

# Use of an Approach for Preventing Test Debt in a Brazilian Fintech: An Experience Report

Levi Almeida da Silva  
levi.silva@aluno.uece.br  
Universidade Estadual do Ceará  
Fortaleza, Ceará, Brasil

Ismayle de Sousa Santos  
ismayle.santos@uece.br  
Universidade Estadual do Ceará  
Fortaleza, Ceará, Brasil

Rossana M. C. Andrade  
rossana@ufc.br  
Universidade Federal do Ceará  
Fortaleza, Ceará, Brasil

## RESUMO

The demand for software innovation has challenged organizations to meet deadlines and quality standards. Technical Debt (TD) refers to aspects of software development that are deferred to achieve immediate benefits but result in negative consequences for the product's long-term quality. In the software development cycle, the Testing phase also accumulates Technical Debt, known as Test Debt, compromising this stage. Aiming to support the management of such debts, this paper aims to present an experience report on the application of strategies to prevent Test Debt in a Brazilian fintech. This work describes how the approach was implemented, the results obtained, and the lessons learned from the experience.

## CCS CONCEPTS

• Software and its engineering → Software development methods.

## KEYWORDS

Test Debt. Software Testing. Technical Debt Prevention

### ACM Reference Format:

Levi Almeida da Silva, Ismayle de Sousa Santos, and Rossana M. C. Andrade. 2024. Use of an Approach for Preventing Test Debt in a Brazilian Fintech: An Experience Report. In *XXIII Brazilian Symposium on Software Quality (SBQS 2024)*, November 5–8, 2024, Salvador, Brazil. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3701625.3701639>

## 1 INTRODUÇÃO

A demanda por inovações em software tem gerado desafios para as organizações de desenvolvimento de software, especialmente em relação ao cumprimento de prazos e padrões de qualidade. Dessa forma, a necessidade de entregar novos produtos ou funcionalidades pode exigir ajustes devido à falta de tempo, ou outros fatores. Esses ajustes podem resultar em “Dívidas Técnicas” (DT).

Conforme descrito por Tom *et al.* [13], a Dívida Técnica é uma metáfora que representa as consequências de um desenvolvimento de software deficiente, podendo se manifestar em várias etapas do processo de desenvolvimento.

Dentro desse contexto, Primão *et al.* [8] destacam que o processo de teste é uma etapa intrínseca e relevante no ciclo de desenvolvimento de software, desempenhando um papel crucial na promoção da maturidade do sistema. Segundo Roger e Bruce [11], em contextos de empreendimentos de software, a escassez de recursos temporais ou materiais leva as instituições a comprometer os procedimentos de garantia de qualidade. Esse tipo de comportamento também resulta na “Dívida Técnica”, que no campo de Teste de software é chamado de “Dívida de Teste”. Essa forma de Dívida Técnica ocorre devido a decisões inadequadas nas atividades de teste, como falta de testes adequados e estimativas incorretas, como destaca por Samarthya *et al.* [12].

Diante desse cenário, objetiva-se a gestão da ocorrência de Dívida Técnica (DT) mediante a implementação de práticas de gerenciamento. De acordo com Li *et al.* [5], as práticas de gerenciamento podem abordar oito fundamentos relacionados às Dívidas Técnicas, são elas: (i) Prevenção; (ii) Identificação; (iii) Medição; (iv) Priorização; (v) Monitoramento; (vi) Reembolso/Pagamento; (vii) Documentação e (viii) Comunicação.

Conscientes dessas bases, Pérez *et al.* [7] sustentam que técnicas preventivas desempenham um papel crucial na mitigação das Dívidas Técnicas, o que é corroborado por Rios *et al.* [10] ao observar que a prevenção de Dívidas pode ser mais eficaz e econômica para a equipe de desenvolvimento do que a posterior liquidação.

Conforme Aragão *et al.* [3], a maioria dos estudos se concentra apenas em algumas causas específicas de Dívida de Teste (por exemplo, falta de testes automatizados). Além disso, os autores afirmam que não foram identificadas ferramentas ou metodologias que abordassem de maneira abrangente a prevenção da DT para esse tipo de dívida.

Com base nisso, o trabalho de Silva [4] propõe um catálogo de estratégias de prevenção para Dívidas de Teste, visando proporcionar uma abordagem para profissionais envolvidos no desenvolvimento de software.

Assim, no contexto de uma fintech brasileira, foi aplicado a proposta de Silva [4] com o intuito de avaliar se sua abordagem se torna útil e se as estratégias propostas previnem as dívidas de testes reportadas pelos profissionais de teste de software da empresa.

