UNIVERSIDADE ESTADUAL DO CEARÁ

CENTRO DE CIÊNCIAS E TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

MESTRADO ACADÊMICO EM CIÊNCIA DA COMPUTAÇÃO

CARLOS ALBERTO FERNANDES MEDEIROS

UMA ANÁLISE DO USO DE TÉCNICAS E FERRAMENTAS DE ENGENHARIA DIRIGIDA POR MODELOS A PARTIR DE PLATAFORMAS DE PERGUNTAS E RESPOSTAS NO ÂMBITO DA ENGENHARIA DE SOFTWARE

FORTALEZA – CEARÁ

2019

CARLOS ALBERTO FERNANDES MEDEIROS

UMA ANÁLISE DO USO DE TÉCNICAS E FERRAMENTAS DE ENGENHARIA DIRIGIDA POR MODELOS A PARTIR DE PLATAFORMAS DE PERGUNTAS E RESPOSTAS NO ÂMBITO DA ENGENHARIA DE SOFTWARE

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Estadual do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Ciência da Computação

Orientador: Prof. PhD. Paulo Henrique Mendes Maia

Co-Orientador: Prof. PhD. Matheus Henrique Esteves Paixão

FORTALEZA – CEARÁ

2019

Dados Internacionais de Catalogação na Publicação

Universidade Estadual do Ceará

Sistema de Bibliotecas

Medeiros, Carlos Alberto Fernandes.
Uma análise do uso de técnicas e ferramentas de engenharia dirigida por modelos a partir de plataformas de perguntas e respostas no âmbito da engenharia de software [recurso eletrônico] / Carlos Alberto Fernandes Medeiros. - 2019.
1 CD-ROM: il.; 4 ¾ pol.

CD-ROM contendo o arquivo no formato PDF do trabalho acadêmico com 84 folhas, acondicionado em caixa de DVD Slim (19 x 14 cm x 7 mm).

Dissertação (mestrado acadêmico) - Universidade Estadual do Ceará, Centro de Ciências e Tecnologia, Mestrado Acadêmico em Ciência da Computação, Fortaleza, 2019.
Área de concentração: Engenharia de Software.
Orientação: Prof. Dr. Paulo Henrique Mendes Maia.
Coorientação: Prof. Dr. Matheus Henrique Esteves Paixão.

1. Engenharia Dirigida por Modelos (MDE). 2. Mineração de repositórios de software. 3. Stack Overflow. 4. Software Engineering Stack Exchange. I. Título.

CARLOS ALBERTO FERNANDES MEDEIROS

UMA ANÁLISE DO USO DE TÉCNICAS E FERRAMENTAS DE ENGENHARIA DIRIGIDA POR MODELOS A PARTIR DE PLATAFORMAS DE PERGUNTAS E RESPOSTAS NO ÂMBITO DA ENGENHARIA DE SOFTWARE

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Estadual do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Ciência da Computação

Aprovada em: 18/09/2019

BANCA EXAMINADORA

Prof. PhD. Paulo Henrique Mendes Maia (Orientador)
Universidade Estadual do Ceará – UECE

Prof. PhD. Matheus Henrique Esteves Paixão (Co-Orientador)
Universidade Estadual do Ceará - UECE

Profa. Dra. Mariela Inés Cortés
Universidade Estadual do Ceará - UECE

Prof. Dr. João Bosco Ferreira Filho
Universidade Federal do Ceará - UFC

Dedico este trabalho aos meus pais:
Maria Aparecida (*Dona Cida*), pelos ensinamentos, amor, compreensão e orações durante minha caminhada;
Francisco Carlos (*Seu Carlos*), pelo apoio, segurança, amor e sua presença na minha vida.

AGRADECIMENTOS

Ao Deus que me criou, pois sei que tudo que tenho vem Dele, através do seu filho Jesus e da poderosa intercessão de Maria Santíssima a quem recorri inúmeras vezes buscando paz e força durante esta caminhada.

Aos meus pais, pelo exemplo de pessoas humildes e trabalhadoras. Obrigado por compartilharem ensinamentos e aprendizados que me acompanham hoje e sempre.

À minha esposa, Lueuda, pela paciência, companheirismo e apoio incondicional nesta jornada.

Ao meu filho, Pedro Bruno, razão do meu viver, por me ensinar o sentido da vida.

Ao meu orientador, professor Paulo Henrique, pelas palavras de apoio nos momentos difíceis, por ter acreditado em mim e ter compartilhado seus conhecimentos durante minha jornada no mestrado. Meus sinceros agradecimentos ao senhor.

Ao meu co-orientador, professor Matheus Paixão, por partilhar seu tempo, conversas e ensinamentos visando o melhor para este trabalho. Muito obrigado.

Aos meus amigos e companheiros do GESAD, de modo especial: Adson, Alan, Lucas, Matheus, Davi e Léo (agregado do lab). Obrigado por tornarem essa caminhada mais agradável.

À professora Mariela, por ter aceitado o convite para esta banca, por ler e colaborar com o crescimento deste singelo trabalho.

Ao professor Bosco, pela disponibilidade de aceitar o convite para participar da banca deste humilde trabalho.

A todo o corpo docente do Programa de Pós-graduação em Ciências da Computação, pela seriedade e dedicação ao trabalho.

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

‘Pois que aproveitará ao homem ganhar o mundo inteiro, se vier a perder a sua alma?”

(Marcos 8:36)

RESUMO

Engenharia Dirigida por Modelos (*Model-driven Engineering* - MDE) é uma abordagem de Engenharia de Software que considera modelos não apenas como artefatos de documentação, mas como artefatos primários que podem ser utilizados em qualquer domínio de aplicação. MDE tem ganhado notoriedade tanto na academia como na indústria pelo modo que trata a crescente complexidade do software moderno. Apesar disso, a MDE não é amplamente adotada no desenvolvimento de sistemas. Diante dessa problemática, é preciso investigar e entender como esta abordagem tem sido debatida por desenvolvedores. Com base nesse cenário, um estudo foi conduzido analisando as discussões entre os desenvolvedores em fóruns que abrangem assuntos no contexto da Engenharia de Software. A partir do conjunto de dados fornecido pelo *Stack Exchange* das suas plataformas *Stack Overflow* e *Software Engineering Stack Exchange*, alinhado com técnicas de mineração, foi possível indentificar o que os desenvolvedores têm discutido sobre MDE envolvendo conceitos teóricos, discussões técnicas e questões sobre a utilidade da referida abordagem.

Palavras-chave: Engenharia Dirigida por Modelos (MDE). Mineração de repositórios de software. Stack Overflow. Software Enginnering Stack Exchange.

ABSTRACT

Model Driven Engineering (MDE) is a Software Engineering approach that considers models not only as documentation artifacts, but as primary artifacts that can be used in any application domain. MDE has gained notoriety in both academia and industry for its take on of the growing complexity of modern software. Nevertheless, MDE has not been largely adopted in software development. Furthermore, it is necessary to investigate whether this approach has been debated by the developers. Based on this scenario, a study was conducted to analyze the discussions between developers in forums that cover issues in the context of Software Engineering. Using the dataset from the Stack Overflow and Software Engineering Stack Exchange platforms, it was possible to identify what the developers have been discussing about MDE according to theoretical concepts, technical discussions and the utility of this approach.

Keywords: Model-Driven Engineering (MDE). Mining Software Repositories. Stack Overflow, Software Engineering Stack Exchange.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

CIM	Computation-Independent Model
CSV	Comma Separeted Values
DSL	Domain Specific Languages
DSML	Domain Specific Modeling Language
EBNF	Extended Backus-Naur form
ISM	Implementation-Specific Model
LDA	Latent Dirichlet Allocation
M2M	Model To Model
M2T	Model To Text
MDA	Model-driven Architecture
MDD	Model-driven Development
MDE	Model-Driven Engineering
MSR	Mining Software Repositories
NLP	Natural Language Processing
OMG	Object Management Group
PIM	Platform-Independent Model
PSM	Platform-Specific Model
SE	Software Engineering Stack Exchange
SO	Stack Overflow
TI	Tecnologia da Informação
XML	Extensible Markup Language

SUMÁRIO

1 INTRODUÇÃO

Na busca por novos conhecimentos ou esclarecimentos, o ser humano pode utilizar de sensações ou manifestações imediatas para entender eventos ou novos fenômenos. Conforme Brambilla, Cabot e Wimmer (2012), a mente humana continuamente retrabalha a realidade aplicando processos cognitivos que alteram a sua percepção. Dentre esses processos, a abstração é um dos mais proeminentes. Segundo Sayão (2001), quatro abstrações semânticas são comumente utilizadas: generalização, agregação, classificação e associação, sendo que os três primeiros fazem parte do comportamento natural que a mente humana é capaz de desempenhar. Além disso, Brambilla, Cabot e Wimmer (2012) afirmam que a abstração é amplamente utilizada em ciência e tecnologia, onde é frequentemente referida como modelagem.

De acordo com Giere (2002), um modelo pode ser definido informalmente como uma representação simplificada ou parcial da realidade. Modelos são utilizados todos os dias em diversas áreas visando auxiliar os profissionais a entenderem problemas complexos e como eles podem solucioná-los. Neste sentido, engenheiros de software também utilizam modelos, por exemplo, para melhor entenderem os requisitos de um software.

Conforme Martínez, Cachero e Meliá (2013), no âmbito da Engenharia de Software, defende-se o uso de modelos buscando melhorar as práticas no desenvolvimento de software. Uma das justificativas para isso é que modelos aumentam o nível de abstração no qual os problemas são tratados e, portanto, auxiliam os desenvolvedores a gerenciar a complexidade do software. Além disso, no entendimento de Silva (2015), modelos permitem compartilhar conhecimento e uma visão comum entre *stakeholders* técnicos e não técnicos, facilitando e promovendo a comunicação entre eles.

Diante da crescente complexidade e pervasividade do software na sociedade, é necessário que os desenvolvedores possam utilizar meios para tratar e gerenciar tal cenário. Neste contexto, Kramer (2007) afirma que é preciso desenvolver software recorrendo a métodos apropriados de abstração. De acordo com Hutchinson, Whittle e Rouncefield (2014), o estado da arte em abstração de software é a Engenharia Dirigida por Modelos (*Model-Driven Engineering-MDE*), a qual utiliza, de modo sistemático, modelos como artefatos primários no processo de Engenharia de Software.

Segundo Cuadrado, Izquierdo e Molina (2014), MDE emergiu como uma nova disciplina de Engenharia de Software que enfatiza o uso de modelos para aperfeiçoar a produtividade e alguns aspectos da qualidade do software tais como manutenibilidade e interoperabilidade.

MDE tem ganhado notoriedade tanto na academia como na indústria pelo modo que trata a crescente complexidade do software moderno (HUTCHINSON; WHITTLE; ROUNCEFIELD, 2014). Além disso, de acordo com Cuadrado, Izquierdo e Molina (2014), as técnicas fornecidas pela MDE podem ser aplicadas não apenas visando o desenvolvimento de novas aplicações, mas também a reengenharia de sistemas legados e configuração dinâmica e automatizada de sistemas em execução. Ainda sobre essa discussão, vale ressaltar que, na compreensão de Ciccozzi *et al.* (2017), abstração, automação e análise são aspectos específicos da MDE que auxiliam a lidar com a complexidade e heterogeneidade dos sistemas, expressando conceitos específicos de domínio e fornecendo suporte à comunicação dos *stakeholders*.

Segundo Hutchinson *et al.* (2011), MDE tem sido adotada com sucesso em diversos setores da indústria de software, tais como automotivo, aeroespacial, telecomunicações e sistemas de informações. Apesar disso, a MDE não foi amplamente adotada no desenvolvimento de sistemas (MUSSBACHER *et al.*, 2014). Acerca deste fato, Maia *et al.* (2016) destacam que um dos motivos está na forma como a MDE tem sido ensinada no meio acadêmico. Esse fato é corroborado por Whittle, Hutchinson e Rouncefield (2014), salientando que os conceitos de MDE são ensinados de forma fragmentada e em momentos distintos ao longo dos cursos de Computação. Além dos problemas envolvendo o ensino de MDE, Vallecillo (2015) cita barreiras culturais como a ausência de habilidade e experiência dos desenvolvedores, além da falta de informações adequadas sobre conceitos, objetivos, ferramentas e casos reais envolvendo a aplicação satisfatória da abordagem MDE na indústria.

Diante da rápida evolução da tecnologia, linguagens e plataformas, os desenvolvedores recorrem aos fóruns baseados em perguntas e respostas para se manterem atualizados, buscarem auxílio técnico ou obterem respostas eficientes. Assim, esses fóruns se tornam um repositório de conhecimento no âmbito do desenvolvimento de software. Um dos processos que auxilia na análise de dados relacionados às práticas de desenvolvimento de software é a mineração de repositório de software (*Mining Software Repositories*) (THOMAS; HASSAN; BLOSTEIN, 2014). Adicionalmente, mineração de repositório de software analisa e cruza dados disponíveis nesses repositórios para descobrir informações relevantes (HASSAN, 2008).

Diversas técnicas de mineração de repositório têm sido empregadas no contexto de fóruns baseados em perguntas e respostas (BALTES *et al.*, 2018; Zhang *et al.*, 2018). Algumas dessas técnicas envolvem: identificar os assuntos a partir de *Tags*, utilizar bases de dados que armazenem o histórico das discussões bem como o seu conteúdo, utilização de métricas de popularidade para identificar o interesse da comunidade em assuntos específicos, dentre outros.

