

# Technical Debt's State of Practice on Stack Overflow: a Preliminary Study

Eliakim Gama

State University of Ceará (UECE)  
Fortaleza, Brazil  
eliakim.gama@aluno.uece.br

Matheus Paixao

State University of Ceará (UECE)  
Fortaleza, Brazil  
matheus.paixao@uece.br

Emmanuel Sávio Silva Freire

Federal Institute of Ceará (IFCE)  
Morada Nova, Brazil  
savio.freire@ifce.edu.br

Mariela Inés Cortés

State University of Ceará (UECE)  
Fortaleza, Brazil  
mariela.cortes@uece.br

## RESUMO

**Context:** Previous work have investigated the industry's state of practice on technical debt (TD) by surveying invited practitioners. However, these studies have included only practitioners with a sound understanding of TD, which might incur biases in their findings. One way to reduce this bias is to consider Q&A platforms, such as Stack Overflow (SO), because of containing practitioners with different roles and levels of TD knowledge. Therefore, the discussions on SO can broaden the perception of the state of practice on TD in the industry. **Goal:** To investigate how TD concepts have been discussed by SO's practitioners. **Method:** First, we employed data mining techniques to identify TD-related discussions on SO. Next, based on significance and popularity metrics, we considered 195 discussions to be further investigated. For each discussion, we analyzed its contents to identify types of TD followed by activities, strategies, and tools used in its management. **Results:** We found that code, infrastructure, and architecture debt are the most discussed types of TD on SO. Similarly, TD identification and payment are the most discussed activities. Regarding tools and strategies for TD management, SonarQube and refactoring have been mostly mentioned by practitioners. **Conclusion:** Our findings may indicate new research directions, as well as expanding the empirical knowledge on how the industry has handled TD.

## CCS CONCEPTS

• General and reference → Empirical studies; • Software and its engineering → Maintaining software.

## KEYWORDS

Technical Debt, Stack Overflow, Mining Software Repositories.

### ACM Reference Format:

Eliakim Gama, Emmanuel Sávio Silva Freire, Matheus Paixao, and Mariela Inés Cortés. 2019. Technical Debt's State of Practice on Stack Overflow: a Preliminary Study. In *XVIII Brazilian Symposium on Software Quality*

Publication rights licensed to ACM. ACM acknowledges that this contribution was authored or co-authored by an employee, contractor or affiliate of a national government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.

*SBQS'19, October 28-November 1, 2019, Fortaleza, Brazil*

© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-7282-4/19/10...\$15.00

<https://doi.org/10.1145/3364641.3364668>

(*SBQS'19*), October 28-November 1, 2019, Fortaleza, Brazil. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3364641.3364668>

## 1 INTRODUÇÃO

O termo Dívida Técnica (do inglês *technical debt*) (DT) [7] descreve as consequências em longo prazo ocasionadas por atalhos tomados durante o processo de desenvolvimento de *software* para alcançar objetivos a curto prazo. Esses atalhos podem resultar em documentação desatualizada, testes planejados mas não executados, código complexo que precisa ser reestruturado ou refatorado, dentre outros. O acúmulo desta dívida pode acarretar em mudanças futuras mais complexas e custosas, ou mesmo inviáveis. Mais informações sobre DT podem ser encontradas na Seção 2.

O conceito de DT tem gerado muito interesse nos últimos anos, porém ainda não existe uma clara compreensão do estado da arte por parte da academia e da indústria. Segundo Li *et al.* [11], pesquisadores e *practitioners*<sup>1</sup> percebem o conceito de DT de diferentes formas gerando ambiguidades na interpretação do termo e suas implicações. Nesse sentido, estudos empíricos vêm sendo realizados para obter uma melhor percepção de como *practitioners* entendem os conceitos de DT, seus efeitos em produtos de *software* e metodologias para sua gestão [2, 3, 11].

Vários autores [9, 13, 14] têm se concentrado na consolidação dos diversos conceitos de DT a partir de uma perspectiva acadêmica, identificando tipos, atividades necessárias para o seu gerenciamento e sintomas. No entanto, evidências práticas geralmente são limitadas à aplicação de questionários (*surveys*) a uma amostra de participantes convidados [3] com conhecimento prévio do termo DT e seus conceitos. Assim, as observações extraídas de tais questionários apresentam vieses de pesquisa por não explorarem, por exemplo, *practitioners* que têm problemas de DT mas que ainda não conseguem entendê-los de tal forma [16]. Uma forma de reduzir esses vieses pode ser a utilização de plataformas de perguntas e respostas (Q&A) como fonte de coleta de dados, pois apresentam *practitioners* com diferentes níveis de conhecimento sobre DT.