Este artigo apresenta um relato de experiência sobre o uso da abordagem mencionada, destacando os resultados obtidos e as lições aprendidas no processo de prevenção de Dívidas de Teste. Os resultados obtidos não apenas podem apoiar empresas desenvolvedoras de software na adoção de boas práticas de gestão de requisitos, mas também incentivar pesquisadores da área de qualidade de software a realizar novas pesquisas sobre a prevenção de Dívidas de Teste.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

SBQS 2024, November 5–8, 2024, Salvador, Brazil

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-1777-2/24/11

<https://doi.org/10.1145/3701625.3701639>

## 2 TRABALHOS RELACIONADOS

O estudo de Ampatzoglou *et al.* [2] apresenta o desenvolvimento da ferramenta SDK4ED, voltada para o gerenciamento de Dívida Técnica, e descreve a experiência de sua aplicação em quatro empresas. A coleta de dados ocorreu durante um período de teste de 30 dias, precedido por um workshop introdutório de um dia. Foram utilizados dois questionários e um grupo focal para a obtenção dos dados. A validação da plataforma combinou análises quantitativas e qualitativas: a análise quantitativa avaliou a usabilidade da ferramenta, com uma média de 76,8 de 100, e a análise qualitativa utilizou a técnica de Análise de Conteúdo Qualitativa (QCA) para interpretar as informações obtidas no grupo focal. Por mais que o trabalho aborde a prevenção como consequência do gerenciamento de DT pela plataforma, o trabalho não foca em DTs no contexto de Teste de Software.

No estudo de Oliveira [6], que replica a pesquisa de Ribeiro [9], participantes atuaram como engenheiros de software em uma reunião para monitoramento da Dívida Técnica (DT). A reunião envolveu a revisão de itens de DT para decidir quais deveriam ser eliminados antes do próximo lançamento do sistema. Os participantes avaliaram listas de itens com e sem a utilização da estratégia suportada pela ferramenta TD Manager, preenchendo formulários sobre a utilidade, facilidade de uso e possíveis melhorias da estratégia. A replicação mostrou resultados positivos, com 86% dos participantes indicando que utilizariam a estratégia para gerenciar a DT. Enquanto o estudo anterior visa a gestão de itens já existentes de DT, este relato abordou iniciativas práticas para prevenir o acúmulo de dívidas no contexto de testes, antes que se tornem um problema que precise ser gerenciado.

No trabalho de Alfayez, [1], os autores examinaram a percepção e a priorização de itens de DT identificados pelo SonarQube para fins de pagamento. Realizado em quatro empresas e em um curso de mestrado, com a participação de 89 indivíduos, o estudo revelou que a maioria dos participantes priorizou os itens com base em uma combinação de valor e custo. Enquanto poucos participantes deram prioridade a itens de maior valor, apenas um focou em itens de menor custo. Os resultados indicam que o valor atribuído a um item de DT é subjetivo e contextual, embora a maioria dos participantes considere as estimativas de custo fornecidas pelo SonarQube como confiáveis para a priorização de DT. Dessa forma, enquanto o estudo em questão se concentra na tomada de decisões sobre dívidas técnicas já identificadas e na priorização de seu pagamento com base em valor e custo, porém este trabalho foca na prevenção de DT no contexto de testes, buscando minimizar a necessidade de priorização e posterior pagamento dessas dívidas.

Ademais, o trabalho de Silva [4], constrói um catálogo que relaciona Dívidas de Teste com potenciais estratégias de prevenção para tais dívidas. A intenção dessa abordagem é que esta seja usada por profissionais do desenvolvimento de software a fim de dar suporte à gestão de Dívidas de Teste. No entanto, a solução proposta não foi implementada diretamente no ambiente da indústria, de modo a validar sua eficácia ou utilidade prática no gerenciamento de Dívidas de Teste.

Além disso, embora existam trabalhos que abordam a gestão de dívida técnica de maneira geral, muitos não enfatizam a prevenção,

e a maioria desses estudos está voltada para aspectos de código em vez de testes.

## 3 CONTEXTUALIZAÇÃO

As informações apresentadas nas subseções seguintes foram as autorizadas pela empresa para divulgação. Alguns detalhes foram suprimidos em razão da confidencialidade das atividades da empresa.

### 3.1 Fintech

A fintech em questão atua em uma plataforma tecnológica que fornece uma infraestrutura e ferramentas para entidades interessadas em oferecer crédito. Esta plataforma é classificada como um modelo de “Credit as a Service One-Stop-Shop”. De acordo com a Resolução BCB nº 465666/18, esta modalidade pode ser considerada uma Sociedade de Empréstimo entre Pessoas (SEP), caracterizada como uma instituição financeira que realiza operações de empréstimo e financiamento exclusivamente por meio de uma plataforma eletrônica. No período da pesquisa, a empresa era composta por 8 equipes e contava com um total de 150 a 200 funcionários. O setor de tecnologia, em particular, incluía entre 20 e 80 profissionais, dos quais aproximadamente 12 atuavam como QAs.

### 3.2 Organização das Equipes

Para a evolução e manutenção da plataforma em questão, a equipe de desenvolvimento está organizada em diversas equipes, cada uma com responsabilidades específicas. Cada squad inclui pelo menos um profissional de Quality Assurance (QA), encarregado principalmente da execução dos testes. As squads trabalham com microsserviços e incluem profissionais especializados em back-end, front-end e gestão de produto, geralmente representados por um Product Owner (PO). A maioria das equipes adota metodologias ágeis, principalmente Scrum. Predominantemente, as equipes operam em modalidade home office, embora alguns membros prefiram trabalhar presencialmente.