Diante deste cenário, este trabalho pretende, recorrendo à técnicas de mineração de repositório, identificar as principais questões discutidas pelos desenvolvedores sobre MDE, incluindo dúvidas conceituais e técnicas a partir das discussões sobre o tema nas plataformas de perguntas e respostas *Stack Overflow*[1] (SO) e *Software Engineering Stack Exchange* (SE)[2]. Conforme Linares-Vásquez, Dit e Poshyvanyk (2013), o *Stack Overflow* é um *website* no formato pergunta e resposta, amplamente utilizado por desenvolvedores para encontrar e fornecer respostas a questões técnicas voltadas ao desenvolvimento de software. Adicionalmente, segundo Verma, Sardana e Lal (2019), o *Software Engineering Stack Exchange* engloba os principais tópicos sobre Engenharia de Software.

Apesar de existirem outros trabalhos que relatam problemas no uso de MDE na indústria (HUTCHINSON *et al.*, 2011; VETRO; BOHM; TORCHIANO, 2015; TOMASSETTI *et al.*, 2012) e trabalhos que utilizam o *Stack Overflow* como fonte de informação para estudar fenômenos em diferentes áreas (BAJAJ; PATTABIRAMAN; MESBAH, 2014; BARUA; THOMAS; HASSAN, 2014; KOCHHAR, 2016; VILLANES *et al.*, 2017), não foi encontrado nenhum trabalho que realize análise nas discussões do *Stack Overflow* ou do *Software Engineering Stack Exchange* para entender as dificuldades teóricas e práticas dos desenvolvedores que utilizam MDE, sendo esta uma das contribuições deste trabalho.

O presente estudo é norteado pelas seguintes questões de pesquisa:

- **QP1:** Quais temas relacionados a MDE os desenvolvedores têm discutido nas plataformas de perguntas e respostas em Engenharia de Software?
- **QP2:** Quais são os domínios que MDE está sendo empregada a partir das discussões identificadas?
- **QP3:** Quais são as principais ferramentas MDE discutidas pelos desenvolvedores?
- **QP4:** Quais temas sobre MDE atraíram mais a atenção dos desenvolvedores?

1.1 OBJETIVOS

As seguintes subseções descrevem os objetivos que almeja-se ser alcançados ao término deste trabalho.

[1] https://stackoverflow.com/
[2] https://softwareengineering.stackexchange.com/

1.1.1 Objetivo Geral

O presente trabalho tem como objetivo principal realizar uma análise das discussões entre os desenvolvedores em plataformas de perguntas e respostas de Engenharia de Software elencando práticas e temáticas discutidas pela comunidade no uso de MDE.

1.1.2 Objetivos Específicos

Para atingir o objetivo geral, os seguintes objetivos específicos devem ser alcançados:

a) Extrair as discussões dos desenvolvedores a partir do *dataset* fornecido pelo *Stack Exchange* utilizando técnicas de mineração de dados;
b) Definir uma metodologia para extrair discussões válidas sobre MDE no *Stack Overflow* e no *Software Engineering Stack Exchange*;
c) Realizar uma categorização das discussões do *Stack Overflow* e do *Software Engineering Stack Exchange* no âmbito da MDE;
d) Responder as questões de pesquisa a partir dos dados coletados.

1.2 ORGANIZAÇÃO DO TRABALHO

Este capítulo apresentou o contexto que norteou a condução desta pesquisa de mestrado, bem como os objetivos esperados. O restante deste documento é organizado da seguinte forma:

- **Capítulo 2 - Fundamentação Teórica:** Discute os conceitos intrínsecos deste trabalho, como abordagens dirigidas por modelo e mineração de repositórios de software;
- **Capítulo 3 - Trabalhos Relacionados:** Aborda os trabalhos encontrados na literatura relacionados com a presente dissertação;
- **Capítulo 4 - Metodologia:** Discorre sobre a metodologia que guia esta pesquisa, explicando cada fase utilizada para reunir os dados necessários que servirão de base para análise;
- **Capítulo 5 - Resultados:** Apresenta os resultados alcançados com base na interpretação dos dados gerados a partir das discussões nas plataformas *Stack Overflow* e *Software Engineering Stack Exchange*;
- **Capítulo 6 - Conclusão:** Compreende as conclusões referentes ao estudo reali-

zado neste trabalho. Além disso, são abordadas às contribuições, limitações e trabalhos futuros desta pesquisa.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo descreve com mais detalhes os conteúdos abordados nesta dissertação. É apresentada uma visão geral sobre os conceitos de Engenharia Dirigida por Modelos e aspectos relevantes que apoiam essa abordagem. Além disso, são apresentados conceitos importantes ligados à mineração de repositórios de software. Por fim, os repositórios que serão analisados são apresentados.

2.1 ENGENHARIA DIRIGIDA POR MODELOS

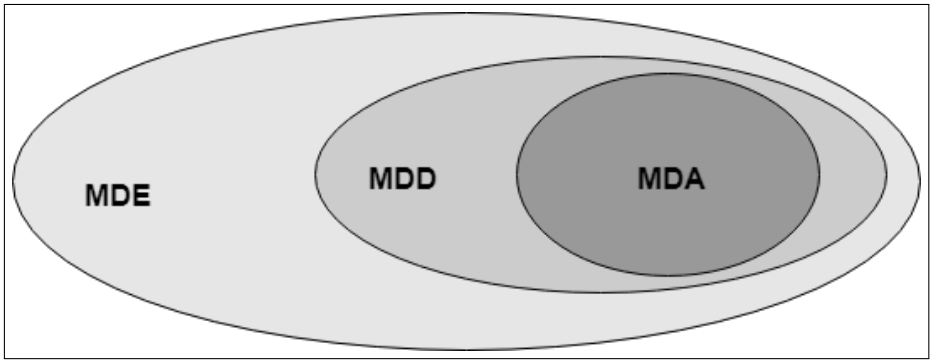
Segundo Silva (2015), Engenharia Dirigida por Modelos é uma abordagem de Engenharia de Software que considera modelos não apenas como artefatos de documentação, mas como artefatos primários, onde os modelos podem ser utilizados ao longo de todas as disciplinas de engenharia e em qualquer domínio de aplicação. Conforme France e Rumpe (2007), MDE estabelece meios para usar os modelos na automação de tarefas inerentes ao desenvolvimento de software, tais como, análise, projeto, geração de código, refatoração e tradução a nível de código entre plataformas.

Ganssle (2008) afirma que o software na sociedade tem se tornado cada vez mais complexo. A partir desse cenário, Kramer (2007) discorre que, para gerenciar esta complexidade, é preciso utilizar abstração. Nesse contexto, na perspectiva de Freitas (2016), a MDE tem como objetivo principal prover aos desenvolvedores um aumento no nível da abstração durante o processo de desenvolvimento de software, permitindo criar aplicações independente de plataforma, ficando assim o desenvolvedor livre da complexidade fornecida pelas diversas plataformas de programação.

Dentro do contexto dirigido por modelo, é possível encontrar na literatura a utilização da abreviação MD* (VöLTER, 2009) para englobar: Engenharia Dirigida por Modelos (*Model-driven Engineering* - MDE), Desenvolvimento Dirigido por Modelos (*Model-driven Development* - MDD) e Arquitetura Dirigida por Modelos (*Model-driven Architecture* - MDA). A Figura 1 mostra a relação entre as abordagens de modelagem.

Conforme Silva (2015), MDD é uma abordagem que foca principalmente nas disciplinas de requisitos, análise, projeto e implementação. MDD define diferentes níveis de abstração sobre o sistema em uso de modo a fornecer transformações tanto de modelos para modelos quanto de modelos para código, visando melhorar a produtividade e a qualidade do processo e do sistema desenvolvido. Além disso, Atkinson e Kuhne (2003) afirmam que o aumento da

Figura 1 – Relação entre MDE, MDD e MDA.



Fonte**:** Adaptado de (BRAMBILLA; CABOT; WIMMER, 2012)

produtividade é a principal motivação para as empresas aderirem ao MDD no desenvolvimento de software, pois os desenvolvedores podem agregar mais valor às entregas e diminuir a taxa de artefatos obsoletos.

MDD é uma abordagem centrada na utilização de modelos como artefatos primários no processo de desenvolvimento. Com isso, os desenvolvedores podem abstrair complexidades inerentes às linguagens de programação, focando apenas na complexidade real do problema em um domínio específico (SILVA; BARBOSA; MALDONADO, 2011). Além disso, conforme Nikiforova, Cernickins e Pavlova (2009), MDD permite reduzir os atrasos e os custos de capitalização dos esforços envolvidos no projeto em cada etapa do ciclo de desenvolvimento de software, objetivando automatizar as transições entre estas etapas.

Outra abordagem situada na MD* é a MDA. Segundo Brambilla, Cabot e Wimmer (2012), MDA é uma visão particular do MDD proposta pela *Object Management Group* (OMG)[1]. Portanto, ela é um subconjunto do MDD na qual as linguagens de modelagem e transformação são padronizadas pela própria OMG (WHITTLE; HUTCHINSON; ROUNCEFIELD, 2014). Concebida em 2001, MDA busca estabelecer meios para satisfazer as necessidades da indústria de software.

Guttman e Parodi (2007) definem MDA como sendo uma abordagem holística tendo como objetivo melhorar todo o ciclo de vida da Tecnologia da Informação (TI) - especificação, arquitetura, projeto, desenvolvimento, implantação, manutenção e integração - baseado em modelagem formal. Com base nisso, MDA pode ser utilizada pelos profissionais de TI para evoluir e aperfeiçoar o desenvolvimento de software alinhado a ferramentas e processos em conformidade com padrões estabelecidos pela OMG.

Nikiforova, Cernickins e Pavlova (2009) afirmam que um dos aspectos-chave da

[1] https://www.omg.org/

abordagem MDA é reconhecer que transformações podem ser aplicadas em descrições abstratas a fim de adicionar detalhes, refinamentos nessas descrições ou mesmo fazer uma conversão entre diferentes representações. Além disso, o nível de abstração dos modelos pode variar dependendo dos seus objetivos. Brambilla, Cabot e Wimmer (2012) apresentam os níveis de abstração definidos pela MDA:

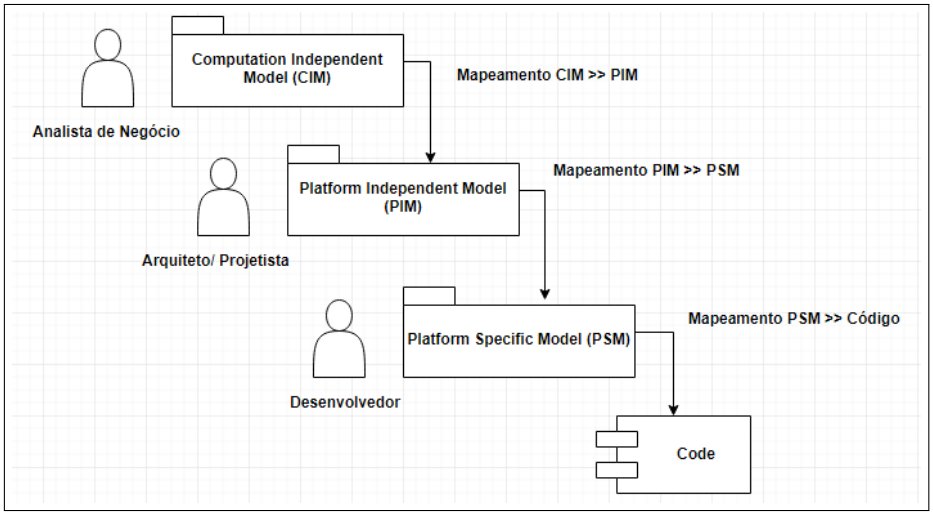
- Modelo Independente de Computação (*Computation-Independent Model* - CIM): Nível de modelagem mais abstrato que se concentra no contexto, requisitos e propósitos da solução sem levar em consideração implicações computacionais. Portanto, ele apresenta exatamente o que o sistema deverá fazer. O CIM também pode ser referenciado como Modelo de Negócio por se utilizar de um vocabulário acessível aos *stakeholders*, sejam eles técnicos ou não;
- Modelo Independente de Plataforma (*Platform-Independent Model* - PIM): Este nível descreve o comportamento e a estrutura da aplicação sem especificar detalhes de tecnologia;
- Modelo Específico de Plataforma (*Platform-Specific Model* - PSM): Deve conter todas as informações definidas no PIM mais a especificação de como o sistema deverá atuar em uma determinada plataforma ou linguagem de programação.

Em adição aos níveis de abstração citados anteriormente, é possível encontrar na literatura mais um nível denominado de Modelo Específico de Implementação (*Implementation-Specific Model* - ISM), que descreve detalhes em termos de código (KRIOUILE *et al.*, 2014). A relação entre os níveis de abstração definidos pela MDA e os respectivos participantes no processo é ilustrado na Figura 2.

A Figura 2 descreve o processo de transformação entre modelos iniciando do nível mais abstrato até o nível mais concreto. O processo se inicia com a transformação do Modelo Independente de Computação (CIM) para o Modelo Independente de Plataforma (PIM). Logo em seguida, o Modelo Independente de Plataforma (PIM) é convertido para o Modelo Específico de Plataforma (PSM), que ao final é transformado em código.