Nesse sentido, o *Stack Overflow* (SO) é a maior plataforma colaborativa de Q&A voltada para o desenvolvimento de *software* e programação<sup>2</sup>. Seu grupo de usuários varia desde desenvolvedores

<sup>1</sup>Este termo é comumente usado para representar diferentes profissionais de *software*, como desenvolvedores, arquitetos, gerentes etc. A expressão em inglês será usada ao longo do artigo pela sua extensa utilização na literatura relacionada.

<sup>2</sup>Essa informação pode ser encontrada em <https://stackoverflow.com/>

e arquitetos de *software* com anos de experiência até estudantes em formação [18]. Assim, dados e conhecimentos extraídos do SO representam uma população de *practitioners* que não é somente vasta mas também variada. Mais informações sobre SO podem ser encontradas na Seção 2.

O presente trabalho tem como objetivo *investigar como o conceito de DT está sendo discutido pela comunidade de usuários no SO*. Para tanto, foram utilizadas técnicas de mineração de dados para extrair uma amostra estatisticamente significativa de discussões que englobasse DT. Essas discussões foram analisadas por, pelo menos, dois pesquisadores e os seguintes elementos da DT foram identificados: tipo, atividade de gestão, e estratégia e ferramenta para suporte a essas atividades. O projeto experimental é apresentado na Seção 3.

Os principais resultados indicam que o termo DT está sendo discutido no SO, enfatizando dívidas de código, de infraestrutura e de arquitetura. Além disso, as atividades de identificação, pagamento e prevenção foram as mais discutidas em relação à gestão de DT. A ferramenta *SonarQube* e as estratégias refatoração e desenvolvimento orientado por testes (do inglês, *test driven development* (TDD)) foram as mais discutidas como suporte às atividades de gestão de DT. Os resultados são detalhados na Seção 4.

Com isso, busca-se identificar os diversos aspectos da DT a partir das discussões disponíveis no SO para avançar o entendimento dos conceitos de DT ao explorar uma população de *practitioners* que é vasta, internacional e variada, mitigando assim os principais vieses causados por questionários orientados a participantes convidados. Um maior detalhamento das conclusões está disponível na Seção 5.

## 2 REFERENCIAL TEÓRICO

O termo **Dívida Técnica** (DT) descreve as consequências em longo prazo ocasionadas por atalhos tomados durante o processo de desenvolvimento de *software* para alcançar objetivos a curto prazo [7]. DT pode ocorrer em diferentes artefatos durante todo o ciclo de vida de um produto, e pode ter uma natureza diferente dependendo do momento em que é incorrida e a qual atividade está associada. Considerando esses aspectos, a DT é classificada nos seguintes tipos [14]: *design*, código, arquitetura, testes, documentação, defeitos, infraestrutura, requisitos, pessoas, *build*, processo, testes automatizados, usabilidade, serviço e versionamento.

As atividades tipicamente envolvidas na gestão de DT são: prevenção, identificação, monitoramento e pagamento. Estas atividades de gestão da DT são implementadas com uso de ferramentas de apoio, e sua execução é guiada por estratégias.

Além de *surveys* e estudos terciários, plataformas de Q&A têm sido utilizadas como fonte de dados para o entendimento de fenômenos associados ao desenvolvimento de *software*. Dentre elas, o **Stack Overflow** (SO) tem se popularizado entre *practitioners* por possibilitar discussões associadas, principalmente, à fase de desenvolvimento. Perguntas podem ser respondidas por mais de um usuário, caracterizando uma discussão sobre o tema da pergunta. Dado o perfil de usuários do SO [18], as discussões realizadas na plataforma não só representam uma ampla comunidade de *practitioners* como também podem conter informações relevantes para o avanço do conhecimento [4]. Nesse sentido, dados do SO têm sido utilizados para a melhoria de práticas de desenvolvimento de

*software* como identificação automatizada de APIs comumente utilizadas [10], melhoramento automatizado de documentação [19] e melhoria de IDEs [12]. As discussões dos usuários sobre desenvolvimento móvel [15], decisões arquiteturais [17] e microsserviços [6] também foram foco de pesquisas. *Entretanto, apesar da extensa utilização do SO em diversos estudos na literatura, não foram encontrados trabalhos que explorem discussões sobre DT no SO.*

## 3 PROJETO EXPERIMENTAL

Com o objetivo de investigar como o conceito de dívida técnica está sendo discutido pela comunidade de usuários do SO foram formuladas as seguintes Questões de Pesquisa (QP):