### 3.3 Processo de Teste

O processo de testes adotado seguia as fases de Planejamento, Especificação, Execução e Encerramento. Na fase de Planejamento, eram definidas as estratégias de testes, incluindo objetivos, recursos necessários, cronograma e critérios de aceitação. Em seguida, na fase de Especificação, eram elaborados os casos de teste com base nos requisitos do sistema, documentando cenários, dados de entrada e resultados esperados.

Durante a Execução, os testes eram realizados, comparando os resultados obtidos com os esperados, registrando defeitos e comunicando-se com a equipe de desenvolvimento para resolver problemas. Já na fase de Encerramento, analisavam-se os resultados dos testes, elaborava-se um relatório com os resultados obtidos e dificuldades encontradas e discutiam-se lições aprendidas, arquivando os artefatos para futuras referências.

É importante destacar que o Jira <sup>1</sup> foi utilizado para a gestão das histórias de usuário, permitindo que os QAs acompanhassem o status de cada funcionalidade. Com base nesses status, o Confluence <sup>2</sup>

<sup>1</sup><https://www.atlassian.com/br/software/jira>

<sup>2</sup><https://www.atlassian.com/br/software/confluence>

era empregado para a documentação dos artefatos correspondentes às etapas de Planejamento, Especificação, Execução e Encerramento, consolidando todas essas informações em um único documento.

4 ABORDAGEM DE PREVENÇÃO APLICADA

4.1 Trabalho de Silva (2023)

O trabalho de Silva [4] relaciona 47 categorias de estratégias de prevenção de Dívidas Técnicas com 17 categorias de Dívidas de Teste. As categorias de Dívidas de Teste podem ser encontradas na Tabela 1 e as categorias de estratégias de prevenção de Dívidas Técnicas podem ser encontradas na Tabela 2. Além disso, este trabalho faz a relação entre esses elementos, assim como mostra a Tabela 3. Vale ressaltar que ausência de conteúdo na célula, implica que aquela estratégia de prevenção não evita tal Dívida de Teste. Por exemplo, tendo conhecimento dos IDs da Tabela 1 e dos IDs da Tabela 2, pode-se observar que na Tabela 3, a estratégia de prevenção 1 (Adotar boas práticas de testes) pode prevenir a Dívida de Teste 3 (Atraso na realização de testes).

Tabela 1: Categorias de Dívidas de Teste (Silva 2023)

ID	Categoria de Dívida de Testes
1	Adiamento de testes
2	Alocação inadequada da equipe
3	Atraso na realização de testes
4	Baixa cobertura de código
5	Baixa cobertura de testes
6	Defeitos não encontrados em testes
7	Equipamento inadequado para realização de testes
8	Erro de estimativa de teste
9	Erros de execução de teste
10	Falta de automação de testes
11	Falta de interpretação de teste
12	Falta de testes
13	Falta de testes unitários
14	Testes incompletos
15	Testes caros
16	Teste inadequado
17	Testes interrompidos

4.2 Coleta de Dados

O processo de coleta de informações durante a aplicação da abordagem de prevenção de dívidas de testes seguiu as etapas delineadas na Figura 1. Inicialmente, buscamos traçar o perfil dos QAs que participaram da experiência identificando o tempo de atuação na função, o nível de escolaridade, tempo de atuação na equipe atual e a familiaridade com os conceitos de DT e Dívida de Teste. Para isso, foi elaborado um formulário que apresentava, em sua descrição, os conceitos de DT e Dívida de Teste e outras perguntas para identificação do perfil. O perfil dos respondentes pode ser encontrado na Tabela 4.

Para identificar as principais Dívidas de Teste ocorridas nos últimos dois meses, os QAs avaliaram a frequência dessas dívidas em suas equipes durante o período mencionado, classificando-as

Tabela 2: Categorias de Estratégias de Prevenção Dívidas Técnicas (Silva 2023)

ID	Categorias de Estratégias de Prevenção de DT
1	Adotar boas práticas de testes
2	Alocar adequadamente as tarefas entre a equipe
3	Alocar Profissionais Qualificados na equipe
4	Alocar recursos na equipe com um todo
5	Analisar o custo-benefício do teste
6	Analisar profundamente as funcionalidades envolvidas
7	Analisar risco e impactos dos testes
8	Automatizar tarefas (não necessariamente testes)
9	Considerar restrições técnicas
10	Criar testes automatizados
11	Criar testes para as novas funcionalidades
12	Entender o processo de desenvolvimento da empresa
13	Escrever bons casos de testes de integração
14	Escrever bons casos de testes de sistema
15	Escrever bons casos de testes unitários
16	Fornecer os requisitos mais cedo
17	Implementar ações preventivas de dívidas técnicas
18	Localizar e corrigir bugs com rapidez e precisão
19	Manter as documentações pertinentes atualizadas e bem definidas
20	Melhorar estimativa de esforço de teste
21	Melhorar planejamento de projeto
22	Monitorar e controlar atividades do projeto
23	Organizar repositório do projeto
24	Pagar dívidas técnicas
25	Remover tarefas de baixa prioridade
26	Ter ambientes para testes
27	Ter atividades padronizadas
28	Ter claro a Definição de pronto
29	Ter comprometimento com sua responsabilidade
30	Ter conhecimento técnico
31	Ter conscientização e gerenciamento de Dívidas Técnicas
32	Ter integração continua (CI)
33	Ter métricas de testes bem definidas
34	Ter padrões de código
35	Ter prazos bem definidos
36	Ter reuniões de retrospectiva ou para analisar causa e consequência de DT
37	Ter um código bem escrito
38	Ter uma arquitetura do sistema bem definida
39	Ter uma boa comunicação com a equipe
40	Ter uma cobertura de testes adequada
41	Ter uma definição clara do escopo de trabalho
42	Ter uma equipe organizada
43	Ter uma governança em TI
44	Treinar adequadamente os membros da equipe
45	Usar diversas estratégias de testes
46	Usar interações curtas de feedback
47	Utilizar as lições aprendidas para uma próxima etapa