No âmbito da abordagem MDE, os autores Brambilla, Cabot e Wimmer (2012) discorrem sobre quatro conceitos que são fundamentais: modelo, sintaxe concreta, transformação de modelos e geração de código. Esses conceitos são abordados com mais detalhes nas seções seguintes.

Figura 2 – Papéis e Processos em MDA.



Fonte: Adaptado de (QUINTERO; ANAYA, 2007)

2.1.1 Modelos

Na literatura é possível encontrar diversas definições para modelo, algumas das quais são destacadas a seguir:

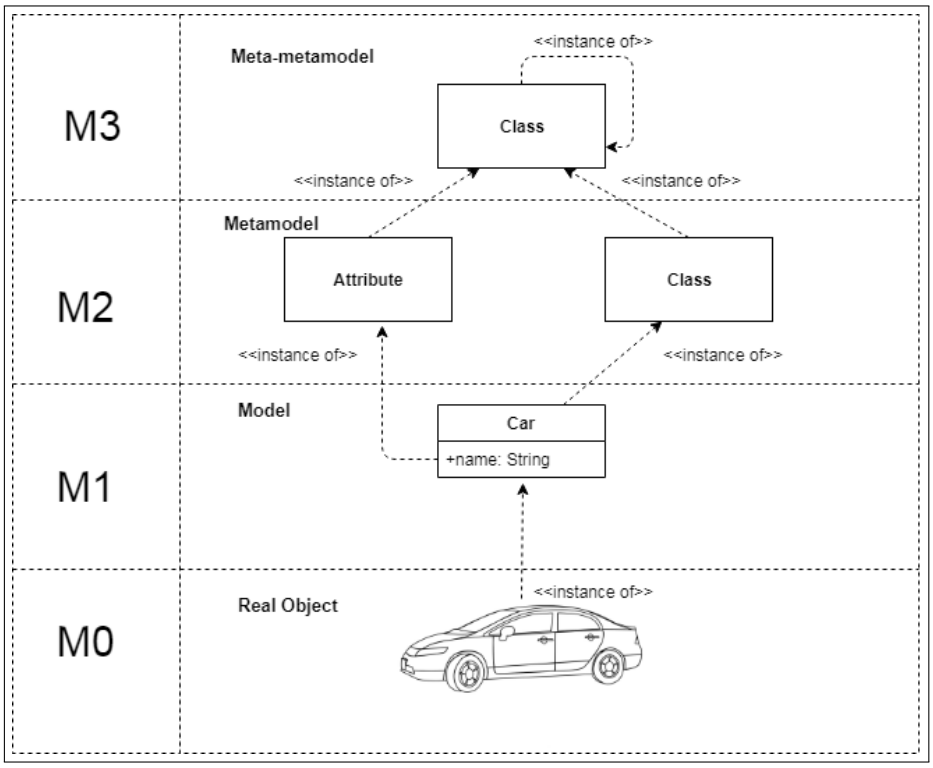
- Kühne (2006) afirma que um modelo é uma abstração de um sistema (real ou baseado em linguagem) que permite que previsões ou inferências sejam feitas;
- Selic (2003) discorre que um modelo é uma representação reduzida de algum sistema que remove detalhes irrelevantes destacando propriedades de interesse a partir de um determinado ponto de vista;
- Seidewitz (2003) refere-se a um modelo como um conjunto de declarações sobre um sistema;
- Silva (2015) define um modelo como um sistema que ajuda a definir e a fornecer respostas do sistema em estudo sem a necessidade de considerá-lo diretamente.

Os modelos ajudam a entender a complexidade de um problema e suas potenciais soluções através da abstração, servindo como a principal fonte de documentação, análise, construção, implantação e manutenção de um sistema.

No contexto da MDE existem tipos de modelos que definem informações de acordo com o seu nível de abstração e especificidade, como ilustrado na Figura 3. No topo da hierarquia há a camada de meta-metamodelo (designada como M3), que é responsável por fornecer uma linguagem para especificar metamodelos; na camada abaixo (designada como M2) se encontra o metamodelo, que é a instanciação do meta-metamodelo, ou seja, cada elemento do metamodelo é

uma instância de um elemento definido no meta-metamodelo; em seguida encontra-se a camada do modelo (M1), criado a partir de um metamodelo e usado para atender as necessidades e interesses do usuário em diferentes domínios de aplicação; finalmente, a camada M0 contém as instâncias reais dos elementos definidos no modelo. Segundo Cetinkaya e Verbraeck (2011), a metamodelagem é um processo que pode ser utilizado para especificar de forma completa Linguagens de Modelagens Específica de Domínio (*Domain Specific Modeling Language* - DSML). O processo de metamodelagem é ilustrado na Figura 4.

Figura 3 – Tipos de Modelos.

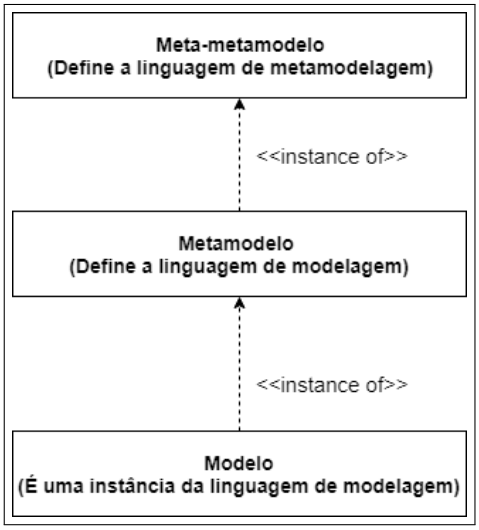


Fonte**:** Elaborado pelo próprio autor

2.1.2 Sintaxe Concreta

Segundo Brambilla, Cabot e Wimmer (2012), metamodelos definem apenas a sintaxe abstrata, mas não a notação concreta da linguagem, isto é, os elementos gráficos e textuais usados para renderizar os elementos do modelo no editor de modelagem. No âmbito das linguagens de modelagem, a sintaxe concreta define as suas representações específicas, cobrindo questões de codificação e/ou aparência visual. Conforme Brambilla, Cabot e Wimmer (2012), a sintaxe concreta pode ser tanto de natureza gráfica ou textual, mas é possível definir ambas em uma mesma linguagem de modelagem.

Figura 4 – Metamodelagem.



Fonte**:** Adaptado de Cetinkaya e Verbraeck (2011)

Moody (2009) afirma que uma linguagem textual permite codificar a informação usando uma sequência de caracteres, enquanto linguagens gráficas codificam a informação utilizando arranjos espaciais de elementos gráficos. Portanto, representações textuais são unidimensionais, ao contrário das linguagens gráficas, que são representações bidimensionais. Por exemplo, os elementos dos diagramas UML estão localizados na modelagem bidimensional.

Como discutido anteriormente, dentro do contexto das linguagens de modelagem, para definir a sintaxe concreta, é possível utilizar duas abordagens: sintaxe gráfica e sintaxe textual. A sintaxe gráfica define diversos elementos gráficos (figuras geométricas, gráficos, ícones) visando representar os conceitos pertencentes ao metamodelo visualmente. Conforme Moody (2009), todos os elementos concretos contidos no metamodelo devem ser mapeados nos elementos gráficos de modo a definir quais propriedades deverão ser exibidas na interface. Deste modo, é possível obter uma representação consistente da sintaxe gráfica.

Segundo Brambilla, Cabot e Wimmer (2012), a sintaxe textual pode ser dividida em duas abordagens: genérica e específica. A genérica pode ser aplicada de maneira abrangente a todo tipo de modelo. A abordagem específica é definida com base em regras similares a *Extended Backus–Naur* - EBNF e, por isso, ela pode ser processada por compiladores para criar processadores de texto da linguagem projetada. As linguagens específicas de domínio (*Domain Specific Languages* - DSL) estão situadas nesse contexto.

Conforme Jouault, Bézivin e Kurtev (2006), DSL é uma linguagem projetada para solucionar um delimitado conjunto de problemas. Uma DSL fornece meios para gerar linguagens

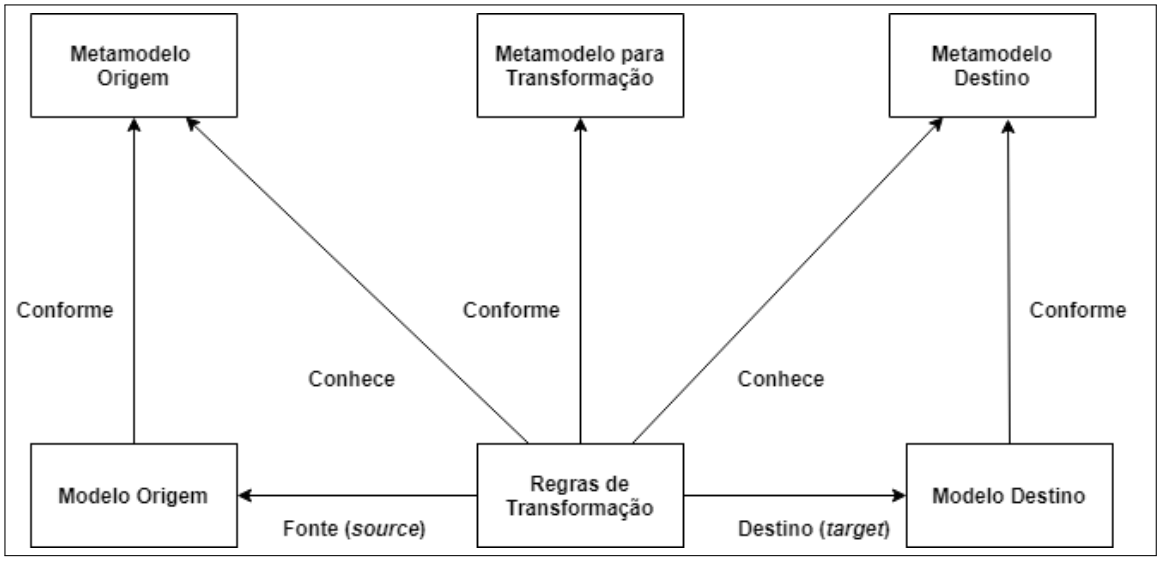
de programação expressando conceitos para atender as necessidades de um domínio de aplicação específico.

2.1.3 Transformação

Além dos modelos, outro conceito de suma importância para MDE são as transformações de modelos. Kleppe, Warmer e Bast (2003) afirmam que esse processo consiste de uma coleção de regras de transformação, ou seja, especificações não ambíguas que definem como (parte de) um modelo pode ser usado para criar (uma parte de) outro modelo. Portanto, uma transformação é a geração automática de um modelo destino a partir de um modelo fonte em conformidade com as regras definidas na transformação. As regras são definidas com base nos metamodelos origem e destino. Desta forma o modelo resultante está em conformidade com o seu metamodelo, prevenindo a transformação de gerar modelos inválidos.

A transformação *Model to Model* (M2M) é um processo no qual um modelo origem é transformado em um modelo destino, sendo estes instâncias ou não do mesmo metamodelo, de acordo com as regras de transformação. De acordo com Brambilla, Cabot e Wimmer (2012), a transformação M2M segue as seguintes etapas: (i) criação dos metamodelos origem e destino; (ii) especificação das regras de transformação; (iii) criação do modelo instância de acordo com o metamodelo origem; (iv) por fim, geração do modelo destino. O processo envolvendo transformação M2M é ilustrado na Figura 5

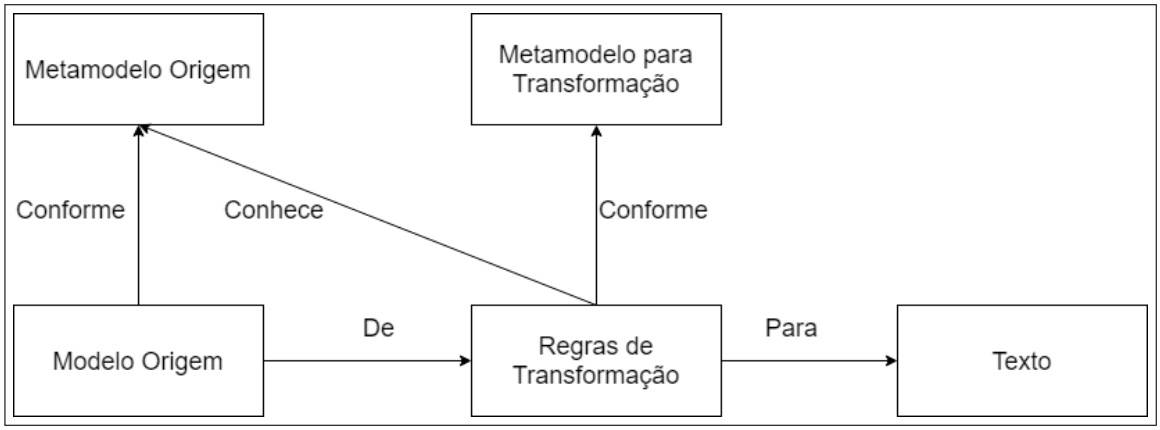
Figura 5 – Transformação entre modelos (M2M).



Fonte: Adaptado de Ma e He (2016)

A transformação *Model to Text* (M2T), conforme Silva (2015), é um processo de geração ou produção de artefatos de software - código-fonte, XML ou outros arquivos textos -, a partir de modelos. O processo envolvendo transformação M2T é ilustrado na Figura 6.

