciamento de projetos/produtos

4 - Qual sua faixa etária?

- Menos de 25 anos
- entre 26 à 30 anos
- entre 31 à 35 anos
- entre 36 à 40 anos
- entre 41 à 45 anos
- entre 46 à 50 anos
- Mais de 50 anos

5 - Quanto tempo de experiência você possui na área de tecnologia?

- Menos de 1 ano
- 1 à 3 anos
- 4 à 6 anos
- 7 à 10 anos
- Mais de 10 anos

### **PERFIL DO AMBIENTE DE TRABALHO DO PARTICIPANTE**

O objetivo desta seção é de conhecer melhor o ambiente de trabalho em que o participante está inserido. Entender as características do ambiente em que os profissionais de tecnologia atuam é fundamental para analisar os fatores que influenciam a gestão de débitos técnicos e identificar possíveis desafios e oportunidades.

6 - Qual é o tamanho da equipe em que você trabalha?

- Menos de 5 pessoas
- 5 à 10 pessoas
- 11 à 15 pessoas
- 16 à 20 pessoas
- Mais de 20 pessoas

7 - Qual seu regime de trabalho?

- Presencial
- Remoto
- Híbrido

8 - Qual metodologia de gerenciamento de projetos de *software* você utiliza atualmente?

- Metodologia preditiva (Cascata, etc)
- Metodologia ágil (*Scrum, Kanban, Lean, Extreme Programming, etc*)
- Outro.

### **ENTENDIMENTO ACERCA DE DÉBITOS TÉCNICOS EM PROJETOS DE SOFTWARE**

Débitos técnicos, dívidas técnicas ou *techdebt* em projetos de *software* são problemas, ou deficiências encontrados no código-fonte, na arquitetura ou em outros aspectos técnicos de um *software*. Eles surgem devido a escolhas de projeto questionáveis, prazos apertados, falta de documentação adequada ou outras razões. Os débitos técnicos podem incluir código mal estruturado, falta de testes adequados, dependências desatualizadas, baixa manutenibilidade do *software*, entre outros.

Esses débitos técnicos podem impactar negativamente a qualidade do *software*, levando a problemas como maior complexidade, dificuldade em adicionar novas funcionalidades, aumento de erros e maior dificuldade em realizar manutenção. Ignorar ou acumular débitos técnicos ao longo do tempo pode resultar fragilidade de manter a aplicação.

9 - Depois desta contextualização informada acima, você tem familiaridade com o conceito de débitos técnicos / dívidas técnicas / *techdebt* no cenário do seu projeto?

- Sim  Não

### **ENTENDIMENTO ACERCA DE DÉBITOS TÉCNICOS EM PROJETOS DE SOFTWARE (parte 2)**

Nesta seção, faz-se pertinente compreender o entendimento sobre débitos técnicos em projetos de *software*. Os débitos técnicos são problemas ou deficiências encontrados no código-fonte, na arquitetura ou em outros aspectos técnicos de um *software*, eles podem surgir devido a escolhas de projeto. Entender como os profissionais de tecnologia percebem e lidam com os débitos técnicos é essencial para aprimorar a qualidade dos projetos de *software* e tomar decisões informadas sobre o gerenciamento desses débitos.

10 - Com que frequência débitos técnicos são identificados na sua equipe?

- Em todas as iterações ou sprints

- Regularmente, mas não em todas as iterações ou sprints
- Ocasionalmente, apenas quando problemas são graves
- Raramente, em casos excepcionais
- Nunca são identificados
- Não tenho certeza
- Outro.

11 - Quando um débito técnico é identificado em um projeto, como a equipe é informada sobre sua existência. Selecione a(s) opção(ões) que você acredita que se aplique.

- Em reuniões de equipes regulares
- Através de comunicação escrita (e-mail, chat, etc.)
- Utilizando ferramentas de rastreamento de problemas ou tarefas (Jira, Trello, GitHub Issues, etc.)
- Outro.

12 - Quais são as principais causas que você acredita que geram débitos técnicos a partir do contexto do seu projeto? Selecione a(s) opção(ões) que você acredita que se aplique.

- Documentação inadequada
- Escopo mal definido
- Falta de comunicação
- Falta de conscientização da equipe
- Falta de entendimento das dependências de atividades
- Falta de recursos
- Falta de suporte da alta gestão
- Falta de tempo
- Falta de visão geral do projeto
- Mudança técnica
- Pressão para cumprir prazos
- Usuários resistentes
- Outro.