**Tabela 3: Catálogo de estratégias de Prevenção de Dívidas de Teste (Silva 2023)**

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	x		x	x		x			x		x	x	x	x	x	x	x
2	x	x	x					x						x			
3	x	x	x	x	x	x		x	x	x	x	x	x	x		x	x
4	x	x	x				x							x			
5					x		x							x	x		
6	x		x		x	x		x	x		x	x				x	
7	x		x		x	x						x			x	x	
8	x		x		x			x		x							
9								x	x		x					x	
10	x		x		x	x		x	x	x	x	x				x	
11				x	x	x						x	x				
12	x		x		x	x			x			x					x
13	x		x		x	x		x	x		x	x				x	
14	x		x		x	x		x	x		x	x				x	
15				x									x				x
16	x		x		x			x	x		x	x				x	
17	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
18									x			x					
19	x		x		x	x		x	x		x	x		x			
20	x		x					x									
21	x		x					x			x	x	x		x		
22	x		x			x		x				x	x		x		x
23				x									x				x
24	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
25	x		x											x			
26	x		x			x						x		x	x	x	
27	x		x			x		x			x	x					
28	x		x	x	x	x		x	x		x			x		x	
29	x		x	x	x	x		x	x		x	x	x	x		x	
30	x		x	x	x	x		x	x	x	x	x	x	x	x	x	x
31	x	x	x	x	x	x	x	x	x			x	x	x	x	x	
32				x									x				x
33	x		x	x	x	x		x					x		x		
34				x						x			x				x
35	x		x									x	x	x			
36	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
37				x						x			x				x
38				x						x			x				x
39	x	x	x	x	x	x	x	x		x	x	x	x	x		x	
40	x		x		x	x				x		x	x	x	x	x	
41	x		x		x	x		x	x	x	x	x		x	x	x	
42	x	x	x		x						x	x	x				
43	x		x										x				x
44	x	x	x	x	x	x		x	x	x	x	x	x	x		x	x
45	x		x	x	x	x				x		x	x	x	x	x	
46	x		x	x	x	x		x	x	x	x	x	x			x	x
47	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

em uma escala que variava de “sem ocorrência” a “muito frequente”. Para os QAs 1, 3 e 4, as principais dívidas de teste foram aquelas classificadas como “frequência moderada” e “muito frequente”. Em contrapartida, para o QA 2, as dívidas de teste consideradas foram as categorizadas como “pouco frequente”, uma vez que não relatou

ocorrências com “frequência moderada” ou “muito frequente”. O resultado das respostas pode ser encontrado na Tabela 5.

Outra etapa na coleta de dados, após a identificação das principais dívidas de teste, envolveu a realização de entrevistas semiestruturadas com os respondentes. A condução dessas entrevistas incluiu uma breve explicação sobre dívidas técnicas e dívidas de teste, além