Figura 6 – Transformação de Modelo para Texto (M2T).



Fonte**:** Adaptado de Freitas (2016)

2.2 MINERAÇÃO DE REPOSITÓRIOS DE SOFTWARE

Na concepção de Thomas, Hassan e Blostein (2014), mineração de repositório de software (*Mining Software Repositories* - MSR) é o processo de analisar dados relacionados às práticas de desenvolvimento de software. O autor ressalta que esse é um campo emergente de pesquisa cujo objetivo é refinar as tarefas relacionadas a entender o processo de evolução de software. Acerca dessa discussão, Hassan (2008) afirma que mineração de repositório de software analisa e cruza dados disponíveis nesses repositórios para descobrir informações relevantes sobre projetos e sistemas de software.

No contexto da Engenharia de Software é possível encontrar na literatura trabalhos que, utilizando a mineração e análise de repositórios de softwares, buscam melhorar o desenvolvimento de sistemas através da aplicação desta abordagem: (GODFREY *et al.*, 2009; HASSAN, 2008; MANDELIN *et al.*, 2005; MICHAIL; XIE, 2005; XIE; PEI, 2006). Além disso, estudos mostram que resultados práticos podem ser obtidos ao analisar esse tipo de repositório (TICHY, 2010). Como consequência direta, a equipe de desenvolvimento pode entender melhor seu sistema de modo a aumentar a qualidade do seu produto de maneira significativa e gerenciável. A análise e mineração desse tipo de repositório pode ser realizada observando informações contidas em mudanças nos códigos-fonte, arquivos e metadados contidos em listas de discussões, base de

dados de *bug reports*, logs de execução, dentre outros.

Apesar dos repositórios possuírem dados estruturados, por exemplo, código-fonte, *traces* de execução e logs de mudanças, a maioria dos repositórios possuem dados não estruturados, ou seja, textos em linguagem natural presentes em relatórios que reportam bugs, lista de e-mails, documentos de requisitos, comentários em códigos-fonte e nomes de identificadores. Tal cenário é corroborado com base em dados na literatura que apontam que entre 80% e 85% dos dados presentes em repositórios de software são não estruturados (BLUMBERG; ATRE, 2003) (GRIMES, 2008).

Na compreensão de Thomas (2011), uma das dificuldades inerentes desta área se dá em razão da natureza não estruturada dos dados presentes em grande parte dos repositórios de softwares, exigindo um maior esforço para tentar analisar e realizar a mineração da informação. Hassan (2008), ao discutir essa questão, considera que dados não estruturados são um desafio para área por serem vagos e apresentarem ruído.

Sobre essa problemática, Thomas, Hassan e Blostein (2014) apresentam um exemplo extraído a partir da base de dados de *bug report* do Eclipse, conforme ilustrado na Figura 7. Os autores discorrem que esse dado é vago, pois ele não contém ligação explícita ao código-fonte, tarefa ou tópico ao qual ele se refere. Além disso, frases como "*link creation constraints*", sem informações adicionais, não revelam nenhuma informação relevante, podendo ainda gerar ambiguidades. Nesse exemplo, o dado possui ruído devido à presença de erros ortográficos ("*spashscreen*"), acrônimos não convencionais ("NPE"), além de múltiplas frases usadas para o mesmo conceito ("*unittests*, *unit tests*").

Figura 7 – Exemplo de dado não estruturado

"NPE caused by no spashscreen handler service available" **(#112600)**
"Provide unittests for link creation constraints" **(#118800)**
"jaxws unit tests fail in standalone build" **(#300951)**

Fonte**:** Adaptado de Thomas, Hassan e Blostein (2014)

2.2.1 Repositórios de Software Não Estruturados

Nesta subseção são elencados os principais repositórios de software não estruturados, bem como as suas características. Tais repositórios possuem uma gama de dados que podem ser utilizados pela comunidade científica de modo a entender, por exemplo, questões que norteiam o processo e evolução no desenvolvimento de software.

2.2.1.1 Código-fonte

Código-fonte é a especificação executável que define o comportamento do software. De acordo com Thomas, Hassan e Blostein (2014), um repositório de código-fonte consiste de um número de documentos ou arquivos escritos em uma ou mais linguagens de programação que podem ser agrupados em módulos ou pacotes. Esse tipo de artefato possui dados estruturados como: sintaxe e semântica do programa, fluxo de controle, dentre outros. Por outro lado, é possível notar a presença de uma série de dados não estruturados, como por exemplo, comentários dos desenvolvedores no código, identificadores (nomes de classes, métodos e variáveis), literais em comandos *prints*, dentre outros. A análise dos dados não estruturados contidos no código-fonte são objetos de estudo em diferentes trabalhos que visam entender o comportamento e a evolução do software (ANTONIOL *et al.*, 2000; KUHN; DUCASSE; GîRBA, 2007; MALETIC; MARCUS, 2000).

2.2.1.2 *Bug Databases*

Conforme Serrano e Ciordia (2005), esse tipo de base de dados mantém informações sobre a criação e resolução de *bugs*, *patches* de atualização, dentre outras tarefas, atuando como um sistema de rastreamento de *bugs*. A descrição do *bug* encontrada no sistema pode ser reportada por desenvolvedores ou usuários na forma de um *bug report* (ou *issue*), que possui uma série de informações que poderão ser utilizadas pelos responsáveis em manter o funcionamento do sistema. Nesse contexto, Thomas, Hassan e Blostein (2014) afirmam que os principais dados não estruturados que possuem relevância em um *bug report* são:

- **Título**: uma mensagem resumindo o conteúdo do *bug report*, escrito pela pessoa que está reportando o *bug*;
- **Descrição**: uma mensagem descrevendo os detalhes sobre o *bug*;
- **Comentários**: mensagens deixadas pelos desenvolvedores ou outros usuários sobre o *bug*.

Exemplos de estudos que utilizam as informações contidas em *bug reports* para realizarem seus experimentos podem ser encontrados na literatura (ANVIK; MURPHY, 2007; BETTENBURG *et al.*, 2008; LINSTEAD; BALDI, 2009).

2.2.1.3 Listas de Discussões

Na compreensão de Shihab *et al.* (2010), listas de discussões são arquivos de textos que registram a comunicação entre desenvolvedores, gerentes e outros *stakeholders*. Tais listas, usualmente, compreendem um conjunto de mensagens de e-mail, que contém um cabeçalho (informando remetente, destinatário, data e hora), o corpo da mensagem (texto não estruturado) e um conjunto de anexos (documentos adicionais). Acerca deste tema, Rigby e Hassan (2007) afirmam que em comparação a *bug reports* e código-fonte, essas listas de discussões sãos menos estruturadas e mais ambíguas. Devido ao seu conteúdo, este estilo de comunicação torna-se uma fonte rica de informação para que pesquisadores possam entender e melhorar práticas e processos no desenvolvimento de software.

2.2.1.4 Sistema de Controle de Versão

O sistema de controle de versão mantém e registra o histórico das mudanças em um repositório de documentos. No âmbito do desenvolvimento de software, este tipo de sistema é utilizado para manter e controlar edições em código-fonte ou atuar como um *back-up* remoto (CARLSSON, 2013). Na compreensão de Thomas, Hassan e Blostein (2014), grande parte das ferramentas de controle de versão (SVN[2], CVS[3], Git[4]) permite os desenvolvedores fornecer uma mensagem descrevendo as mudanças realizadas no código do sistema. Tais mensagens não estruturadas podem agregar valor aos pesquisadores, pois fornecem um meio de compreender como o código está evoluindo com o passar do tempo. Com base nisso, diversos estudos que utilizam as informações contidas em sistemas de controle de versão podem ser encontrados na literatura (BIRD *et al.*, 2009; CANFORA; CERULO; PENTA, 2007; CARLSSON, 2013; KIM; ERNST, 2007; ZIMMERMANN *et al.*, 2005).

2.2.1.5 Documento de Requisitos

Um documento de requisitos descreve as características e o comportamento de um sistema. Conforme Thomas, Hassan e Blostein (2014), requisitos podem ser categorizados em funcionais, que especificam o comportamento do sistema, ou de qualidade, que especificam os atributos de qualidade do sistema. Este tipo de documento, usualmente, é escrito na forma

[2] https://subversion.apache.org/
[3] https://www.nongnu.org/cvs/
[4] https://git-scm.com/

de texto em linguagem natural e pode ser uma fonte de mineração para extrair informações relevantes e prover uma visão mais estruturada do sistema (SAMPAIO; RASHID, 2008).

2.2.1.6 Repositórios de Software

Na concepção de Thomas, Hassan e Blostein (2014), repositório de software é uma coleção de sistemas *open source* que contém milhares de sistemas cujo o código-fonte pode ser facilmente consultado por terceiros. Torna-se necessário, ainda, salientar que no entendimento de Kagdi, Collard e Maletic (2007), repositório de software refere-se aos artefatos produzidos e arquivados durante a evolução do software. Portanto, esses repositórios podem incluir as informações dos repositórios discutidos previamente. Alguns dos repositórios mais populares são: *SourceForge*[5], *Bitbucket*[6] e o *GitHub*[7].

2.2.2 Técnicas para Mineração de Dados Não Estruturados

Segundo Thomas, Hassan e Blostein (2014), no campo da mineração de repositórios não estruturados, um processo recorrente aplicado pelos pesquisadores dessa área é guidado em conformidade com a seguinte metodologia: (i) o dado é pré-processado aplicando uma ou mais técnicas de processamento de linguagem natural (*Natural Language Processing* - NLP); (ii) é aplicada uma técnica de mineração ao dado pré-processado, permitindo a execução de alguma tarefa no contexto da Engenharia de Software. O processo discutido é ilustrado conforme a Figura 8.

Nesta subseção serão descritas com mais detalhes as técnicas de pré-processamento de dados e a abordagem para extração da informação.

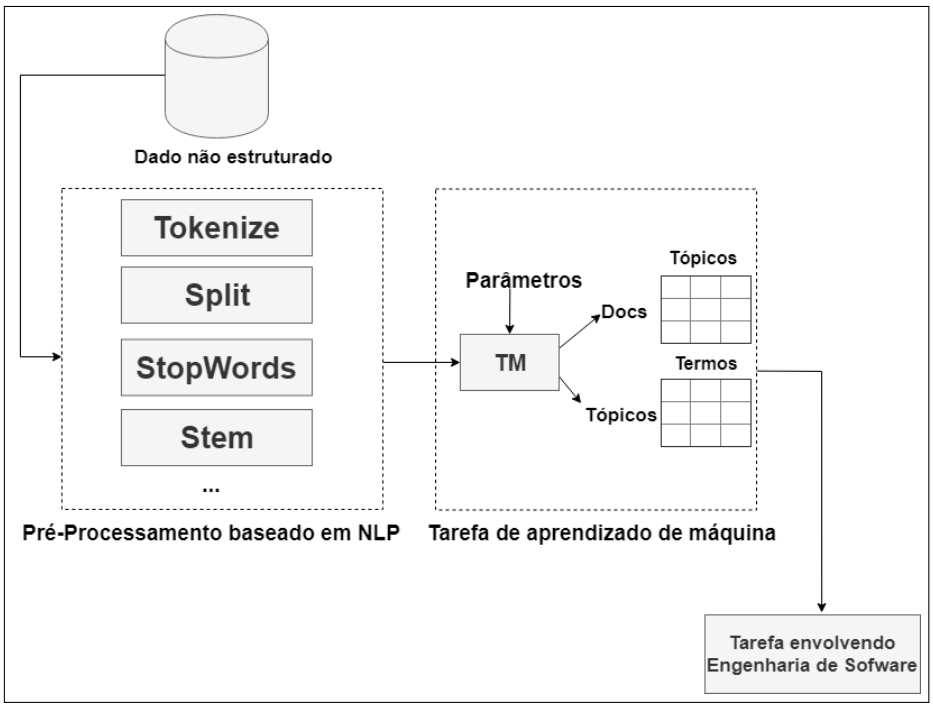
2.2.2.1 Pré-processamento de Dados com NLP

Na concepção de Sun *et al.* (2014), o pré-processamento de dados não estruturados desempenha um papel importante no processo de análise. Dessa forma, os ruídos que são causados pela natureza desse tipo de dado podem ser tratados de modo a otimizar o processo de análise das técnicas de mineração.