- **QP1:** *Dívida técnica está sendo discutida no Stack Overflow?* Esta questão verifica se *practitioners* discutem DT no SO. Essa análise serve como motivação para o desenvolvimento das próximas questões de pesquisa.
- **QP2:** *Quais são os tipos de Dívida Técnica discutidos no Stack Overflow?* O objetivo desta questão é identificar os diferentes tipos de DT discutidos no SO conforme a classificação apresentada por Rios *et al.* [14].
- **QP3:** *Quais são as atividades de gestão de Dívida Técnica discutidas no Stack Overflow?* A finalidade dessa questão é identificar as possíveis atividades de gestão de DT discutidas pelos *practitioners* do SO.
- **QP4:** *Quais são as ferramentas ou estratégias para a gestão de Dívida Técnica discutidas no Stack Overflow?* Esta questão busca identificar ferramentas ou estratégias utilizadas pelos *practitioners* na gestão de DT.

Todos os dados, análises e resultados utilizados e gerados nesse artigo estão disponíveis em um pacote de replicação [8].

### 3.1 Identificação de Discussões Relacionadas à Dívida Técnica no Stack Overflow

Para identificação das discussões relacionadas à DT no SO, foi utilizado o *SOTorrent* [5] (versão 21/06/19), uma base de dados curada por pesquisadores que apresenta dados atuais e históricos sobre discussões no SO. A utilização da base curada do *SOTorrent* possibilitou acesso ao histórico de versões do conteúdo do SO, além de contribuir no processo de coleta, filtragem e manipulação dos dados, o que é inviável através da interface do próprio site do SO. O processo de identificação de discussões relevantes para o estudo consistiu em quatro passos (Veja a Figura 1).

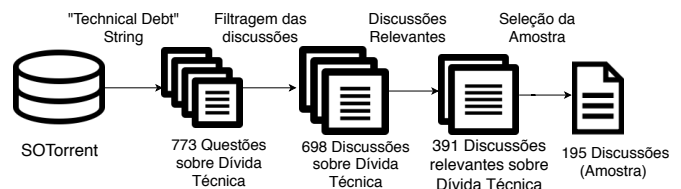


Figura 1: Filtragem de discussões sobre DT no SO

No primeiro passo, foi realizada uma busca pela expressão ‘*technical debt*’ por ser amplamente utilizada tanto por *practitioners* quanto na literatura para se referir à Dívida Técnica. A busca foi

aplicada aos seguintes campos de uma discussão no SO: título da pergunta, corpo da pergunta, *tags* da pergunta e corpo das respostas. Como resultado, foi obtido um total de 773 discussões.

Dado o volume de perguntas no SO, não é incomum que uma pergunta fique sem resposta, ou ainda que o próprio autor da pergunta seja o responsável por sua resposta. Tais casos não caracterizam uma discussão entre *practitioners* [6]. Dessa forma, foi considerada uma **discussão no SO** como sendo uma pergunta que tenha pelo menos uma resposta e o conjunto de respostas apresente um usuário que não seja o autor da pergunta. Assim, o segundo passo da identificação de discussões relevantes no SO aplicou a definição descrita acima e identificou um total de 698 discussões sobre DT.

Nem todas as discussões sobre DT no SO foram relevantes para o presente estudo. Discussões com poucas visualizações e poucas respostas indicam discussões muito específicas que podem não atrair o interesse de *practitioners*. Assim, o terceiro passo para a identificação de discussões relevantes considerou somente as discussões sobre DT mais populares no SO.

A análise de popularidade foi baseada em 5 diferentes métricas: quantidade de visualizações, quantidade de comentários, quantidade de respostas, quantidade de favoritos e *score*. Esta última é uma métrica de popularidade calculada pelo próprio SO. Para cada métrica de popularidade, selecionou-se o conjunto de discussões acima do terceiro quartil da distribuição. Ao final deste processo, foram identificadas as 391 discussões sobre DT mais populares no SO de acordo com as 5 métricas supracitadas. Este quarto e último passo possibilitou a eliminação da quantidade de ruído<sup>3</sup> nos dados selecionados para análise [1, 10].

A metodologia para análise de cada discussão envolveu um conjunto de passos manuais (vide Subseção 3.2). Por ser um estudo preliminar, optou-se por considerar uma amostra estatisticamente significativa das discussões relevantes. Logo, a população de 391 discussões relevantes foi dividida nos seguintes grupos, considerando o local onde o termo '*technical debt*' estava presente:

- **Pergunta:** A expressão '*technical debt*' foi encontrada somente na pergunta (título, corpo ou *tag*). Este grupo obteve 70 discussões e representa 18% da população.
- **Resposta:** A expressão '*technical debt*' foi encontrada somente no corpo das respostas. Este grupo obteve 300 discussões e representa 76% da população.
- **Ambos:** A expressão '*technical debt*' foi encontrada na pergunta e na resposta. Este grupo obteve 21 discussões e representa 6% da população.