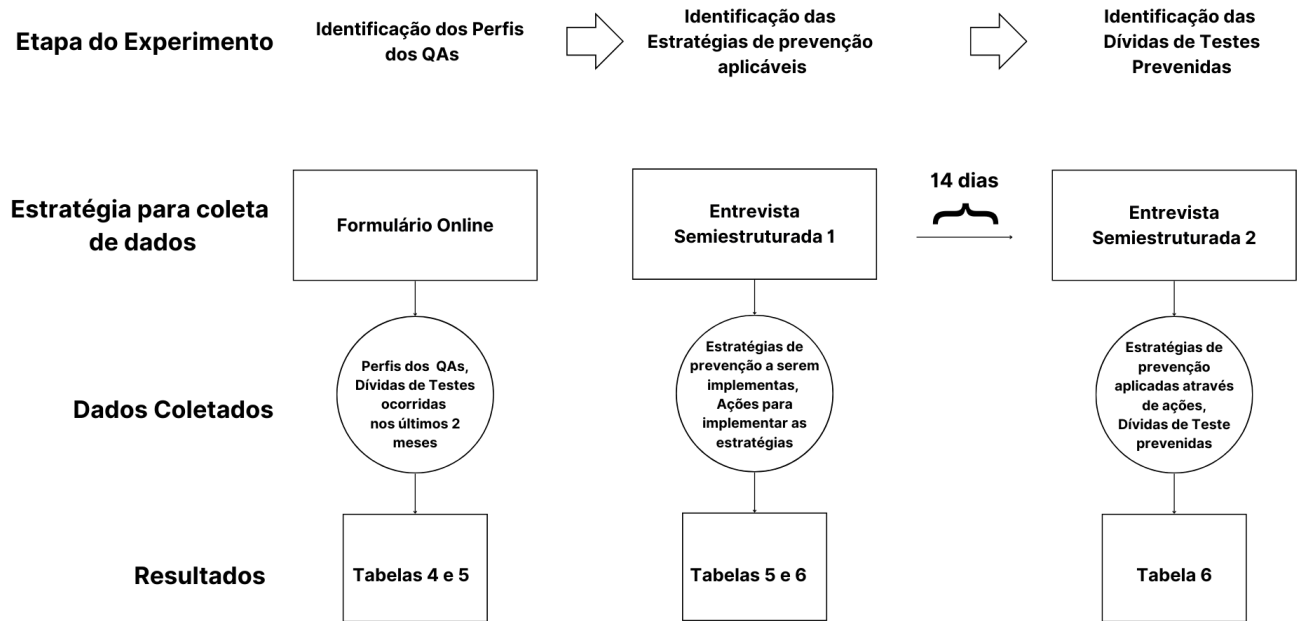


Figura 1: Etapas do processo de coleta de dados

de perguntas específicas, tais como: “Quais ações você toma para prevenir dívidas de teste?” “Quais ações de prevenção você aplica para as respectivas dívidas de teste reportadas?” “Quais dessas estratégias são factíveis de serem aplicadas em sua rotina?” “Na sua rotina, como você aplicaria essas estratégias?”. A sumarização dessas respostas podem ser encontrada na Tabela 5.

Os profissionais de QA aplicaram as estratégias consideradas viáveis durante um período de duas semanas. Após esse período, foi realizada uma nova entrevista semiestruturada com os seguintes questionamentos: “Quais das estratégias você conseguiu aplicar?”, “Quais ações você conseguiu executar para implementar essas estratégias?” “Você sentiu que as estratégias aplicadas preveniram as dívidas de teste reportadas no formulário?”, “Você continuaria usando essas estratégias de prevenção ou as considera úteis?”.

## 5 RESULTADOS E DISCUSSÕES

A seguir são exibidos os resultados com base nas coletas de dados realizadas.

### 5.1 Perfil dos Respondentes

A Tabela 4 apresenta o perfil dos quatro profissionais de QA que participaram deste estudo, destacando suas formações acadêmicas, tempo de atuação como QA, tempo na atual squad, e familiaridade com Dívida de Teste. Os dados revelam uma diversidade significativa entre os participantes, tanto em termos de formação quanto de experiência prática.

Os QAs possuem diferentes níveis de formação, variando desde graduandos até pós-graduandos, o que pode influenciar suas abordagens no tratamento de Dívida de Teste. O tempo de experiência como QA também varia de menos de um ano até três anos, sugerindo que alguns profissionais possuem maior vivência em atividades de qualidade, enquanto outros ainda estão em uma fase mais inicial de suas carreiras.

Adicionalmente, o tempo de atuação na atual squad varia entre seis meses e dois anos, o que pode impactar o grau de adaptação e entendimento dos processos da equipe. Por fim, a familiaridade com o conceito de Dívida de Teste também apresenta variações, com níveis de 1 a 5, o que indica que alguns profissionais estão mais confortáveis com o conceito, enquanto outros ainda estão em fase de aprendizado. Esses fatores são importantes para interpretar as contribuições de cada participante ao longo do estudo.

### 5.2 Entrevista Semiestruturada: Estratégias de Prevenção de DT Personalizadas

Conforme explicado na seção anterior, o objetivo da primeira entrevista semiestruturada foi identificar quais estratégias de prevenção eram aplicáveis de acordo com o contexto de cada QA. Sendo assim, os QAs identificaram quais Dívidas de Teste eram mais frequente em suas squads nos últimos 2 meses. Vale ressaltar que devido ao tempo e outros fatores, foram propostas as estratégias que tinham o potencial de prevenir todas as Dívidas de Testes reportadas pelos QAs. Os resultados estão apresentados na Tabela 5. Percebe-se que as Dívidas de teste “Erros de Estimativa de Testes” e “Falta

**Tabela 4: Perfil dos Respondentes**

QA	Grau de Formação	Tempo atuando como QA	Tempo atuando na atual Squad	Familiaridade com Dívida de Teste
QA 1	Graduando	Entre 1 e 2 anos	6 meses	2
QA 2	Graduado	Entre 2 e 3 anos	6 meses	4
QA 3	Graduado	Menos que 1 ano	10 meses	1
QA 4	Pós - Graduando	Entre 2 e 3 anos	2 anos	5

de interpretação de Teste” foram reportados por 3 dos 4 QAs que participaram da pesquisa.

Durante as entrevistas, também foram discutidas as ações a serem implementadas para aplicar as estratégias sugeridas, dado o caráter abrangente de cada uma. Para a estratégia de realizar reuniões sobre as causas da Dívida Técnica (DT), todos os QAs que identificaram essa necessidade, indicaram que planejavam integrar esses momentos nas suas dailys.