A Figura 9 ilustra alguns passos que podem ser aplicados na fase de pré-processamento dos dados, conforme descrito a seguir:

[5] https://sourceforge.net/
[6] https://bitbucket.org/
[7] https://github.com/

Figura 8 – Processo de mineração de repositório de software não estruturado



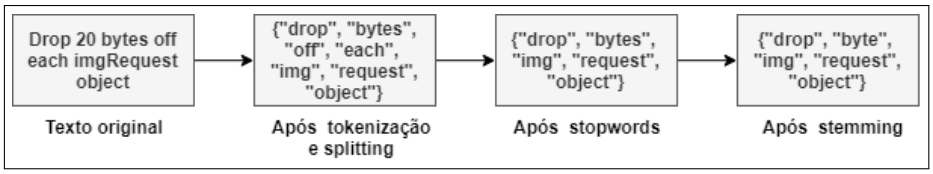
Fonte**:** Adaptado de Thomas, Hassan e Blostein (2014)

- ***Tokenização*:** o texto original é transformado em *tokens*. Durante esse passo, pontuações e números são removidos. Com base no exemplo ilustrado na Figura 9, após a *tokenização*, o número 20 é removido.
- ***Splitting:*** identificadores (nomes de classes, métodos e variáveis) são divididos em múltiplas partes baseado em convenções, por exemplo, *camel case* (nomeClasse) e *underscores* (nome_classe). De acordo com o exemplo ilustrado, a palavra *"imgRequest"* é divida em *"img"* e *"request"*. Esse tipo de abordagem é bastante utilizada na mineração de código-fonte de modo a extrair informação relevante com base nos identificadores (LAWRIE *et al.*, 2006; LAWRIE *et al.*, 2007; MADANI *et al.*, 2010).
- **Remoção de Palavra (*Stopword*)**: remoção de palavras que não possuem relevância a análise. Uma lista de *stopwords* com base na língua inglesa poderia incluir, por exemplo, "*the*", "*on*", "*at*", "*in*", dentre outros. Vale salientar que essa lista é customizável, e assim, jargões específicos de domínio também podem ser incluídos. Conforme o exemplo, as palavras *"off"* e *"each"* foram removidas.
- ***Stemming***: esse processo consiste em reduzir a palavra ao seu radical, isto é, palavras como "*fishing*", "*fished*"e "*fisher*"seriam reduzidas a "*fish*". Com base no exemplo, a palavra *"bytes"* é reduzida a *"byte"*. Vale salientar, ainda, que

outra abordagem utilizada é a *lemmatization*, ou seja, o processo de descobrir a forma normalizada de uma palavra.

- ***Pruning*:** a remoção das palavras é feita com base na sua ocorrência. Por exemplo, palavras que ocorrem acima de 80% e abaixo de 2%.

Figura 9 – Exemplo de pré-processamento em NLP



Fonte**:** Adaptado de Thomas, Hassan e Blostein (2014)

2.2.2.2 Topic Modeling

Na perspectiva de Corley, Damevski e Kraft (2018), *topic modeling* é uma técnica que extrai um conjunto de tópicos de um *corpus* de documentos. Os autores ressaltam que um *tópico* representa a coocorrência de um conjunto de palavras que idealmente corresponde a um conceito entendível ao ser humano. Landauer *et al.* (2013), em seu trabalho, afirmam que essa abordagem fornece uma visão alto nível e interpretável de um *corpus* de documentos. Além disso, devido à presença de dados textuais nos repositórios de software, Thomas (2011) afirma que *topic modeling* fornece meios para automaticamente indexar, buscar, agrupar e estruturar documentos não estruturados.

Latent Dirichlet Allocation (LDA), de acordo com Blei, Ng e Jordan (2003), é um modelo probabilístico utilizado em coleções de dados discretos tais como texto. Corley, Damevski e Kraft (2018) afirmam que o LDA é um modelo probabilístico baseado em tópicos que extrai uma representação interpretável a partir de um amplo conjunto de dados. Torna-se necessário, ainda, ressaltar que o LDA é amplamente utilizado em pesquisas no âmbito da Engenharia de Software para modelagem de tópicos em repositórios de software (CHEN; THOMAS; HASSAN, 2016). Em resumo, LDA gera duas matrizes: matriz *topic-word* e matriz *document-word*. Na primeira, são exibidas as palavras mais prováveis no *tópico*, já a última, mostra os tópicos mais prováveis em cada documento.

Em LDA, um *tópico* é uma coleção de palavras que frequentemente ocorrem juntas. Dado um conjunto de n documentos $d_1, \cdots, d_n$, LDA descobre automaticamente um conjunto de Z *tópicos*, Z = $\{z_1, \cdots, z_k\}$, bem como o mapeamento θ entre os *tópicos* e documentos. O

número de *tópicos*, K, é uma entrada que controla a granularidade dos *tópicos*. A notação θ_{ij} descreve o valor associado ao *tópico* z_i no documento d_j.

Figura 10 – Topic Modeling

	Top words
z1	thread, sleep, notify, interrupt
z2	network, bandwith, timeout
z3	view, html, javascript, css

(a) Tópicos (Z)

	z1	*z2*	*z3*
d1	0.2	0.8	0.0
d2	0.0	0.8	0.2
d3	0.6	0.0	0.4
d4	1.0	0.0	0.0

(b) mapeamento (θ) Tópicos/Documentos

Fonte**:** Adaptado de Li *et al.* (2018)

A Figura 10 ilustra um exemplo da execução da técnica *topic modeling*, onde três *tópicos* são descobertos a partir de quatro documentos. Na Figura 10a, os três *tópicos* (z_1, z_2, z_3) são definidos de acordo com as palavras com maior distribuição de probabilidade sobre todas as palavras contidas no *corpus*, enquanto que na Figura 10b, os quatro documentos (d_1, d_2, d_3, d_4) são representados pelo vetor que informa a participação do *tópico* (z_i) em cada documento (d_j).

De acordo com Thomas, Hassan e Blostein (2014), LDA tem sido utilizado para analisar tópicos e tendências presentes no *Stack Overflow*, permitindo aos pesquisadores quantificar a popularidade de certos tópicos e tecnologias que estão mudando com o passar do tempo. Desta forma, é possível extrair evidências que podem direcionar pesquisadores em seus trabalhos ou interessados em, por exemplo, melhorar alguma ferramenta ou ambiente de desenvolvimento de software.

2.3 STACK EXCHANGE

Conforme Barua, Thomas e Hassan (2014), a Engenharia de Software é um campo composto por um conjunto diverso de tecnologias, ferramentas, linguagens e plataformas. Deste modo, é esperado do desenvolvedor uma proficiência em vários paradigmas. Contudo, mesmo para desenvolvedores experientes, manter-se atualizado diante da rápida evolução destes paradigmas é uma tarefa complexa.

Diante dessa problemática, Shirani *et al.* (2019) afirmam que comunidades de domínio específico baseadas em perguntas e respostas estão se tornando populares entre os profissionais que buscam por informações de modo eficiente. Por estas razões, segundo Barua, Thomas e Hassan (2014), os desenvolvedores frequentemente recorrem a sites baseados em

perguntas e repostas (*Questions and Answers - Q&A*) para buscarem assistência em relação aos desafios técnicos que estes enfrentam. Além disso, Treude, Barzilay e Storey (2011) afirmam que o corpo de conhecimento gerado por esses sites no âmbito do desenvolvimento de software tem complementado, ou mesmo substituído, a documentação do produto quando esta é insuficiente ou inexistente.

Segundo Begel, Bosch e Storey (2013), o *Stack Exchange*[8] é uma rede de plataformas no formato pergunta e resposta contemplando tópicos desde programação à filosofia. Conforme Posnett *et al.* (2012), a maior plataforma sobre programação do *Stack Exchange* é o *Stack Overflow*. Ponzanelli, Bacchelli e Lanza (2013) afirmam que o *Stack Overflow* é um importante veículo para compartilhar conhecimento no âmbito do desenvolvimento de software.

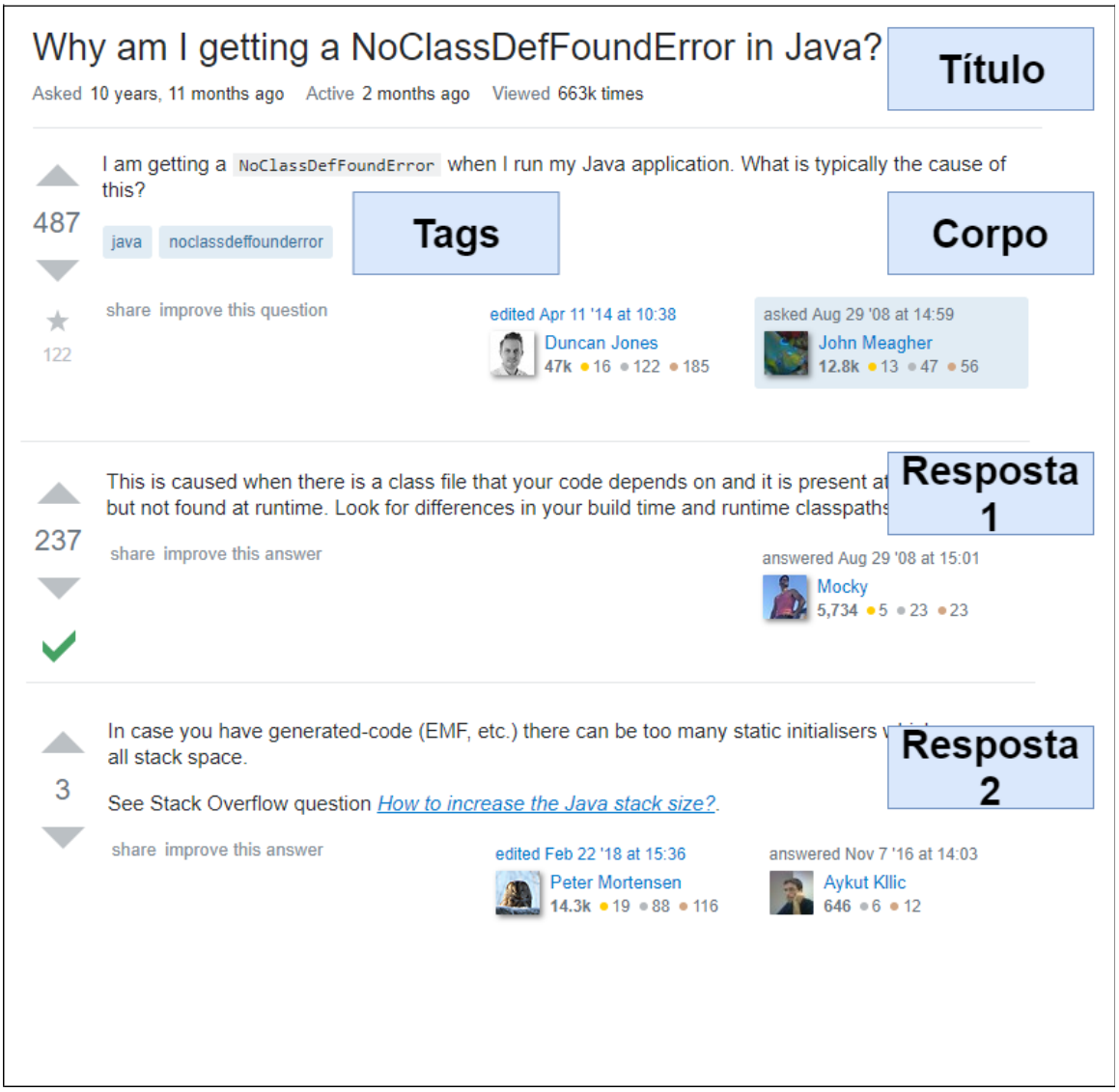
De acordo com os dados fornecidos por Shirani *et al.* (2019), cerca de 50 milhões de desenvolvedores visitam a plataforma mensalmente e mais de 85% dos usuários visitam o *Stack Overflow* diariamente. Além disso, Mamykina *et al.* (2011) mostram que a maioria das questões são respondidas em média 11 minutos após a sua postagem. A Figura 11 ilustra um exemplo de um *post*, feito na plataforma *Stack Overflow*, composto por: título da pergunta, corpo da questão, *tags* (para agrupar discussões similares) e, por fim, as respostas fornecidas pelos usuários.

Outra plataforma que faz parte da família *Stack Exchange* é o *Software Engineering Stack Exchange*. Nele são discutidos diversos assuntos voltados exclusivamente para o domínio da Engenharia de Software (ENGINEERING, 2019). Além disso, toda a estrutura básica dos *posts* presente no *Stack Overflow* e ilustrada na Figura 11 é replicada no *Software Engineering Stack Exchange*.

Nesse contexto, Treude, Barzilay e Storey (2011) ressaltam que ambientes baseados em perguntas e respostas são particularmente efetivos para responder questões conceituais, questões que envolvem revisão de código e questões recém postadas.

[8] https://stackexchange.com

Figura 11 – Exemplo de discussão no Stack Overflow



Fonte: Adaptado de Shirani et al. (2019)

3 TRABALHOS RELACIONADOS

Neste capítulo são apresentados os trabalhos encontrados durante o andamento desta pesquisa e que estão, de algum modo, relacionados com o objeto de estudo do presente trabalho.

3.1 MINERAÇÃO DO REPOSITÓRIO DE FÓRUNS SOBRE ENGENHARIA DE SOFTWARE

Kahani *et al.* (2016) buscam entender quais questões primárias norteiam os desenvolvedores no uso das ferramentas MDE e quais destas questões são comumente enfrentadas por desenvolvedores novatos na comunidade MDE. A partir disso, os pesquisadores conduzem uma abordagem qualitativa e quantitativa no fórum de discussão do Eclipse (dados de 2002 a 2015) realizando uma análise manual e recorrendo à técnica de *topic modeling* com LDA. Os autores concluem que as perguntas mais frequentes feitas por usuários novatos diz respeito a configuração e interoperabilidade. Além disso, em relação as transformações entre modelos, a maioria dos usuários novatos (58%) requisitam ajuda a nível de código, enquanto uma outra parte (33%) demandam dúvidas conceituais. Outros problemas citados pelos os autores são: problemas envolvendo *plugins* e falta de documentação.