A partir da população obtida e considerando o agrupamento realizado no passo anterior, foi extraída uma amostra estratificada com 95% de confiança com uma taxa de erro de 5%. Essa amostra resultou em um total de 195 discussões divididas entre: *Pergunta* - 35 (18%), *Ambos* - 12 (6%), *Resposta* - 148 (76%). Cada uma das 195 discussões contidas na amostra passou pelo processo de análise descrito na seção a seguir.

### 3.2 Análise das Discussões

A metodologia adotada para análise de cada discussão foi composta por três etapas, apresentadas na Figura 2. Os dois primeiros autores

exerceram o papel de pesquisadores, enquanto os dois últimos foram revisores. Inicialmente, foi realizada a **etapa 1**, considerando a análise total da amostra pelos pesquisadores de forma independente a partir da leitura e identificação dos elementos de DT na discussão. Para cada discussão, foi preenchido o *checklist* ilustrado na Tabela 1, onde cada pesquisador indicou informações como tipo de DT, atividade de gestão e ferramenta ou estratégia de gestão que estavam sendo discutidas. Essas informações, quando encontradas, foram registradas de acordo com a percepção de cada pesquisador, tendo como base conceitos apresentados por Rios *et al.* [14].

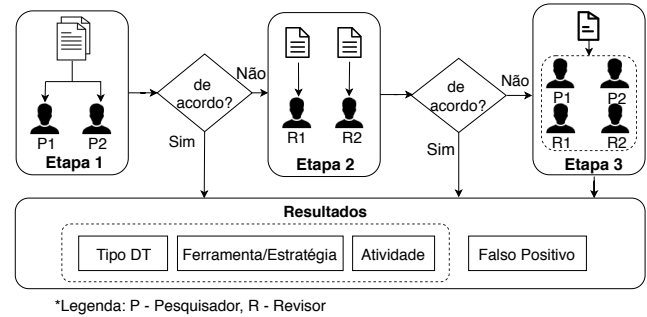


Figura 2: Metodologia para análise das discussões

Tabela 1: Checklist para análise dos elementos de Dívida Técnica em uma discussão do Stack Overflow

#### Perguntas (P)

**P1.** É possível identificar algum **tipo** de Dívida Técnica na discussão? (Sim/Não)

**P1.1.** Se sim, qual/quais?

**P2.** É possível identificar alguma **atividade de gestão** de Dívida Técnica na discussão? (Sim/Não)

**P2.1.** Se sim, qual/quais?

**P3.** É possível identificar alguma **ferramenta ou estratégia** de apoio à gestão de Dívida Técnica na discussão? (Sim/Não)

**P3.1.** Se sim, qual/quais?

**P4.** A discussão é um falso positivo? (Sim/Não)

Vale ressaltar que os elementos referentes à DT evidenciados por [14] poderiam não estar claramente descritos nas discussões. Esse fato ocorreu porque as discussões encontradas surgiram devido à necessidade dos próprios usuários do SO sem a interferência dos pesquisadores desse estudo. Logo, o entendimento e a interpretação do contexto da discussão pelos pesquisadores foram necessários durante o processo de análise. Um exemplo de análise e preenchimento do *checklist* é ilustrado na Figura 3.

De acordo com o *checklist*, foi possível verificar se uma discussão era um falso positivo, caracterizando que a expressão '*technical debt*' foi encontrada na discussão, porém a discussão em si não era de fato relacionada à DT. Para ilustrar essa situação, considere a Figura 4. Pode-se notar que a pergunta é relacionada a um problema de configuração da ferramenta *SonarQube*, no qual o autor incluiu o *log* da ferramenta como forma de contextualizar seu problema. Esse

<sup>3</sup>Ruído é uma solicitação de perda que não reflete nas correlações de dados, porém pode afetar os padrões reais da mineração [21].