13 - Em sua opinião, quais são os atributos de qualidade mais afetados por débitos

técnicos em projetos de *software*? (Selecione as opções que você acredita que se aplique).

- Confiabilidade (refere-se à sua capacidade de executar suas funções de maneira consistente e previsível, sem erros críticos ou falhas, garantindo a disponibilidade, integridade e segurança dos dados e operações.)
- Eficiência (diz respeito à capacidade do *software* de realizar suas tarefas de forma otimizada, utilizando recursos mínimos, como tempo de processamento, memória e largura de banda, para entregar resultados rápidos e eficazes.)
- Funcionalidade (avalia a capacidade do *software* em fornecer um conjunto de funcionalidades e características que atendam aos requisitos especificados, garantindo que o *software* realize as tarefas e operações esperadas de acordo com as necessidades do usuário.)
- Manutenibilidade (refere-se à facilidade de modificar, corrigir, aprimorar e adaptar o *software* ao longo do tempo. Um *software* altamente manutenível é aquele em que as alterações podem ser feitas com facilidade, minimizando riscos de introdução de erros.)
- Portabilidade (avalia a capacidade do *software* de ser transferido de um ambiente para outro, como diferentes sistemas operacionais ou plataformas. Um *software* portátil é aquele que mantém sua funcionalidade e desempenho em diferentes ambientes.)
- Usabilidade (diz respeito à facilidade de uso do *software*, permitindo que os usuários interajam com ele de forma intuitiva e eficaz. Um *software* com alta usabilidade oferece uma interface amigável e experiência agradável ao usuário, tornando as tarefas mais simples de serem realizadas.)
- Outro.

14 - Para você, qual a importância de uma cultura de qualidade e boas práticas no tratamento de débitos técnicos em projetos de *software*?

- Não é importante
- Pouco importante
- Tanto faz
- Importante
- Muito importante

15 - Do seu ponto de vista, qual(is) é(são) o(s) principal(is) desafio(s) ao lidar com débitos técnicos?

Resposta aberta.

16 - Como você classificaria a importância de gerenciar débitos técnicos em projetos de *software*?

- Não é importante
- Pouco importante
- Tanto faz
- Importante
- Muito importante

Alguns exemplos de artefatos e práticas que projetos de *software* utilizam para auxiliar no gerenciamento de débitos técnicos:

**Registro de Débitos Técnicos:** Um documento ou sistema de registro onde a equipe pode documentar e acompanhar os débitos técnicos identificados. Isso pode incluir uma descrição do débito, seu impacto, contexto e possível solução. Ferramentas como planilhas, sistemas de rastreamento de problemas ou ferramentas específicas para gerenciamento de débitos técnicos podem ser usadas.

**Backlog de Débitos Técnicos:** Um componente do backlog geral do projeto que é dedicado exclusivamente aos débitos técnicos. Cada débito é registrado como um item de backlog, permitindo que a equipe priorize e trate-os de acordo com sua importância e urgência.

**Histórias ou Tarefas de Refatoração:** Incluir refatorações como histórias ou tarefas no backlog de desenvolvimento.

**Documentação de Arquitetura e *Design*:** Manutenção de documentação atualizada sobre a arquitetura, *design* e decisões técnicas do projeto.

**Revisões de Código:** Incorporação de revisões de código regulares no processo de desenvolvimento. Isso permite que os membros da equipe identifiquem e resolvam débitos técnicos antes que se tornem um problema maior.

**Matriz de Risco Técnico:** Uma matriz que avalia os débitos técnicos em termos de impacto e urgência. Isso ajuda a priorizar ações de tratamento de débitos técnicos com base no seu potencial risco.

**Indicadores de Saúde do Projeto:** Desenvolvimento de indicadores ou métricas que refletem a saúde geral do projeto, incluindo a quantidade e o impacto dos débitos técnicos. Isso

fornece uma visão quantitativa do progresso na redução de débitos. Essas métricas ou indicadores podem ser: taxa de cobertura de testes, número de violações de padrões de codificação, número de problemas de desempenho reportados e taxa de débito técnico acumulada.

**Relatórios de Avaliação de Débitos Técnicos:** Periodicamente, a equipe pode produzir relatórios que resumem a quantidade, a situação e as áreas problemáticas dos débitos técnicos no projeto. Esses relatórios são úteis para comunicação com stakeholders e para acompanhar a evolução ao longo do tempo.

17 - Os débitos técnicos são gerenciados no projeto que você trabalha?