Zou *et al.* (2015) apresentam um estudo empírico visando compreender as necessidades dos desenvolvedores que utilizam o *Stack Overflow* acerca do tema requisito de qualidade. Para alcançar o objetivo proposto no trabalho, os autores extraíram tópicos do *corpus* (dados de jul/2008 a set/2014) recorrendo a técnica de *topic modeling* utilizando o *Latent Dirichlet Allocation* (LDA). Além disso, os tópicos foram categorizados a partir da taxonomia fornecida pela ISO9126. Por fim, após a análise dos dados, os autores concluíram que os desenvolvedores possuem um interesse nos atributos confiabilidade e usabilidade. Por outro lado, os atributos relacionados a manutenibilidade e eficiência possuem menos interesse por parte da comunidade. Além disso, os autores perceberam que, com o passar do tempo, houve uma crescente busca acerca dos atributos funcionalidade e confiabilidade.

Bajaj, Pattabiraman e Mesbah (2014) realizaram uma análise empírica sobre as discussões relacionadas ao desenvolvimento web no *Stack Overflow* visando entender as principais dificuldades, equívocos e erros conceituais entre os desenvolvedores. Para alcançar esse objetivo, os autores conduzem uma análise qualitativa e quantitativa em mais de 500.000 discussões (dados de jun/2008 a mar/2013) acerca do tema que norteia o trabalho. De modo a extrair os tópicos mais relevantes das discussões, os autores utilizam técnica de *topic modeling* através

do LDA. Após analisar os dados, os autores obtiveram as seguintes conclusões: (i) discussões relacionadas a *cross-browser*, predominante no passado, estão se tornando menos importante; (ii) questões envolvendo *Document Object Model* (DOM) APIs e tratamento de eventos são conceitos ainda não tão claros aos desenvolvedores; (iii) HTML5 tem se tornado popular em aplicações web e *mobile*; (iv) desenvolvedores experientes estão enfrentando dificuldades na utilização de novas *features* adicionadas ao JavaScript e HTML5.

Villanes *et al.* (2017) apresentam um estudo empírico cujo o objetivo principal é identificar as práticas relacionadas a testes em aplicações móveis, mais especificamente, na plataforma Android. Os autores analisaram cerca de 18.000 discussões (dados de set/2008 a mar/2017) coletadas a partir do repositório do *Stack Overflow*. A técnica *topic modeling* implementada pelo LDA foi utilizada para categorizar os dados. Os autores concluíram que as principais ferramentas que os desenvolvedores utilizam são: Appium, Espresso, Monkey e Robotium. Além disso, os autores discorrem que os principais conceitos relacionados a teste de aplicações móveis são: ferramentas de teste, teste de ambiente, UI teste, teste funcional e teste de unidade.

Kochhar (2016) apresenta um estudo sobre quais são os desafios e os principais assuntos das discussões acerca de teste de software. De modo a alcançar este objetivo, o autor realiza uma mineração no repositório do *Stack Overflow* coletando mais de 38.000 questões (de jan/2009 a dez/2014). Além disso, a técnica de *topic modeling*, apoiada pelo LDA, foi utilizada para categorizar os assuntos mais relevantes sobre o tema estudado. O autor, em seu trabalho, obtém as seguintes conclusões: (i) os tópicos predominantes nas discussões são: *framework* para realização de testes, testes em banco de dados e questões relacionadas a cliente/servidor; (ii) com o passar do tempo, questões ligadas a teste em aplicações móveis têm despertado interesse na comunidade de desenvolvedores/analista de teste; (iii) outros conceitos como melhores práticas em teste, também são discutidos.

Rahman (2017) realizou uma mineração do repositório do *Stack Overflow* para entender padrões e efeitos desconhecidos de ataque distribuído de negação de serviço (*Distributed Denial of Service* - DDoS) a partir de discussões acerca deste tema. Para extrair os principais tópicos das discussões, o autor recorre a técnica *topic modeling* implementada pelo o LDA. Os dados extraídos a partir da análise foram utilizados como insumo pelo autor para propor um *framework* que visa proteger a estrutura da nuvem AWS contra ataques do tipo DDoS.

Barua, Thomas e Hassan (2014) propuseram uma metodologia para descobrir e quantificar tópicos e tendências tendo como objetivo de entender as necessidades dos desenvol-

vedores que utilizam a plataforma do *Stack Overflow*. De modo a atingir este objetivo, os autores utilizam o LDA para analisar mais de 3.474.987 discussões, dentre as quais, 973.267 (28%) são questões e 2.501.720 (72%) são respostas, coletadas a partir de jul/2008 a set/2010. Os resultados mostram que questões em alguns tópicos levam a discussões em outros tópicos. Além disso, os tópicos que estão ganhando mais notoriedade entre os desenvolvedores são: desenvolvimento web (especialmente com jQuery), aplicações móveis (plataforma Android), Git e MySQL.

A Tabela 1 agrupa os trabalhos relacionados, citados previamente, de forma sucinta destacando os seguintes pontos:

- **Trabalho:** elenca os autores do trabalho;
- **Objetivo:** qual o objetivo principal da pesquisa;
- **Técnica Aplicada**: qual a técnica foi aplicada para auxiliar na análise dos dados. A técnica pode incluir abordagem por tópicos, fornecida pelo LDA, e/ou uma abordagem manual;
- **Repositório**: qual a plataforma alvo utilizada para realizar a mineração;
- **Total Discussões:** o *corpus* que engloba a quantidade de discussões analisadas no trabalho.

A maioria dessas pesquisas combinaram o uso da abordagem por tópicos, implementada pelo LDA, e uma abordagem manual. Nesta última, os autores selecionavam uma parte da amostra das discussões e realizavam uma análise mais profunda objetivando responder questões de pesquisas de acordo com contexto dos estudos realizados.

3.2 SURVEY COM OS DESENVOLVEDORES

Agner *et al.* (2013) realizaram um estudo objetivando compreender como a UML e práticas relacionadas às abordagens dirigidas por modelo são aplicadas na indústria, mais especificamente a projetos de software no domínio de sistemas embarcados no Brasil. Os pesquisadores compilaram os resultados de um *survey* envolvendo 209 desenvolvedores brasileiros que atuam no domínio supracitado. A partir do estudo, os autores concluíram que UML é a linguagem mais utilizada para realizar a modelagem do sistema. Além disso, os resultados mostraram que cerca de 23% dos participantes conhecem e usam, parcialmente ou não, as abordagens dirigidas por modelo.

Vetro, Bohm e Torchiano (2015) buscaram obter um melhor entendimento sobre os benefícios e barreiras na adoção das técnicas dirigidas a modelo no desenvolvimento de sistemas

Tabela 1 – Síntese dos trabalhos relacionados

Trabalho	Objetivo	Técnica Aplicada	Repositório	Total Discussões
Kahani *et al.* (2016)	Entender quais questões primárias norteiam os desenvolvedores no uso das ferramentas MDE e quais destas questões são comumente enfrentadas por desenvolvedores novatos na comunidade MDE	LDA/Manual	Fórum Eclipse	188.915
Zou *et al.* (2015)	Compreender as necessidades dos desenvolvedores acerca do tema requisito não funcional	LDA/Manual	*Stack Overflow*	21.000.000
Bajaj, Pattabiraman e Mesbah (2014)	Entender as principais dificuldades, equívocos e erros conceituais entre os desenvolvedores no domínio de desenvolvimento web	LDA/Manual	*Stack Overflow*	500.000
Villanes *et al.* (2017)	Identificar as práticas relacionadas a testes em aplicações móveis, mais especificamente, na plataforma Android	LDA	*Stack Overflow*	18.000
Kochhar (2016)	Desafios e os principais assuntos das discussões acerca de teste de software	LDA/Manual	*Stack Overflow*	38.000
Rahman (2017)	Entender padrões e efeitos desconhecidos de ataque distribuído de negação de serviço	LDA/Manual	*Stack Overflow*	Não informado
Barua, Thomas e Hassan (2014)	Descobrir e quantificar tópicos e tendências tendo como objetivo de entender as necessidades dos desenvolvedores	LDA/Manual	*Stack Overflow*	3.474.987

Fonte**:** Elaborado pelo autor

embarcados. Os pesquisadores conduziram o estudo em empresas situadas na Alemanha, a partir disso, os autores disponibilizaram um *survey* e confrontaram os dados, de modo a responder as questões de pesquisas que guiaram o trabalho. Os autores concluem que há um ganho na qualidade do software, por outro lado, existe um esforço requerido, bem como a falta de competência e suporte ferramental.

Tomassetti *et al.* (2012) investigaram o nível de maturidade no uso de MDE na indústria Italiana. Os autores analisaram os resultados do *survey* disponibilizado na forma de um questionário *online* para 155 desenvolvedores italianos. Os autores concluem que 68% dos entrevistados utilizam técnicas de MDE, entre eles, 44% utilizam geração de código a partir dos modelos, enquanto que, 10% aplicam transformação entre modelos.

Hutchinson *et al.* (2011) apresentam os resultados de um estudo empírico avaliando a MDE na indústria, visando entender as práticas que levam ao sucesso e a falha na adoção

de MDE no desenvolvimento de software. Dentre as abordagens utilizadas no estudo, os pesquisadores aplicaram um *survey*, disponibilizado *online*, para 250 praticantes de MDE. A partir da compilação das respostas, os autores identificaram que o uso de MDE melhora a produtividade (principalmente em geração de código) e manutenibilidade. Por outro lado, os entrevistados acham que o uso de MDE implica em uma carga de treinamento nas ferramentas MDE.

Forward e Lethbridge (2008) realizaram um estudo objetivando entender aspectos relacionados a MDE. Para isso, os autores aplicaram um *survey* resultando na participação de 113 desenvolvedores. Os resultados mostraram que poucos desenvolvedores (23%) utilizam linguagens de modelagem para projetar ou entender o sistema a ser desenvolvido, enquanto que, em relação a geração de código, apenas 18% geravam parcialmente código a partir de modelos.

3.3 DISCUSSÃO

Os trabalhos abordados e resumidos na Tabela 1 são estudos que utilizam a técnica de mineração de repositório para entender como os desenvolvedores lidam com aspectos concernentes ao processo de desenvolvimento de software. Nesse contexto, a mineração, nos trabalhos citados anteriormente, é realizada em ambientes de discussões no formato pergunta e resposta, os quais fornecem um meio propício para extrair conhecimento em relação às práticas de Engenharia Software.

Vale ressaltar que a maioria dos estudos realizaram a mineração de repositório na base do *Stack Overflow*, considerado o principal ambiente de discussão técnica entre os desenvolvedores (LINARES-VáSQUEZ; DIT; POSHYVANYK, 2013). Por fim, nos trabalhos que realizaram a análise das discussões do *Stack Overflow*, o tópico MDE não foi abordado.

Além disso, enquanto os trabalhos apresentados focam em apenas um repositório, a presente pesquisa analisa os dados gerados a partir de duas plataformas que discutem aspectos teóricos e técnicos no contexto da Engenharia Dirigida por Modelos.

Por fim, os estudos discutidos e apresentados na Seção 3.2, agregam valor a presente pesquisa por abordarem os aspectos que envolvem as vantagens e desvantagens na adoção das práticas MDE na indústria de software.

4 METODOLOGIA

Neste capítulo é descrita a metodologia desta dissertação, que consistiu em realizar uma mineração em duas plataformas de discussões relacionadas a Engenharia de Software, são eles: *Stack Overflow* e *Software Engineering Stack Exchange*. No contexto desta pesquisa, os assuntos de interesse permeiam sobre o uso das tecnologias e discussões de conceitos no âmbito da MDE.

Além disso, as seguintes questões de pesquisa norteiam esse trabalho de mestrado e guiam a metodologia:

QP1: Quais temas relacionados a MDE os desenvolvedores têm discutido nas plataformas de perguntas e respostas em Engenharia de Software?

O propósito desta questão é identificar quais são os temas da abordagem MDE que estão sendo discutidos pelos desenvolvedores em plataformas de perguntas e respostas em Engenharia de Software. Desta forma, é possível elencar às diversas perspectivas das discussões debatidas pelos desenvolvedores, isto é, os debates envolvendo MDE são discussões apenas técnicas e/ou teóricas?.

QP2: Quais são os domínios que MDE está sendo empregada?

O propósito desta questão é identificar, a partir das discussões entre os desenvolvedores, quais são os domínios que a MDE está sendo utilizada.

QP3: Quais são as ferramentas MDE discutidas pelos desenvolvedores?

Objetivo principal desta questão é elencar quais são as principais ferramentas MDE que estão gerando discussões entre os desenvolvedores. Esta questão visa elencar as ferramentas citadas pelos desenvolvedores quando estes debatem assuntos relacionados a MDE.