### Pergunta

In agile development how do you deal with the "less-well-architected" code that results from a sprint-focussed mindset [closed]

Figura 3: Exemplo de Interpretação das Discussões

log incluía a expressão ‘technical debt’, indicando que tal discussão fosse considerada na análise. Entretanto, ao analisar a discussão, pode-se verificar que ela não está relacionada ao contexto da DT, mas sim à utilização ou à configuração da ferramenta.

the analysis result not being published with 4.1.1 using ANT Task

Figura 4: Exemplo de Falso Positivo

Em seguida, realizou-se a checagem de concordância das informações registradas de forma independente por cada pesquisador. As informações coletadas comuns a ambos pesquisadores foram automaticamente adicionadas aos resultados. As demais passaram por uma segunda etapa de avaliação (etapa 2).

Na **etapa 2**, foi realizada a análise das discussões que obteve informações divergentes na etapa 1. Essas discussões foram divididas entre os dois revisores de forma independente. Após o término da análise, realizou-se uma checagem de concordância entre os resultados obtidos pelos pesquisadores e revisores. No caso de concordância da resposta do revisor com pelo menos um dos pesquisadores, a discussão era adicionada aos resultados. As divergências passaram por uma terceira e última etapa de avaliação (etapa 3).

Na **etapa 3**, todos os pesquisadores e revisores realizaram conjuntamente a análise de cada divergência encontrada na etapa 2 para garantir que a concordância geral da amostra.

Em suma, na etapa 1 houve divergência em 51% das 195 discussões. Desse percentual, 42% foi resolvido na etapa 2, e 9% foi resolvido na etapa 3. Dessa forma, ao final das três etapas, todas as 195 discussões foram devidamente analisadas por pelo menos dois pesquisadores ou revisores.

### 3.3 Ameaças à Validade

As ameaças à validade identificadas foram eliminadas ou mitigadas quando a sua eliminação não foi possível. A seguir, são detalhadas as ameaças seguindo a classificação definida por [20].

- **Validade de Constructo:** Em geral, está relacionada à má definição da base teórica ou à definição do processo de experimentação. No caso, o *checklist* pode não ser suficientemente detalhado para avaliar completamente as discussões e subsidiar a tomada de decisão em relação aos falsos positivos. Esse risco foi reduzido ao se utilizar um referencial teórico baseado em um recente estudo terciário sobre DT [14].
- **Validade Interna:** De forma a evitar resultados tendenciosos na análise das discussões, o processo seguiu três etapas. Na primeira, houve dupla verificação da amostra completa e independente por parte de dois pesquisadores, cujas divergências foram verificadas pelos revisores (segunda etapa). Na terceira etapa, algumas discussões ainda passaram pela revisão conjunta de todos os pesquisadores e revisores envolvidos nesse estudo.
- **Validade Externa:** Em relação à capacidade de generalização das conclusões, a pesquisa foi realizada a partir de uma amostra representativa do SO, a maior comunidade de discussão online utilizada por *practitioners* atualmente. Mesmo assim, sugere-se a não generalização dos resultados.
- **Validade de Conclusão:** A subjetividade envolvida na análise e na interpretação das discussões pode conduzir pesquisadores e revisores a conclusões diferentes, mesmo adotado o mesmo método de análise. Assim, essa ameaça foi minimizada por meio do processo de análise apresentado na Subseção 3.2.

## 4 RESULTADOS E DISCUSSÕES

A seguir são apresentados os principais resultados obtidos a partir da análise realizada. Estes resultados são utilizados para responder às questões de pesquisa propostas no presente artigo.

### 4.1 QP1 - Dívida técnica é discutida no Stack Overflow?

Com o intuito de responder a essa questão, foi elaborada a Tabela 2, indicando a presença de diferentes elementos de DT nas discussões analisadas. Por exemplo, para *Ferramenta*, foram encontradas 17 ocorrências em discussões nas quais algum tipo de ferramenta para gestão de DT era discutida, o que representa 8,71% da amostra. Destas, 4 apresentam a expressão *technical debt* na pergunta, 9 na resposta e 4 em ambas perguntas e respostas.

Pode-se observar que *Atividade* e *Tipo* foram os elementos mais discutidos no SO, com 53,33% e 50,77%, respectivamente. Estes resultados indicam que *practitioners* tendem a discutir diversos aspectos relacionados à DT no SO, o que viabiliza as análises e as questões de pesquisa subsequentes. Entretanto, percebe-se que 28,21% da amostra foi identificada como contendo falso positivos. Logo, a metodologia para identificação de discussões relacionadas à DT no SO deve ser melhorada em trabalhos futuros.

**Tabela 2: Presença de elementos da DT nas discussões**

Elemento DT	Pergunta	Resposta	Ambos	Total*
Atividade	15	80	09	<b>104</b> (53,33%)
Tipo	15	79	05	<b>99</b> (50,77%)
Estratégia	04	16	04	<b>24</b> (12,31%)
Ferramenta	04	09	04	<b>17</b> (8,71%)
Falso Positivo	16	38	01	<b>55</b> (28,21%)

\* Os percentuais foram calculados considerando a amostra de 195 discussões.