- Sim
- Não
- Não sei

## **ENTENDIMENTO ACERCA DO GERENCIAMENTO DE DÉBITOS TÉCNICOS**

Nesta seção busca-se compreender como o time do participante trabalha para gerenciar os débitos técnicos em seus projetos de *software*. Essa seção pretende explorar as práticas adotadas, a existência de processos formais, a utilização de artefatos específicos e a abordagem para priorização e resolução desses débitos.

18 - Com base nos artefatos e práticas mencionados anteriormente, informe qual(is) é(são) utilizado(s) na sua equipe para organizar e rastrear débitos técnicos. Caso não seja utilizado nenhum deles, registre na opção "Outros".

- Registro de Débitos Técnicos
- Backlog* de Débitos Técnicos
- Histórias ou Tarefas de Refatoração
- Documentação de Arquitetura e *Design*
- Revisões de Código
- Matriz de Risco Técnico
- Indicadores de saúde do projeto
- Relatórios de Avaliação de Débitos Técnicos
- Outro.

19 - Qual é o nível de envolvimento dos clientes / usuários quando é abordado tópicos relacionados a débitos técnicos no ciclo de vida do projeto?

- Não são envolvidos
- Pouco envolvidos
- Tanto faz
- Envolvidos
- Muito envolvidos

20 - Com que frequência as atividades de gerenciamento de débitos técnicos são realizadas no seu projeto de *software*?

- Diariamente
- Semanalmente
- Mensalmente
- Em marcos específicos do projeto
- Quando necessário
- Não realizamos atividades voltadas para débitos técnicos, deixamos do jeito que fica

21 - O que você considera como boas práticas para o gerenciamento de débitos técnicos?

Resposta aberta.

22 - Que melhorias ou sugestões você proporia para aprimorar o gerenciamento de débitos técnicos no seu projeto?

Resposta aberta.

### **ENTENDIMENTO ACERCA DA AUSÊNCIA DE GERENCIAMENTO DE DÉBITOS TÉCNICOS**

Essa seção faz-se relevante compreender as razões pelas quais o time do participante não realiza o gerenciamento de débitos técnicos em seus projetos de *software*. Visando explorar os processos adotados, a existência de justificativas específicas, a falta de recursos ou tempo adequados, bem como os possíveis impactos negativos decorrentes dessa ausência.

23 - Existem processos eficazes para evitar a acumulação de débitos técnicos no projeto que você atua?

- Sim
- Não
- Não tenho certeza

24 - Quais são as práticas adotadas pelo seu projeto em relação ao monitoramento e rastreabilidade dos débitos técnicos?

Resposta aberta.

25 - Na sua opinião, quais seriam os benefícios ou vantagens de implementar boas práticas de gerenciamento de débitos técnicos? Selecione a(s) opção(ões) que você acredita que se aplique.

- Maior qualidade do *software*
- Redução de custos de manutenção
- Aumento da produtividade da equipe
- Menos problemas de escalabilidade
- Maior satisfação do cliente
- Outro.

26 - Seu projeto já enfrentou consequências negativas decorrentes da ausência do gerenciamento de débitos técnicos? Se sim, por favor, descreva.

Resposta aberta.

27 - Qual a justificativa ou razão específica para a ausência do gerenciamento de débitos técnicos no time?

Resposta aberta.

28 - Você acredita que a existência de um artefato capaz de auxiliar no processo de gerenciamento de débitos técnicos seria relevante para utilizar no seu projeto?

- Não é relevante
- Pouco relevante

- Tanto faz
- Relevante
- Muito relevante

## APÊNDICE D – JUSTIFICATIVA PARA AS QUESTÕES DO QUESTIONÁRIO FINAL

A seção "Perfil do Participante" desempenha um papel fundamental na coleta de informações relevantes para a compreensão dos débitos técnicos e das práticas de gerenciamento em projetos de *software*. As respostas a essas questões fornecem percepções valiosas sobre os participantes do estudo, permitindo uma análise mais aprofundada das práticas de gerenciamento de débitos técnicos de acordo com diferentes características dos profissionais de tecnologia. A seção é projetada para capturar essas nuances, fornecendo dados cruciais que permitem uma análise diferenciada e contextualizada.