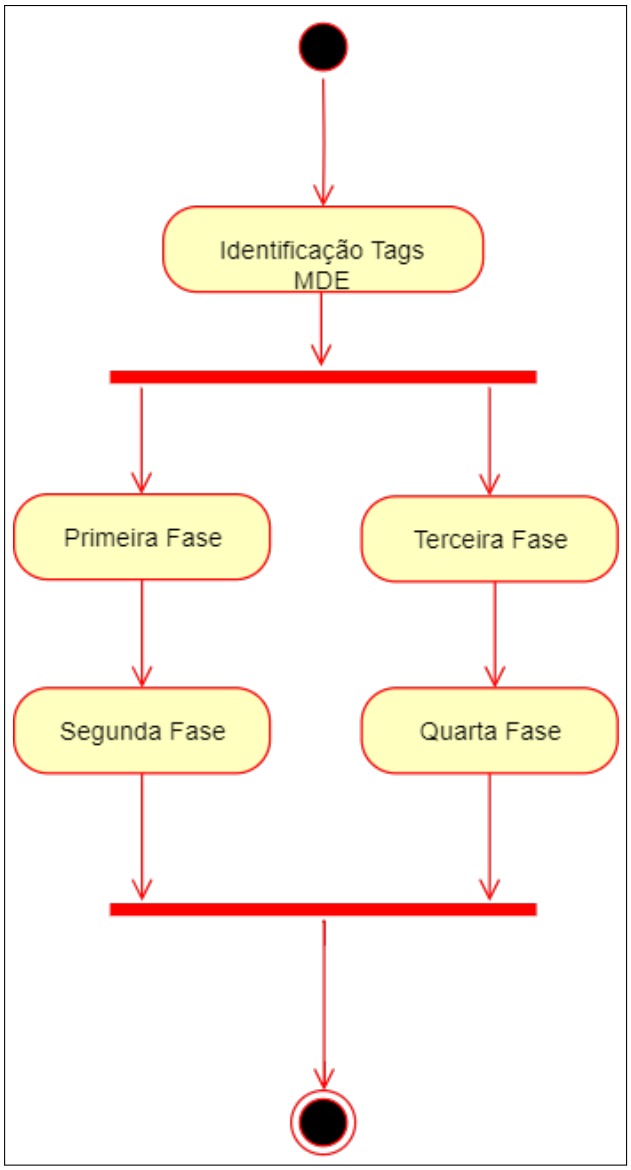
QP4: Quais temas sobre MDE atraíram mais a atenção dos desenvolvedores?

O propósito desta questão é elencar quais temáticas da referida abordagem têm atraído a participação da comunidade. Deste modo, identificar quais assuntos atraem à atenção (visualização das discussões) e quais motivam à interação (geração de respostas nas discussões) entre os desenvolvedores.

A metodologia utilizada nesta pesquisa pode ser dividida em quatro fases: (i) mineração dos repositórios do *Stack Overflow* e *Software Engineering Stack Exchange*; (ii) classificação

das discussões; (iii) identificação das principais ferramentas MDE; (iv) por fim, discussões sobre ferramentas MDE. A Figura 12 ilustra o processo aplicado na metodologia utilizada. Além disso, vale ressaltar, que as fases elencadas não são necessariamente executadas de forma sequencial. A seguir, cada fase será melhor descrita.

Figura 12 – Processo da Metodologia Utilizada



Fonte**:** Elaborado pelo autor

4.1 FASE 1: MINERAÇÃO DOS REPOSITÓRIOS DO STACK OVERFLOW E SOFTWARE ENGINEERING STACK EXCHANGE

Nesta fase, a mineração foi realizada nas plataformas *Stack Overflow* e *Software Engineering Stack Exchange*, mais especificamente pelo *dataset* provido pelo *Stack Exchange Data*

Explorer[1]. O SO e o SE atuam como um repositório de conhecimento no âmbito da Engenharia de Software e, desta forma, é possível extrair, a partir dos *posts* nestas plataformas, quais são as principais discussões sobre determinadas tecnologias ou práticas de desenvolvimento, elencar os problemas mais recorrentes que os desenvolvedores estão enfrentando em um determinado contexto, dentre outros assuntos.

As discussões foram selecionadas a partir de *tags* utilizadas pelos usuários do SO e do SE para classificar uma pergunta. Vale salientar que, no contexto das plataformas analisadas, *tag* é uma palavra ou frase que descreve um tópico de uma questão. Desta forma, esse recurso é utilizado pelo usuário para conectar especialistas que poderão responder uma determinada questão inserida em um domínio específico. Nesse contexto, um usuário que deseje sanar alguma dúvida relacionada a testes unitários, por exemplo, poderia adicionar a *tag "unit-testing"*, previamente cadastrada na plataforma, à sua pergunta.

Objetivando reunir discussões relacionadas ao interesse desta pesquisa, a(s) *tag*(s) utilizada(s) para categorizar uma discussão devem abranger os conceitos de MDE. Para alcançar este objetivo, a seguinte estratégia foi adotada: as *tags* que classificavam uma questão deveriam se enquadrar nos termos-chaves que se encontram na literatura e que caracterizam a abordagem MDE, ou seja, as *tags* deveriam fazer referência aos termos MDE, MDD ou MDA.

A partir disso, uma busca das *tags* cadastradas nas plataformas foi realizada, o que resultou na identificação das seguintes *tags*: *"mde"*, *"mdsd"*, *"mdd"*, *"model-driven"*, *"model-driven-development"* e *"mda"*. As *tags* elencadas foram classificadas no conjunto das *tags* genéricas conforme ilustra a Tabela 2, ou seja, este conjunto possui relação com conceitos de MDE, MDD ou MDA.

Com base nas *tags* elencadas, as questões que caracterizavam discussões sobre MDE foram identificadas. Logo, se uma discussão estivesse marcada com alguma das *tags* citadas anteriormente, tal discussão faria parte do escopo de discussões a serem analisadas de acordo com as perguntas de pesquisa deste trabalho. Vale salientar que uma discussão é uma troca de ideias entre duas ou mais pessoas acerca de um assunto, ou seja, no contexto desta pesquisa, questões não respondidas ou questões respondidas pelo próprio autor são descartadas por não agregarem valor à análise. Portando, as discussões no *Stack Overflow* e no *Software Engineering Stack Exchange* que foram consideradas como pertinentes provêm da combinação entre pergunta e uma (ou mais) resposta(s), onde há pelo menos uma resposta cujo autor não é o autor da pergunta.

[1] https://data.stackexchange.com/

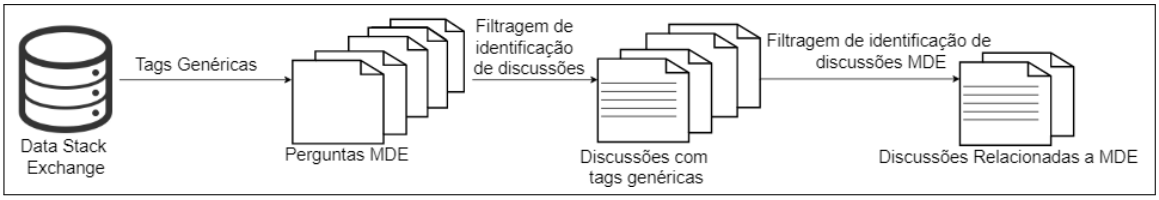
Tabela 2 – Tags Genéricas

	MDE Tags
Tags Genéricas	"mde", "mdsd", "mdd", "model-driven", "model-driven-development", "mda"

Fonte**:** Elaborado pelo autor

Para categorizar as discussões do *Stack Overflow* e do *Software Engineering Stack Exchange*, foram seguidas etapas visando sistematizar o processo de análise dos dados. O processo é exposto na Figura 13 e cada uma de suas etapas será descrita a seguir.

Figura 13 – Primeira Fase da Metodologia Utilizada



Fonte**:** Elaborado pelo autor

4.1.1 Etapa 1: Identificação das Discussões Relevantes

Nesta etapa, para a extração dos dados, consultas SQL foram realizadas na base de dados fornecida pelo *Stack Exchange* utilizando o conjunto de *tags* conforme descrito na Tabela 2. A partir disso, um *corpus* inicial contendo as discussões que incluíam as *tags* genéricas foi retornado.

O resultado gerado das consultas é armazenado em um arquivo na forma de *Comma Separated Values* (CSV), onde cada instância apresenta os atributos descritos na Tabela 3. É necessário ressaltar que foi aplicada uma filtragem para verificar quais eram as publicações que caracterizavam discussões no contexto desta pesquisa. Deste modo, foram filtradas as discussões que apresentavam uma ou mais respostas onde, caso houvesse apenas uma resposta, esta não havia sido feita pelo o autor da pergunta.

4.1.2 Etapa 2: Obtenção das Discussões Relacionadas à MDE

Uma vez que as discussões que utilizaram as *tags* genéricas foram identificadas, a segunda etapa visa verificar, dentro dessas discussões, quais são aquelas que de fato estão relacionadas à temática deste estudo. Esse processo se faz necessário pois o usuário pode utilizar

Tabela 3 – Atributos da Base de Dados

Atributo	Descrição
Id	Identificador único na base de dados
Title	Título da publicação
Body	Corpo da publicação
Tags	*Tag* da publicação
CreationDate	Data de criação
AnswerCount	Número de respostas
ViewCount	Número de visualizações
CommentCount	Número de comentários
FavoriteCount	Número de favoritos
Score	Métrica de popularidade

Fonte**:** Elaborado pelo autor

uma ou mais *tag*s para classificar sua questão no domínio de MDE, porém o assunto central da discussão pode não estar direcionado a MDE. Além disso, a *tag* utilizada pode se referir a outro domínio. O exemplo citado é ilustrado na Figura 14. Nessa situação, o usuário utilizou a *tag "mdd"*, porém o conceito dela está relacionado a *Microprocessor Driver Definition* e não a *Model-driven Development*. Ainda nessa problemática, a Figura 15 ilustra outro caso onde a *tag "mda"* não está relacionada aos conceitos da *Model-driven Architecture*, e sim ligada a *Managed Debugging Assistant*.

Na primeira fase foram obtidas 199 questões com suas respectivas respostas nas quais o usuário utilizou as *tags* genéricas. Nesse cenário, foi aplicada uma filtragem automática para verificar quais eram as publicações que caracterizavam discussões no contexto desta pesquisa. O *corpus* para análise totalizou 187 discussões válidas obtidas a partir da base do *Stack Overflow*. Por fim, a base de dados do *Software Engineering Stack Exchange*, replicando o mesmo processo aplicado ao SO, retornou um total de 11 discussões válidas para análise.

A partir das 187 discussões válidas extraídas do *Stack Overflow*, foram verificadas quais discussões estavam relacionadas à temática deste estudo. Para isso, foram descartadas as discussões nas quais os usuários utilizaram *tags* que não possuíam ligação com MDE. Além disso, foram analisados os conteúdos de cada discussão de modo a validar se o assunto discutido entre os desenvolvedores estava associado a MDE.

Após a seleção das discussões relacionadas a MDE, o subconjunto de discussões do *Stack Overflow*, que estão incluídas no escopo dessa pesquisa, totalizou em 74. Diante da mesma situação, a partir da plataforma *Software Engineering Stack Exchange*, foram extraídas apenas 9 discussões relevantes.

Figura 14 – Exemplo 1 de discussão não relacionada

Fonte**:** https://stackoverflow.com/questions/6756163

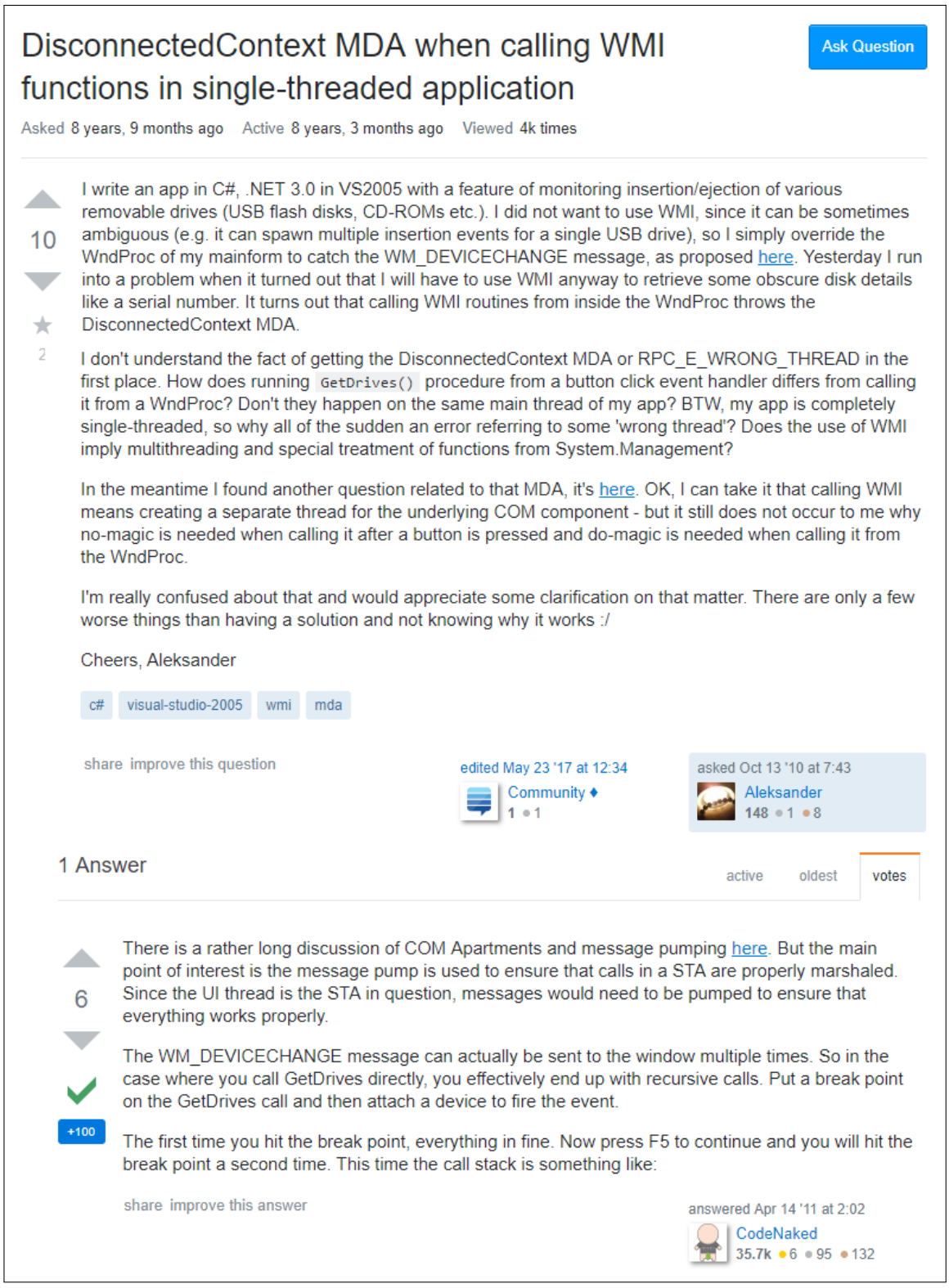
4.2 FASE 2: CLASSIFICAÇÃO DAS DISCUSSÕES

A segunda fase é responsável por desenvolver uma classificação para os subconjuntos de discussões relacionadas à MDE. A partir disso, é possível obter um melhor entendimento sobre o que está sendo discutido entre os desenvolvedores nas plataformas analisadas.