#### 4.2 QP2 - Quais são os tipos de Dívida Técnica discutidos no Stack Overflow?

Por meio da Tabela 3, pode-se perceber que 6 dos 15 tipos de DT apresentados por Rios *et al.* [14] são discutidos pelos *practitioners* no SO, sendo encontradas 108 ocorrências de tipo de DT na amostra. Mais especificamente, a dívida de código foi a mais presente entre as discussões, sendo recorrente 54 vezes. Vale ressaltar que esse tipo de DT também é o mais pesquisado pela academia [14]. Além disso, DT de infraestrutura e arquitetura também se destacaram no *ranking* (veja a Tabela 3).

**Tabela 3: Tipos de Dívida Técnica**

Tipo DT	Pergunta	Resposta	Ambos	Total*
Código	11	41	02	<b>54</b> (54,54%)
Infraestrutura	01	17	01	<b>19</b> (19,19%)
Arquitetura	02	13	03	<b>18</b> (18,18%)
Teste	02	10	01	<b>13</b> (13,13%)
Defeito	00	02	01	<b>03</b> (3,03%)
Usabilidade	00	01	00	<b>01</b> (1,01%)

\* Os percentuais foram calculados considerando as 99 discussões conforme a Tabela 2.

Tais resultados podem estar associados ao perfil dos usuários do SO, pois estes estão mais centrados na atividade de desenvolvimento. Entretanto, DT de testes, defeitos e usabilidade foram recorrentes em 17,17% das discussões que contêm tipos de DT, evidenciando que *practitioners* no SO estão interessados nas atividades de testes, nos defeitos e na usabilidade das suas aplicações.

#### 4.3 QP3 - Quais são as atividades de gestão de Dívida Técnica discutidas no SO?

Considerando as 11 atividades de gestão de DT apresentadas por Rios *et al.* [14], foram evidenciadas 9 delas nas discussões analisadas. Foram encontradas 131 ocorrências na amostra. Esse resultado é apresentado na Tabela 4. A atividade mais citada nas discussões foi *Identificação*, que foi encontrada 57 (54,80%) vezes nas discussões. Em seguida, *Pagamento* e *Prevenção* foram citadas 39 (37,50%) e 14 (13,46%) vezes na amostra, respectivamente. Esse *ranking* difere de Rios *et al.* [14], pois consta que as atividades de *Medição*, *Identificação* e *Pagamento* foram as mais pesquisadas, respectivamente.

É importante ressaltar que a atividade *Prevenção* tem sido discutida pelos usuários do SO para evitar aumento da dívida nos seus projetos. Esse resultado demonstra que os *practitioners* estão interessados em técnicas, métodos e boas práticas para prevenir a DT.

**Tabela 4: Atividade de Gestão de Dívida Técnica**

Atividade	Pergunta	Resposta	Ambos	Total*
Identificação	04	46	07	<b>57</b> (54,80%)
Pagamento	07	28	04	<b>39</b> (37,50%)
Prevenção	03	08	03	<b>14</b> (13,46%)
Medição	00	04	04	<b>08</b> (7,69%)
Monitoramento	02	02	03	<b>07</b> (6,73%)
Priorização	00	01	02	<b>03</b> (2,88%)
Documentação	00	00	01	<b>01</b> (0,96%)
Gerenciamento	00	01	00	<b>01</b> (0,96%)
Visualização	00	00	01	<b>01</b> (0,96%)

\* Os percentuais foram calculados considerando as 104 discussões segundo a Tabela 2.

Com isso, pode-se concluir que as discussões do SO englobam tanto as atividades (i) relacionadas à presença (*Identificação* e *Pagamento*) de DT quanto (ii) para evitar (*Prevenção*) a DT.

#### 4.4 QP4 - Quais são as ferramentas ou estratégias de gestão de Dívida Técnica discutidas no Stack Overflow?

Conforme a Tabela 2, foram encontradas 17 e 24 discussões sobre ferramentas e estratégias de gestão de DT, respectivamente, na amostra. Destas, *SonarQube* foi mencionada 11 (64,71%) vezes. A ferramenta *FindBugs* foi citada 3 (17,65%) vezes nas discussões. Além disso, outras 9 ferramentas foram citadas nas discussões, mas com baixa ocorrência. Esses resultados são apresentados na Tabela 5.