A fim de embasar as escolhas das perguntas do questionário, foram selecionados alguns estudos que validam a necessidade de coletar estas informações. Em Mishra *et al.* (2007), é destacado as questões educacionais como relevantes nas práticas de engenharia de *software* e habilidades críticas de pensamento vitais para a tomada de decisões estratégicas em projetos de *software*. O estudo de Ernst *et al.* (2020) fornece um quadro para entender as interações entre os aspectos sociais e técnicos da engenharia de *software*, de modo que, a compreensão das dinâmicas socio-técnicas permite uma avaliação mais precisa de como as características individuais dos profissionais afetam as práticas e processos no contexto de projetos de *software*. Nesse sentido, a experiência profissional e a pesquisa pessoal dos participantes influenciam suas abordagens, por isso que as respostas fornecidas pelos participantes oferecem uma importante base de dados que, quando analisados, podem revelar correlações significativas e padrões de práticas.

Perguntas:

"Qual seu grau de formação?"

Esta pergunta busca identificar o nível de formação acadêmica dos participantes, o que é importante para entender como as práticas de gerenciamento de débitos técnicos variam entre profissionais com diferentes níveis de educação formal. A resposta a essa pergunta ajudará a determinar se a formação acadêmica influencia as abordagens de gerenciamento de débitos técnicos.

"Qual seu curso de formação?"

A escolha do curso de formação dos participantes é relevante para avaliar se profissionais com diferentes formações acadêmicas têm perspectivas distintas sobre o gerenciamento de débitos técnicos. Conhecer o curso de formação dos participantes é fundamental para identificar qualquer

correlação entre a formação específica e as práticas de gerenciamento de débitos técnicos.

"Seu cargo atual é em qual da seguinte área da Tecnologia de Informação?"

Esta pergunta permite categorizar os participantes conforme a área da Tecnologia de Informação em que atuam. A distinção entre "Desenvolvimento de *software*" e "Gerenciamento de projetos/produtos" é crucial para avaliar se as práticas de gerenciamento de débitos técnicos variam significativamente entre essas áreas de atuação.

"Qual sua faixa etária?"

A faixa etária dos participantes fornece informações valiosas sobre a distribuição etária dos profissionais de tecnologia envolvidos no estudo. Isso é importante para entender se a idade está relacionada a diferentes abordagens de gerenciamento de débitos técnicos, pois experiência e perspectivas podem variar consideravelmente conforme a faixa etária.

"Quanto tempo de experiência você possui na área de tecnologia?"

A experiência profissional dos participantes é um fator relevante para avaliar como as práticas de gerenciamento de débitos técnicos evoluem ao longo do tempo. Essa pergunta permite identificar se a experiência profissional está associada a diferentes estratégias de gerenciamento de débitos técnicos, considerando a diversidade de níveis de experiência.

A seção "Perfil do Ambiente de Trabalho do Participante" desempenha um papel crucial na obtenção de informações essenciais sobre o ambiente em que os profissionais de tecnologia atuam. Esses dados são fundamentais para a análise dos fatores que influenciam a gestão de débitos técnicos e a identificação de desafios e oportunidades associados a esse contexto.

Esta seção atua como uma ferramenta valiosa para entender as nuances do local onde os profissionais de tecnologia desempenham suas funções. Além disso, pode revelar como o ambiente de trabalho influencia a gestão de débitos técnicos e como ele pode ser otimizado para enfrentar desafios e aproveitar oportunidades. Vijayarathy e Butler (2016) investiga a interação entre as características organizacionais, de projetos e de equipes na seleção de metodologias de desenvolvimento de *software*, destacando a cultura organizacional, natureza do projeto, as experiências da equipe e oportunidades de melhoria. Ademais, em Koles e Nagy (2014) é explorado como os mundos virtuais podem ser utilizados como espaços de trabalho digitais, visando entender como as interações sociais e profissionais são transformadas em ambientes virtuais.

Perguntas:

"Qual é o tamanho da equipe em que você trabalha?"

Esta pergunta visa determinar o tamanho da equipe de trabalho do participante. A informação é relevante para entender como o tamanho da equipe pode afetar a gestão de débitos técnicos, uma vez que equipes maiores ou menores podem enfrentar desafios diferentes em termos de comunicação, coordenação e responsabilidade.

"Qual seu regime de trabalho?"

A escolha entre trabalho presencial, remoto ou híbrido é essencial para avaliar como as condições de trabalho impactam a gestão de débitos técnicos. Isso é particularmente relevante no contexto atual, em que muitos profissionais tiveram que se adaptar a diferentes modelos de trabalho devido à pandemia, o que pode influenciar suas práticas de gerenciamento de débitos técnicos.

"Qual metodologia de gerenciamento de projetos de *software* você utiliza atualmente?"