Quanto à estratégia de manter uma boa comunicação com a equipe e realizar interações curtas de feedback, os QAs que lidavam com esses tipos de Dívidas de Teste iriam recorrer a chamadas e chats com os membros de suas equipes para otimizar a colaboração. As demais ações são detalhadas abaixo:

#### **Estratégias e Ações para o QA 1**

- Ter reuniões sobre causas de Dívidas Técnicas
  - A1: Comentar sobre as DTs nas dailys
- Ter uma boa comunicação com a equipe
  - A2: Ser participativo nas reuniões com o time
- Utilizar lições aprendidas na próxima etapa
  - A3: Pedir ajuda de outro QA para entregar os testes no prazo

#### **Estratégias e Ações para o QA 2**

- Manter documentações atualizadas
  - A4: Ficar em contato com o time de Produtos para revisar regras de negócio
- Definição clara do escopo de trabalho
  - A5: Ler a documentação e tirar dúvidas
- Interações curtas de feedback
  - A6: Utilizar chamadas ou chat com a equipe
- Utilizar lições aprendidas na próxima etapa
  - A7: Colaborar com documentações para estar a par da funcionalidade a ser testada

#### **Estratégias e Ações para o QA 3**

- Utilizar lições aprendidas na próxima etapa
  - A8: Pedir acesso a outros membros para revisar ambientes e funcionalidades antes dos testes

#### **Estratégias e Ações para o QA 4**

- Ter reuniões sobre causas de Dívida Técnica
  - A1: Comentar sobre as DTs nas dailys
- Ter boa comunicação com a equipe
  - A2: Ser participativo nas reuniões com o time
- Utilizar lições aprendidas na próxima etapa
  - A9: Finalizar a escrita dos casos de teste antes da funcionalidade estar disponível

Além disso, no tocante às estratégias gerais para a prevenção de Dívidas de Teste, antes de ser apresentada as estratégias do catálogo,

os QAs destacaram ações como a gestão eficiente do tempo, o entendimento detalhado das demandas e o acesso às funcionalidades da plataforma. Essas duas últimas já estavam presente no catálogo.

### **5.3 Entrevista Semiestruturada: Aplicação das Estratégias**

Conforme detalhado anteriormente, o objetivo da segunda entrevista semiestruturada foi avaliar a efetividade das estratégias de prevenção aplicadas e verificar se contribuíram para a mitigação das Dívidas de Teste identificadas. Com base nas respostas dos QAs, foi estabelecida a relação entre as estratégias aplicadas, as ações executadas e as Dívidas de Teste prevenidas.

É importante destacar que as estratégias foram aplicadas antes da execução dos testes nas novas funcionalidades, permitindo que os próprios QAs comparassem se os testes subsequentes foram realizados dentro do prazo ou com maior fluidez em função das estratégias adotadas.

Além disso, todos os QAs entrevistados consideraram as estratégias úteis em suas dinâmicas de trabalho e afirmaram que pretendem continuar utilizando essas práticas. Eles também observaram que, mesmo as estratégias que não foram implementadas durante o período do estudo, apresentam potencial para prevenir Dívidas de Teste associadas. A relação entre o conjunto de ações aplicadas e as Dívidas de Teste prevenidas se encontram na Tabela 6.

Um exemplo dessa relação é sobre a aplicação da estratégia “Utilizar Lições Aprendidas na Próxima Etapa” que variou entre os profissionais. O QA 1, por exemplo, implementou essa abordagem buscando apoio de outros QAs para os testes futuros, enquanto o QA 3 adotou a medida de solicitar acesso antecipado a ambientes e funcionalidades, garantindo maior preparação antes da execução dos testes.

## **6 LIÇÕES APRENDIDAS**

Nesta seção, são apresentadas as lições aprendidas durante o período de aplicação da abordagem.

**Estratégias Personalizadas:** Observou-se que as estratégias de prevenção devem ser adaptadas ao contexto específico de cada equipe de Desenvolvimento de software. Para isso, é fundamental realizar uma análise aprofundada das condições e desafios enfrentados por cada profissional.

**Ações executadas:** As estratégias de prevenção do catálogo utilizado foram descritas de forma superficial, o que inicialmente dificultou a compreensão dos QAs sobre como aplicá-las efetivamente. Um exemplo é a estratégia de “utilizar lições aprendidas para a próxima etapa”, que apresentou variedade de interpretação em sua implementação prática.