**Tabela 5: Ferramentas de Gestão de Dívida Técnica**

Ferramenta	Total*
<i>SonarQube</i>	<b>11</b> (64,71%)
<i>FindBugs</i>	<b>03</b> (17,65%)
<i>Code Maid</i>	<b>01</b> (5,88%)
<i>Checkstyle</i>	<b>01</b> (5,88%)
<i>CRAP</i>	<b>01</b> (5,88%)
<i>Internal tool</i>	<b>01</b> (5,88%)
<i>FxCop</i>	<b>01</b> (5,88%)
<i>IntelliJ</i>	<b>01</b> (5,88%)
<i>Lattix</i>	<b>01</b> (5,88%)
<i>PMD</i>	<b>01</b> (5,88%)
<i>Team Foundation Server</i>	<b>01</b> (5,88%)

\* Os percentuais foram calculados considerando as 17 discussões conforme a Tabela 2.

Ao comparar esses resultados com os evidenciados por Rios *et al.* [14], pode-se verificar que apenas as ferramentas *SonarQube*, *FindBugs*, *FxCop* e *PMD* foram encontradas em ambos os trabalhos. Portanto, as demais ferramentas podem ser exploradas pela comunidade acadêmica para verificar o suporte à gestão de DT.

Sobre as estratégias discutidas no SO pelos seus usuários para lidar com DT, foi identificado que as mais citadas foram refatoração e TDD, com 10 (41,67%) e 9 (37,50%) citações, respectivamente.

Além disso, outras estratégias também foram encontradas, totalizando 9 ocorrências. Esses resultados são encontrados na Tabela 6. Mais especificamente, as discussões encontradas sobre refatoração indicam que algumas vezes utiliza-se esse termo para identificar tanto simples quanto complexas alterações no código-fonte. Vale ressaltar que não foi feita nenhuma diferenciação na análise do presente trabalho.

**Tabela 6: Estratégias de Gestão da Dívida Técnica**

Estratégia	Total*
Refatoração	10 (41,67%)
Test driven development (TDD)	09 (37,50%)
Limpeza de código inútil	02 (8,33%)
Sprints dedicados ao pagamento de DT	02 (8,33%)
Integração contínua	01 (4,17%)
Priorização de correção de bugs	01 (4,17%)
Princípios SOLID	01 (4,17%)
SQUALE	01 (4,17%)
Análise estática de código	01 (4,17%)

\* Os percentuais foram calculados considerando as 24 discussões conforme a Tabela 2.

As estratégias Refatoração e SQUALE foram as únicas presentes tanto nas discussões analisadas quanto no trabalho de Rios *et al.* [14]. Logo, as outras estratégias apresentadas na Tabela 6 podem ser consideradas como ponto de partida para fomentar estratégias mais robustas integrando todas as atividades de gestão de DT.

## 5 CONSIDERAÇÕES FINAIS

O presente trabalho tem como objetivo *investigar como o conceito de dívida técnica está sendo discutido pela comunidade de practitioners do SO*. Considerando técnicas de mineração de dados, os resultados encontrados caracterizam a DT em termos de tipo, atividade de gestão e estratégias e ferramentas para suporte a essas atividades.

Os resultados também revelam novas possibilidades de pesquisa englobando DT, como (i) propor estratégias para lidar com dívidas de testes, defeitos e usabilidade, (ii) investigar mais efetivamente as atividades relacionadas à prevenção de DT, e (iii) verificar como outras ferramentas e estratégias estão sendo utilizadas pelos *practitioners* para dar suporte às atividades de gestão de DT.

Como trabalhos futuros, sugere-se: (i) melhorar o processo de identificação de discussão relevantes, minimizando a presença de falsos positivos, (ii) analisar detalhadamente a população completa de discussões e não somente uma amostra significativa, (iii) incluir outros elementos da DT, como causas, efeitos e sintomas, (iv) utilizar outras plataformas de Q&A direcionadas ao desenvolvimento de *software*, por exemplo, a *Software Engineering Stack Exchange*<sup>4</sup>, e (v) conduzir um *survey* para caracterizar o estado da arte do conhecimento sobre DT na indústria.