A metodologia de gerenciamento de projetos de *software* adotada pelos participantes é uma informação crucial para entender as abordagens e práticas de gerenciamento de débitos técnicos. A distinção entre metodologias preditivas e ágeis permite analisar se diferentes metodologias influenciam as estratégias de gerenciamento de débitos técnicos, já que cada abordagem pode enfatizar aspectos distintos do desenvolvimento de *software*.

A seção "Entendimento Acerca de Débitos Técnicos em Projetos de *software*" tem o objetivo de avaliar o conhecimento e a compreensão dos participantes em relação ao conceito de débitos técnicos. Essa contextualização prévia é essencial para enquadrar as respostas subsequentes dos participantes e entender se estão cientes dos desafios associados a essas dívidas técnicas no contexto de seus projetos. Além disso, busca obter uma compreensão mais detalhada sobre a percepção e a gestão de débitos técnicos por parte dos profissionais de tecnologia. As perguntas desta seção são fundamentais para aprofundar a análise desse conceito no contexto dos projetos de *software*.

A compreensão dos participantes sobre débitos técnicos pode variar significativamente, influenciando diretamente a maneira como eles identificam, priorizam e resolvem essas questões. Em Zazworka *et al.* (2011) é especificado o impacto na qualidade do *software* e o entendimento dos efeitos a longo prazo que as decisões de construção do *software* podem ter na qualidade e na manutenibilidade. Associado a isso, Lavazza *et al.* (2018) sugere que o

débito técnico pode ser considerado um atributo externo do *software* que afeta a conservação do *software*, de modo que, perguntas baseadas neste artigo podem explorar como as equipes percebem o débito técnico em relação a outros atributos de qualidade do *software* e como isso influencia suas decisões de gerenciamento.

Concomitantemente, no estudo de Guo *et al.* (2011) é fornecido uma visão prática do impacto do débito técnico em um projeto de *software* real, inspirando como as equipes rastreiam e priorizam o débito técnico ao longo do ciclo de vida do *software*. E em Falessi e Kruchten (2015) é argumentado a importância de educar futuros engenheiros de *software* sobre débitos técnicos, associando questões de rastreabilidade, conhecimento da área e identificação. Estes estudos agem como peças fundamentais de embasamento para avaliar se o conhecimento dos participantes sobre débitos técnicos é formal e estruturado, possivelmente adquirido através da educação formal, ou se é baseado em experiência prática.

Perguntas:

"Com que frequência débitos técnicos são identificados na sua equipe?"

Esta pergunta é essencial para avaliar a frequência com que os débitos técnicos são reconhecidos no ambiente de trabalho. As respostas podem ajudar a entender se os débitos técnicos são uma preocupação constante, ou se são identificados apenas em situações excepcionais, o que pode influenciar a abordagem de gerenciamento.

"Quando um débito técnico é identificado em um projeto, como a equipe é informada sobre sua existência?"

A maneira como a equipe é informada sobre a existência de débitos técnicos é relevante para entender os processos de comunicação e as práticas de transparência dentro do projeto. Isso é fundamental para garantir que os débitos técnicos sejam tratados de maneira eficaz.

"Quais são as principais causas que você acredita que geram débitos técnicos a partir do contexto do seu projeto?"

Esta pergunta visa identificar as causas percebidas para a ocorrência de débitos técnicos em projetos específicos. A compreensão dessas causas ajudará a identificar áreas de melhoria e a implementar estratégias eficazes de prevenção.

"Em sua opinião, quais são os atributos de qualidade mais afetados por débitos técnicos em projetos de *software*?"

Esta pergunta permite que os participantes identifiquem os atributos de qualidade mais impactados por débitos técnicos, o que é essencial para entender e priorizar ações de mitigação a fim de

facilitar o tratamento de débitos técnicos.

"Para você, qual a importância de uma cultura de qualidade e boas práticas no tratamento de débitos técnicos em projetos de *software*?"

Esta pergunta avalia o reconhecimento da importância de boas práticas e de uma cultura de qualidade na gestão de débitos técnicos, o que é fundamental para promover uma abordagem eficaz na resolução desses problemas.

"Do seu ponto de vista, qual(is) é(são) o(s) principal(is) desafio(s) ao lidar com débitos técnicos?"

Esta pergunta abre espaço para os participantes compartilharem seus desafios e preocupações específicos relacionados à gestão de débitos técnicos, fornecendo informações valiosas para melhorar a abordagem da equipe em relação a esses problemas.

"Como você classificaria a importância de gerenciar débitos técnicos em projetos de *software*?"