Tabela 5: Dívidas de Testes Reportadas e Estratégias de Prevenção Aplicadas

QA	Dívidas de Teste (últimos 2 meses)	Frequência	Estratégias Personalizadas	Estratégias Aplicadas
QA 1	Erros de Estimativa de Testes	Muito Frequente	Ter reuniões sobre causas de Dívidas Técnicas, Ter uma Boa comunicação com a Equipe, Treinar Membros da equipe, Utilizar lições aprendidas na próxima etapa	Ter reuniões sobre causas de Dívidas Técnicas, Ter uma Boa comunicação com a Equipe, Utilizar Lições aprendidas na próxima etapa
	Falta de automação de Testes	Frequência Moderada		
	Falta de interpretação de Testes	Frequência Moderada		
QA 2	Atraso na realização de Testes	Pouco Frequente	Manter Documentações Atualizadas, Ter reuniões sobre causas de Dívidas Técnicas, Definição Clara do Escopo de Trabalho, Treinar Membros da Equipe, Interações Curtas de Feedback, Utilizar lições aprendidas na próxima Etapa, Criar testes automatizados, Escrever Bons casos de testes, Analisar profundamente as funcionalidades envolvidas	Manter Documentações atualizadas, Definição Clara do Escopo de Trabalho, Interações Curtas de Feedback, Utilizar Lições Aprendidas na Próxima Etapa
	Defeitos não encontrado em Testes	Pouco frequente		
	Erro de Estimativa de Teste	Pouco Frequente		
	Erro de execução de Teste	Pouco Frequente		
	Falta de Interpretação de Teste	Pouco Frequente		
	Falta de Teste	Pouco Frequente		
QA 3	Testes interrompidos	Muito Frequente	Ter reuniões sobre causas de Dívidas Técnicas, Utilizar Lições aprendidas na Próxima Etapa	Utilizar lições aprendidas na próxima etapa
	Equipamento Inadequado para Teste	Frequência Moderada		
	Erro de execução de Teste	Frequência Moderada		
	Testes Incompletos	Frequência Moderada		
QA 4	Falta de Automação de Testes	Muito Frequente	Ter reuniões sobre causa de Dívida Técnica, Ter Boa comunicação com a equipe, Treinar Membros da Equipe, Utilizar Lições aprendidas na Próxima Etapa	Ter reuniões sobre causa de Dívida Técnica, Ter boa comunicação com a Equipe, Utilizar Lições aprendidas na próxima etapa
	Adiamento de Testes	Frequência Moderada		
	Alocação inadequada da equipe	Frequência Moderada		
	Atraso na realização de teste	Frequência Moderada		
	Erro de estimativa de Teste	Frequência Moderada		
	Falta de Interpretação de Teste	Frequência Moderada		
	Falta de Teste	Frequência Moderada		
	Falta de Teste Unitário	Frequência Moderada		

Tabela 6: Dívidas de Testes Prevenidas

QA	Dívidas de Teste (últimos 2 meses)	Estratégias Aplicadas	Ações Executadas	Dívidas de Teste Prevenidas
QA 1	Erros de Estimativa de Testes	Ter reuniões sobre causas de Dívidas Técnicas, Ter uma boa comunicação com a equipe, Utilizar lições aprendidas na próxima etapa	Comentar sobre as DTs nas dailys, Ser participativo nas reuniões com o time, Pedir ajuda de outro QA para entregar os testes no prazo	Erro de Estimativa de Teste, Falta de Interpretação de Teste
	Falta de automação de Testes			
	Falta de interpretação de Testes			
QA 2	Atraso na realização de Testes	Manter Documentações atualizadas, Definição clara do escopo de trabalho, Interações curtas de feedback, Utilizar lições aprendidas na próxima etapa	Ficar em contato com o time de Produtos para revisar regras de negócio, Ler a documentação e tirar dúvidas, Utilizar chamadas ou chat com a equipe Colaborar com documentações para estar a par da Funcionalidade a ser testada	Atraso na realização de Teste, Erro de Estimativa de Teste, Erro de Execução de Teste, Falta de Interpretação de Teste
	Defeitos não encontrados em Testes			
	Erro de Estimativa de Teste			
	Erro de execução de Teste			
	Falta de Interpretação de Teste			
	Falta de Teste			
QA 3	Testes interrompidos	Utilizar lições aprendidas na próxima etapa	Pedir acesso a outros membros para revisar ambientes e funcionalidades antes dos testes	Erro de Execução de Teste, Testes Incompletos
	Equipamento inadequado para Teste			
	Erro de execução de Teste			
	Testes Incompletos			
QA 4	Falta de Automação de Testes	Ter reuniões sobre causa de Dívida Técnica, Ter boa comunicação com a equipe, Utilizar lições aprendidas na próxima etapa	Comentar sobre as DTs nas dailys, Ser participativo nas reuniões com o time, Finalizar a escrita dos casos de teste antes da funcionalidade estar disponível	Adiamento de Teste, Atraso na realização de Teste, Falta de Interpretação de Teste, Falta de Teste
	Adiamento de Testes			
	Alocação inadequada da equipe			
	Atraso na realização de teste			
	Erro de estimativa de Teste			
	Falta de Interpretação de Teste			
	Falta de Teste			
	Falta de Teste Unitário			

**Outras Dívidas de Teste e Outras Estratégias de Prevenção:**  
Durante o período de estudo, foram identificadas novas dívidas de teste e estratégias de prevenção, indicando que ainda é necessária uma investigação mais detalhada sobre as práticas e realidades da indústria.