## AGRADECIMENTOS

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

<sup>4</sup><https://softwareengineering.stackexchange.com/>

## REFERÊNCIAS

- [1] M. Ahasanuzzaman, M. Asaduzzaman, C. K. Roy, and K. A. Schneider. 2016. Mining duplicate questions in stack overflow. in *Proceedings of the 13th International Workshop on Mining Software Repositories - MSR* (2016), 402–412.
- [2] Nicolli S.R. Alves, Thiago S. Mendes, Manoel G. de Mendonça, Rodrigo O. Spínola, Forrest Shull, and Carolyn Seaman. 2016. Identification and management of technical debt: A systematic mapping study. *Information and Software Technology* 70 (2016), 100 – 121.
- [3] Nicolli Souza Rios Alves, Rodrigo Oliveira Spínola, Manoel G. Mendonça, and Carolyn B. Seaman. 2018. 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, ESEM 2018, Oulu, Finland, October 11–12, 2018*. 39:1–39:10.
- [4] Ashton Anderson, Daniel Huttenlocher, Jon Kleinberg, and Jure Leskovec. 2012. Discovering value from community activity on focused question answering sites. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '12*. ACM Press.
- [5] Sebastian Baltes, Christoph Treude, and Stephan Diehl. 2019. Sotorrent: Studying the origin, evolution, and usage of stack overflow code snippets. In *Proceedings of the 16th International Conference on Mining Software Repositories*. IEEE Press.
- [6] Alan Bandeira, Carlos Alberto Medeiros, Matheus Paixao, and Paulo Henrique Maia. 2019. We Need to Talk about Microservices: an Analysis from the Discussions on StackOverflow. *International Conference on Mining Software Repositories* 5 (2019), 255–259.
- [7] Ward Cunningham. 1992. The WyCash Portfolio Management System. In *Addendum to the Proceedings on Object-oriented Programming Systems, Languages, and Applications (Addendum)*. 29–30.
- [8] Eliakim Gama, Emmanuel Sávio Silva Freire, Matheus Paixao, and Mariela Inés Cortés. 2019. Replication package for the paper: “Technical Debt’s State of Practice on Stack Overflow: a Preliminary Study”. <https://zenodo.org/record/3383148#XWvH8HVKhE>
- [9] M. Firdaus Harun and Horst Lichter. 2015. Towards a Technical Debt Management Framework based on Cost-Benefit Analysis. In *Proceedings of The Tenth International Conference on Software Engineering Advances*. 70–73.
- [10] D. Kavaler, D. Posnett, C. Gibler, H. Chen, P. Devanbu, and V. Filkov. 2013. Using and asking: Apis used in the android market and asked about in stackoverflow. in *International Conference on Social Informatics* (2013), 402–418.
- [11] Zengyang Li, Paris Avgeriou, and Peng Liang. 2015. A Systematic Mapping Study on Technical Debt and Its Management. *J. Syst. Softw.* 101, C (March 2015).
- [12] Luca Ponzanelli, Gabriele Bavota, Massimiliano Di Penta, Rocco Oliveto, and Michele Lanza. 2014. Mining StackOverflow to turn the IDE into a self-confident programming prompter. In *Proceedings of the 11th Working Conference on Mining Software Repositories*. ACM, 102–111.
- [13] Narayan Ramasubbu and Chris Kemerer. 2018. Integrating Technical Debt Management and Software Quality Management Processes: A Framework and Field Tests. In *Proceedings of the 40th International Conference on Software Engineering*.
- [14] Nicolli Rios, Manoel Gomes de Mendonça Neto, and Rodrigo Oliveira Spínola. 2018. A tertiary study on technical debt: Types, management strategies, research trends, and base information for practitioners. *Information and Software Technology* 102 (2018), 117 – 145.
- [15] Christoffer Rosen and Emad Shihab. 2016. What are mobile developers asking about? A large scale study using stack overflow. *Empirical Software Engineering* 21, 3 (jun 2016), 1192–1223.
- [16] Per Runeson and Martin Höst. 2009. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering* 14, 2 (apr 2009), 131–164.
- [17] Mohamed Soliman, Matthias Galster, Amr R. Salama, and Matthias Riebisch. 2016. Architectural knowledge for technology decisions in developer communities: An exploratory study with stackoverflow. *Working IEEE/IFIP Conference on Software Architecture (WICSA)* 6 (2016), 128–133.
- [18] StackOverflow. [n.d.]. Stack Overflow Annual Developer Survey. <https://insights.stackoverflow.com/survey/> Acessado: 2019-07-23.
- [19] Christoph Treude and Martin P Robillard. 2016. Augmenting api documentation with insights from stack overflow. In *2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE)*. IEEE, 392–403.
- [20] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. 2012. *Experimentation in software engineering*. Springer Science & Business Media.
- [21] Zhu Xudong, Meng Xiaoxuan, Xiao Wenshu, Ke Jian, and Xu Lu. 2009. Z-Miner: Noise Filter In Mining Frequent Access Patterns. *International Conference on Advanced Communication Technology* 05 (2009), 591–595.