Esta pergunta permite que os participantes expressem sua opinião sobre a importância da gestão de débitos técnicos, o que ajuda a entender sua visão sobre o impacto desses débitos no sucesso do projeto.

"Os débitos técnicos são gerenciados no projeto que você trabalha?"

Esta pergunta busca determinar se a gestão de débitos técnicos é uma prática implementada no ambiente de trabalho, o que é fundamental para avaliar a eficácia do tratamento desses débitos. Além disso, esta pergunta ramifica o questionário para dois fluxos de respostas, caso o participante informe que gerencia os débitos técnicos, ele irá para a seção acerca do entendimento de gerenciamento, e caso contrário, ele irá para seção acerca da ausência de gerenciamento.

A seção "Entendimento Acerca do Gerenciamento de Débitos Técnicos" visa explorar as práticas e processos adotados para o gerenciamento de débitos técnicos em projetos de *software*. Cada pergunta nesta seção contribui para uma compreensão mais profunda das abordagens de gerenciamento.

Esta seção é projetada para sondar as estratégias e métodos que as equipes de *software* empregam para lidar com o débito técnico. A inclusão de perguntas específicas nesta seção é essencial para desvendar as práticas de gerenciamento de débito técnico e como elas são percebidas e implementadas no contexto dos projetos de *software*, além de ser uma seção baseada em alguns estudos realizados na área de pesquisa. Em Guo *et al.* (2016a) é investigado

os custos e desafios enfrentados pelas equipes ao gerenciar débitos técnicos, além de destacar a importância de integrar o gerenciamento de débitos técnicos nos processos existentes para superar obstáculos e reduzir custos a longo prazo, frisando a necessidade emergente de tornar este gerenciamento explícito no contexto de desenvolvimento de *software*.

O estudo de Besker *et al.* (2020) compreende como a moral dos desenvolvedores de *software* é influenciado pelo débito técnico, acrescentando também que a ocorrência de débito técnico está associada à falta de progresso e à perda de tempo. Seguindo nessa premissa, o estudo de Besker *et al.* (2020) conclui que o gerenciamento de débitos técnicos aumenta a moral e a produtividade dos desenvolvedores pelo destaque de boas práticas na codificação. Seguindo nessa perspectiva, Guo *et al.* (2016b) afirma que existe um custo inicial significativo quando se começa a acompanhar e monitorizar o débito técnico, mas o custo da gestão contínua diminui rapidamente para níveis muito razoáveis, atrelando às práticas de gerência de projetos, gestão de pessoas, gerenciamento de processos e manutenção de *software*.

Perguntas:

"Com base nos artefatos mencionados anteriormente, informe qual(is) é(são) utilizado(s) na sua equipe para organizar e rastrear débitos técnicos."

Esta pergunta busca identificar quais artefatos específicos são utilizados pela equipe para o gerenciamento de débitos técnicos. Isso é fundamental para compreender a estrutura e a abordagem da equipe na gestão desses débitos.

"Qual é o nível de envolvimento dos clientes / usuários quando são abordados tópicos relacionados a débitos técnicos no ciclo de vida do projeto?"

Esta pergunta pretende avaliar o envolvimento dos clientes e usuários é importante para entender como suas necessidades e perspectivas são consideradas no contexto de débitos técnicos, uma vez que esses problemas podem afetar a qualidade e a entrega de valor ao cliente.

"Com que frequência as atividades de gerenciamento de débitos técnicos são realizadas no seu projeto de *software*?"

Esta pergunta visa determinar a periodicidade das atividades relacionadas ao gerenciamento de débitos técnicos. A frequência pode revelar a importância dada a esse aspecto em comparação com outras tarefas no ciclo de vida do projeto.

"O que você considera como boas práticas para o gerenciamento de débitos técnicos?"

Permitir respostas abertas a essa pergunta é valioso para coletar visões sobre as percepções dos participantes em relação a boas práticas para o gerenciamento de débitos técnicos. Isso pode

incluir ideias inovadoras e práticas específicas adotadas pela equipe.

"Que melhorias ou sugestões você proporia para aprimorar o gerenciamento de débitos técnicos no seu projeto?"

Da mesma forma, a pergunta aberta permite que os participantes compartilhem suas sugestões para aprimorar o gerenciamento de débitos técnicos. Isso pode resultar em recomendações específicas e direcionadas para a melhoria das práticas da equipe.

A seção "Entendimento Acerca da Ausência de Gerenciamento de Débitos Técnicos" busca obter informações detalhadas sobre as razões pelas quais o gerenciamento de débitos técnicos não é realizado em projetos de *software*. Cada pergunta nesta seção contribui para entender as justificativas, os processos, as consequências e o potencial para a implementação do gerenciamento de débitos técnicos.