**Métricas de Prevenção:** Uma das dificuldades encontradas foi a definição de como medir efetivamente a prevenção das Dívidas de Teste. Durante o período do estudo, não foi possível acompanhar de perto o trabalho de cada QA. Por isso, a avaliação sobre a eficácia das estratégias aplicadas foi baseada na percepção dos próprios QAs,



que compararam seus testes antes e depois da implementação das ações preventivas. Logo, a definição de métricas poderia auxiliar a quantificar os ganhos com a prevenção de dívidas técnicas.

**Aplicação do Catálogo:** O catálogo de estratégias de prevenção configura-se como uma planilha extensiva, totalizando 799 células, distribuídas em 47 categorias de estratégias de prevenção e 17 categorias de Dívidas de Teste. Essa complexidade tornou desafiadora a organização e a aplicação dessas estratégias sem o auxílio de um software especializado em gestão de planilhas. A utilização de uma ferramenta específica voltada para a gestão desse tipo de catálogo poderia proporcionar uma interface mais intuitiva e facilitar o processo de implementação das estratégias preventivas

## 7 CONSIDERAÇÕES FINAIS

Neste relato de experiência, foi aplicada na prática a abordagem de prevenção de Dívida de Teste proposta por Silva et al. (2023). A abordagem foi implementada em uma fintech brasileira e utilizada por quatro profissionais de Quality Assurance (QA) da mesma. Os resultados indicaram que a abordagem é de fato útil de acordo com o contexto, mas ainda requer algumas melhorias, como definições mais claras, estratégias mais concretas e uma maior intuição do catálogo.

Como trabalho futuro, pretende-se expandir a aplicação dessa abordagem para outros contextos, validar a implementação de estratégias adicionais e propor uma abordagem aprimorada com base na avaliação obtida a partir da aplicação prática. Além disso, pretende-se definir métricas para mensurar os ganhos com as prevenções realizadas.

## 8 AGRADECIMENTOS

Gostaríamos de agradecer ao CNPQ a bolsa de mestrado concedida ao Levi Almeida da Silva e à bolsa de Produtividade DT à Rossana M.C. Andrade.

## REFERÊNCIAS

- [1] Reem Alfayez, Robert Winn, Wesam Alwehaibi, Elaine Venson, and Barry Boehm. 2023. How SonarQube-identified technical debt is prioritized: An exploratory case study. *Information and Software Technology* 156 (2023), 107147.
- [2] Apostolos Ampatzoglou, Alexander Chatzigeorgiou, Elvira Maria Arvanitou, and Stamatia Bibi. 2022. SDK4ED: A platform for technical debt management. , 1879–1902 pages.
- [3] Bruno S Aragão, Rossana MC Andrade, Ismayle S Santos, Rute NS Castro, Valéria Lelli, and Ticianne GR Darin. 2019. Testdcat: Catalog of Test Debt subtypes and management activities. In *Testing Software and Systems: 31st IFIP WG 6.1 International Conference, ICTSS 2019, Paris, France, October 15–17, 2019, Proceedings 31*. Springer, 279–295.
- [4] Levi Almeida da Silva. 2023. Um Catálogo de estratégias de prevenção para Dívidas Técnicas de Teste. *Trabalho de Conclusão de Curso - Universidade Estadual do Ceará* (2023). <https://siduece.uece.br/siduece/pesquisarItemPublico>
- [5] Zengyang Li, Paris Avgeriou, and Peng Liang. 2015. A systematic mapping study on technical debt and its management. *Journal of Systems and Software* 101 (2015), 193–220.
- [6] Annanette Rabelo Batista de Oliveira. 2021. Estratégia baseada em critérios para apoiar a tomada de decisão sobre o pagamento de itens da dívida técnica: um estudo replicado. (2021).
- [7] Boris Pérez, Camilo Castellanos, Dario Correal, Nicolli Rios, Sávio Freire, Rodrigo Spinola, Carolyn Seaman, and Clemente Izurieta. 2021. Technical debt payment and prevention through the lenses of software architects. *Information and Software Technology* 140 (2021), 106692.
- [8] Aline Pacheco Primão, Patric da Silva Ribeiro, and Diego Luís Kreutz. 2010. Estudo de Caso: Técnicas de Teste como parte do Ciclo de Desenvolvimento de Software. *Universidade Federal do Pampa-UNIPAMPA, Alegrete, RS* (2010).
- [9] Leilane Ferreira RIBEIRO. 2016. Uma estratégia baseada em critérios para apoiar a tomada de decisão sobre o pagamento de itens de dívida técnica.
- [10] Nicolli Rios, Rodrigo Oliveira Spinola, Manoel G. de Mendonça Neto, and Carolyn Seaman. 2018. 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)*. 429–436. <https://doi.org/10.1109/SEAA.2018.00076>
- [11] S Pressman Roger and R Maxin Bruce. 2015. *Software engineering: a practitioner's approach*. McGraw-Hill Education.
- [12] Ganesh Samarthayam, Mahesh Muralidharan, and Raghu Kalyan Anna. 2017. Understanding test debt. *Trends in software testing* (2017), 1–17.
- [13] Edith Tom, Aybüke Aurum, and Richard Vidgen. 2013. An exploration of technical debt. *Journal of Systems and Software* 86, 6 (2013), 1498–1516.