Esta seção é essencial para desvendar as complexidades e desafios que impedem a implementação efetiva de práticas de gerenciamento de débitos técnicos. Os estudos relacionados oferecem uma visão aprofundada sobre as causas comuns, as percepções dos desenvolvedores e os fatores que levam as equipes a acumular débitos técnicos, todos os quais são cruciais para entender a ausência de gerenciamento. Em Rios *et al.* (2018a) é fornecido um panorama das causas e efeitos mais comuns de débitos técnicos identificados por profissionais da indústria de *software*, destacando os prazos apertados, planejamento inadequado, falta de conhecimento e ausência de processos bem definidos como as principais causas de débitos técnicos. Os autores incluem como consequências: baixa qualidade, atrasos na entrega, manutenibilidade reduzida, retrabalho e perdas financeiras. Essas descobertas podem ser diretamente relacionadas às razões pelas quais o gerenciamento de débitos técnicos pode não estar presente em uma organização, fornecendo um contexto para as respostas dos participantes sobre as justificativas para a ausência de gerenciamento.

Por sua vez, Rocha *et al.* (2017) mergulha na percepção dos desenvolvedores sobre débitos técnicos no nível do código, destacando como a compreensão, ou a falta dela, pode influenciar na presença ou ausência de práticas de gerenciamento por parte dos desenvolvedores sobre débitos técnicos. Os autores afirmam que a perspectiva dos desenvolvedores é fundamental, pois são eles que frequentemente tomam decisões que podem resultar em débitos técnicos, seja por pressões de tempo, falta de recursos ou outras limitações.

Da mesma maneira, em Rios *et al.* (2018b) explora os fatores que levam as equipes

de desenvolvimento a acumular débitos técnicos, identificam 57 causas que levam as equipes a contrair débitos, muitas das quais ocorrem em combinação, de modo que, a compreensão desses fatores é crucial para as organizações reconhecerem e abordarem as razões subjacentes à ausência de gerenciamento de débitos técnicos. Os autores indicam que a dívida pode ser evitada e que seria melhor trabalhar em atividades de prevenção do que pagar a dívida mais tarde. Outrossim, apesar da diversidade de estratégias para identificação e gerenciamento do débito técnico na literatura técnica, há escassa evidência sobre como prevenir o surgimento desse débito e como fatores internos e externos podem criar um cenário favorável para a prevenção.

Perguntas:

"Existem processos eficazes para evitar a acumulação de débitos técnicos no projeto que você atua?"

Esta pergunta visa determinar se a equipe adota processos para evitar a acumulação de débitos técnicos. As respostas podem revelar se a ausência de gerenciamento de débitos técnicos é devido a medidas preventivas já estabelecidas.

"Quais são as práticas adotadas pelo seu projeto em relação ao monitoramento e rastreabilidade dos débitos técnicos?"

Esta pergunta permite que os participantes descrevam as práticas de monitoramento e rastreabilidade em relação aos débitos técnicos, oferecendo percepções sobre as ações atualmente em vigor.

"Na sua opinião, quais seriam os benefícios ou vantagens de implementar boas práticas de gerenciamento de débitos técnicos?"

Esta pergunta busca identificar os benefícios ao implementar boas práticas de gerenciamento de débitos técnicos, o que pode ajudar a conscientizar os participantes sobre os impactos positivos.

"Seu projeto já enfrentou consequências negativas decorrentes da ausência do gerenciamento de débitos técnicos? Se sim, por favor, descreva."

Ao coletar exemplos de consequências negativas, esta pergunta visa coletar os impactos reais da falta de gerenciamento de débitos técnicos e destaca a importância de abordar esse problema.

"Qual a justificativa ou razão específica para a ausência do gerenciamento de débitos técnicos no time?"

Esta pergunta permite que os participantes expliquem as razões específicas pelas quais o gerenciamento de débitos técnicos não é realizado em seus projetos, fornecendo informações valiosas sobre os desafios enfrentados.

"Você acredita que a existência de um artefato capaz de auxiliar no processo de gerenciamento de débitos técnicos seria relevante para utilizar no seu projeto?"

Essa pergunta avalia a percepção dos participantes sobre a utilidade de artefatos específicos para o gerenciamento de débitos técnicos, o que pode apontar para oportunidades de melhoria.

## APÊNDICE E – VERSÃO RESUMIDA DO GUIA

Neste apêndice, é apresentado a versão resumida e ilustrada do guia que será postada no LinkedIn, a fim de expandir o trabalho em cenários diversos e perspectivas abrangentes.



Fonte: Elaborado pelo autor.

## O QUE É DÉBITO TÉCNICO?

Débitos técnicos ou dívidas técnicas são decisões tomadas no desenvolvimento de software que priorizam soluções de **curto prazo** em detrimento de práticas adequadas de desenvolvimento.

Isso resulta em **problemas** técnicos, **instabilidade** e **risco** aumentado no projeto.



Fonte: Elaborado pelo autor.

## EXEMPLOS DE DÉBITOS TÉCNICOS

- Código-fonte **mal estruturado** e desorganizado;
- Documentação **precária** do código;
- Testes **inadequados**;
- Uso excessivo de soluções "**gambiarra**s" para resolver problemas.



Fonte: Elaborado pelo autor.

## GERENCIAR DÉBITOS TÉCNICOS

O gerenciamento de débitos técnicos é vital para manter a **saúde do projeto**.

Através da **identificação, monitoramento, priorização, resolução e prevenção** eficiente dos débitos técnicos, a equipe pode evitar consequências negativas.



Fonte: Elaborado pelo autor.

## IDENTIFICAR DÉBITOS TÉCNICOS

Podem se esconder profundamente no código e nos processos do software. É importante que a equipe esteja atenta a sinais de alerta:

- Práticas de codificação **questionáveis**;
- A percepção de que um **atalho** está sendo tomado para cumprir prazos;
- Observação de **fragilidade** de código e propenso a falhas.



Fonte: Elaborado pelo autor.

## IDENTIFICAR DÉBITOS TÉCNICOS

Crie um ambiente onde os desenvolvedores se sintam à vontade para relatar **preocupações** e **compartilhar** suas percepções sobre possíveis débitos técnicos.

A identificação de débitos técnicos não deve ser responsabilidade exclusiva dos desenvolvedores. **Envolva toda a equipe** no processo.



Fonte: Elaborado pelo autor.

## MONITORAR DÉBITOS TÉCNICOS

O monitoramento contínuo de débitos técnicos é essencial para **evitar** que se acumulem e se transformem em problemas que afetam a **qualidade** e a **eficiência** do software. Rastreie com:

- Documentação de código;
- Registro de erros e correções;
- Revisões de código;
- Métricas de qualidade;
- Avaliação de desempenho.



Fonte: Elaborado pelo autor.

## PRIORIZAR DÉBITOS TÉCNICOS

A priorização deve ser baseada em critérios sólidos e alinhados com os **objetivos** do projeto e as **necessidades** da organização. Alguns critérios comuns:

- Impacto no cliente;
- Impacto na qualidade;
- Risco;
- Custo de resolução;
- Valor de negócio.



Fonte: Elaborado pelo autor.

## RESOLVER DÉBITOS TÉCNICOS

Lidar com débitos técnicos durante o desenvolvimento pode ser desafiador, mas é uma prática **crucial** e **estratégica**:

- Crie um plano de ação;
- Analise o impacto;
- Aloque tempo para refatoração;
- Evite soluções de curto prazo;
- Implemente testes;
- Colabore com a equipe.



Fonte: Elaborado pelo autor.

## PREVENIR DÉBITOS TÉCNICOS

A prevenção busca evitar a acumulação de dívidas técnicas desde as fases iniciais do desenvolvimento, promovendo decisões e práticas que **minimizam** a ocorrência de problemas futuros.

Reside na garantia da **qualidade**, **sustentabilidade** e **eficiência** do software ao longo do tempo.



Fonte: Elaborado pelo autor.

## PREVENIR DÉBITOS TÉCNICOS

Boas práticas:

- Definição de padrões de codificação;
- Documentação adequada;
- Revisão de código contínua;
- Mentoria e treinamento;
- Iteração contínua;
- Desenvolvimento orientado a testes;
- Revisão de requisitos;
- Gerenciamento de projetos adequado.



Fonte: Elaborado pelo autor.

## CONCLUSÃO

Ignorar débitos técnicos pode levar a consequências negativas, como **atrasos nas entregas, problemas em produção e custos mais elevados**.

Portanto, adotar uma abordagem proativa para gerenciar débitos técnicos é essencial para o **sucesso** de qualquer projeto de software.

Esta publicação é uma versão reduzida do artefato gerado pelo autor no Trabalho de Conclusão de Curso.



Fonte: Elaborado pelo autor.