Nesse cenário, com base no conteúdo das discussões, as seguintes categorias foram identificadas: discussões conceituais, discussões técnicas, discussões técnicas/conceituais e, por fim, discussões de utilidade. A descrição de cada categoria é fornecida a seguir:

- **Discussões conceituais:** a discussão permeia sobre dúvidas em relação aos conceitos de MDE. A Figura 17 ilustra um exemplo de discussão conceitual onde o usuário do *Stack Overflow* utilizou um conjunto de *tags*, dentre elas a *tag "model-driven-development"*, que faz parte do conjunto genérico, para alcançar outros usuários que pudessem responder ao seu questionamento, ou seja, *"Quais as diferenças entre as abordagens MDSD e MDD"*;

Figura 15 – Exemplo 2 de discussão não relacionada



Fonte**:** https://stackoverflow.com/questions/3921661

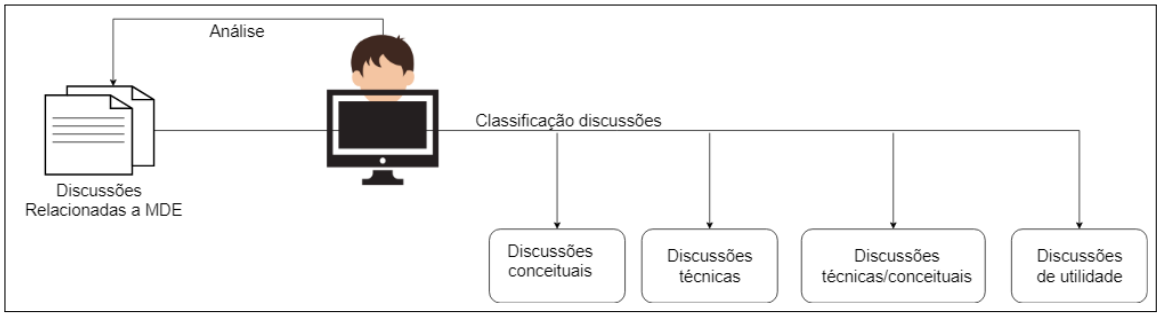
- **Discussões técnicas:** a discussão está relacionada a dúvidas específicas de uma ferramenta utilizada no domínio de MDE. A Figura 18 ilustra um exemplo desse tipo de discussão. Nesse contexto, o usuário do *Stack Overflow* está interessado

em como utilizar um recurso da ferramenta Acceleo responsável por realizar uma transformação M2T;

- **Discussões técnicas/conceituais**: discussões que abrangem questões tanto técnicas (ferramenta) e conceituais ao mesmo tempo. Esse tipo de situação é ilustrada na Figura 19. Nesse cenário, o desenvolvedor busca entender as diferenças conceituais entre duas ferramentas, bem como as suas particularidades;
- **Discussões de utilidade**: discussões que podem envolver assuntos sobre a viabilidade, benefícios, riscos exemplos de usos reais, dentre outros. Esse cenário é ilustrado na Figura 20. Nesse contexto, os usuários tendem a abordar nas suas discussões com outros desenvolvedores alguns dos pontos citados previamente.

A Figura 16 descreve o processo utilizado. Vale salientar, ainda, que essa classificação tem como objetivo traçar os anseios da comunidade sobre diferentes perspectivas de interesse sobre o tema em estudo.

Figura 16 – Segunda Fase da Metodologia Utilizada



Fonte**:** Elaborado pelo autor

4.3 FASE 3: IDENTIFICAÇÃO DAS PRINCIPAIS FERRAMENTAS MDE

O objetivo desta fase é realizar a identificação, a partir das discussões entre os desenvolvedores, das *tags* do *Stack Overflow* que denotam as ferramentas no âmbito da MDE. Vale ressaltar que a plataforma *Software Engineering Stack Exchange* não foi utilizada nesta fase, pois está fora do escopo da própria plataforma que possui o foco de agregar discussões no âmbito teórico (ENGINEERING, 2019). É necessário ressaltar que o autor desta pesquisa possui um conhecimento prévio de um conjunto de ferramentas para MDE devido à sua experiência. Logo, para diminuir o viés no processo de escolha das ferramentas selecionadas, a ferramenta identificada seria incluída no escopo deste trabalho somente se ela estivesse acompanhada das

Figura 17 – Exemplo de discussão conceitual

Fonte**:** https://stackoverflow.com/questions/34046283

tags genéricas. Além disso, ferramentas desconhecidas seriam validadas no *site* da própria ferramenta, comprovando desta maneira sua existência.

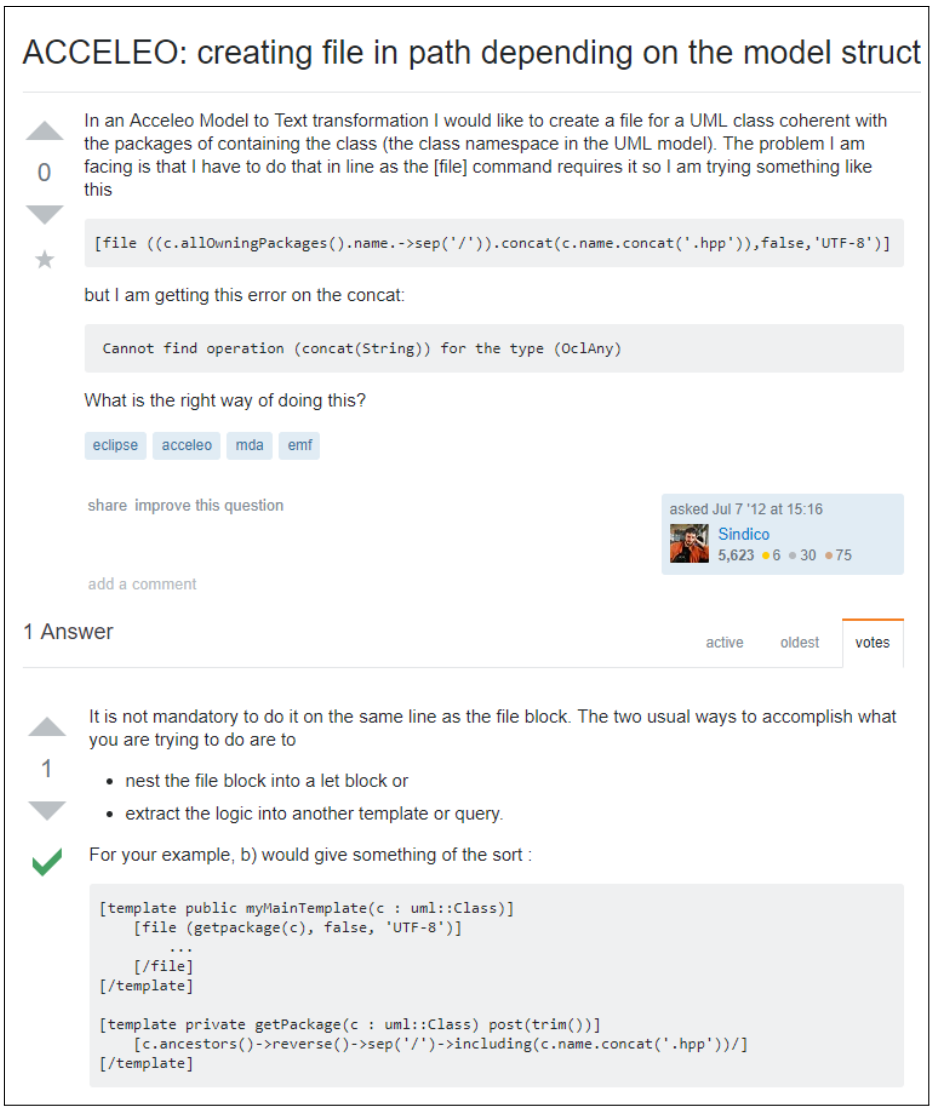
Visando sistematizar esse processo de busca, as seguintes etapas, conforme ilustrado na Figura 21, foram desenvolvidas:

4.3.1 Etapa 1: Busca Primária das Ferramentas

Esta etapa visa efetuar uma busca inicial das ferramentas que se encontram no domínio da MDE. De modo a alcançar esse objetivo, foi utilizado o conjunto das *tags* genéricas, conforme descrito na Tabela 2. Nesse contexto, foram observadas quais *tags*, relacionadas às ferramentas, acompanhavam as *tags* do conjunto genérico. É necessário destacar que, conforme Bajaj, Pattabiraman e Mesbah (2014), os usuários do *Stack Overflow* comumente adicionam várias *tags* para tornar suas questões visíveis durante as buscas na base de dados da plataforma de forma a aumentar a probabilidade de receber uma resposta rapidamente. Baseado nisso, como ilustrado na Figura 22, a partir da *tag* genérica *"mde"* foi identificada uma discussão na qual o usuário também utilizou a *tag "acceleo"* para designar a ferramenta Acceleo[2].

[2] https://www.eclipse.org/acceleo/

Figura 18 – Exemplo de discussão técnica



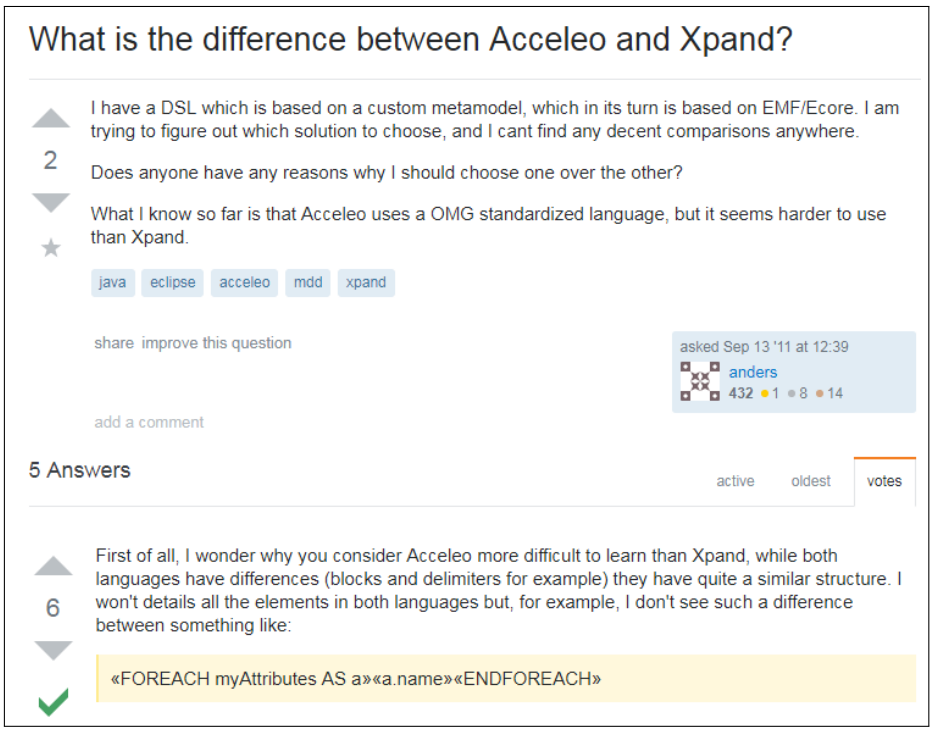
Fonte**:** https://stackoverflow.com/questions/11376162

4.3.2 Etapa 2: Adicionar Novas Tags

Esta etapa objetiva realizar uma nova busca para identificar ferramentas que não foram alcançadas com o conjunto das *tags* genéricas. Para isso, a partir do grupo citado (*tags* genéricas), foram observadas quais *tags* acompanhavam o conjunto genérico. Fundamentado nisso, a nova *tag* identificada é incluída ao conjunto de *tags* genéricas e, em seguida, executa-se novamente a etapa descrita na subseção 4.1.2 de modo a encontrar novas ferramentas. A Tabela 4 apresenta o conjunto final das *tags* genéricas utilizado para auxiliar a busca das ferramentas MDE. Por fim, a *tag "metamodel"* foi adicionada ao conjunto das *tags* genéricas.

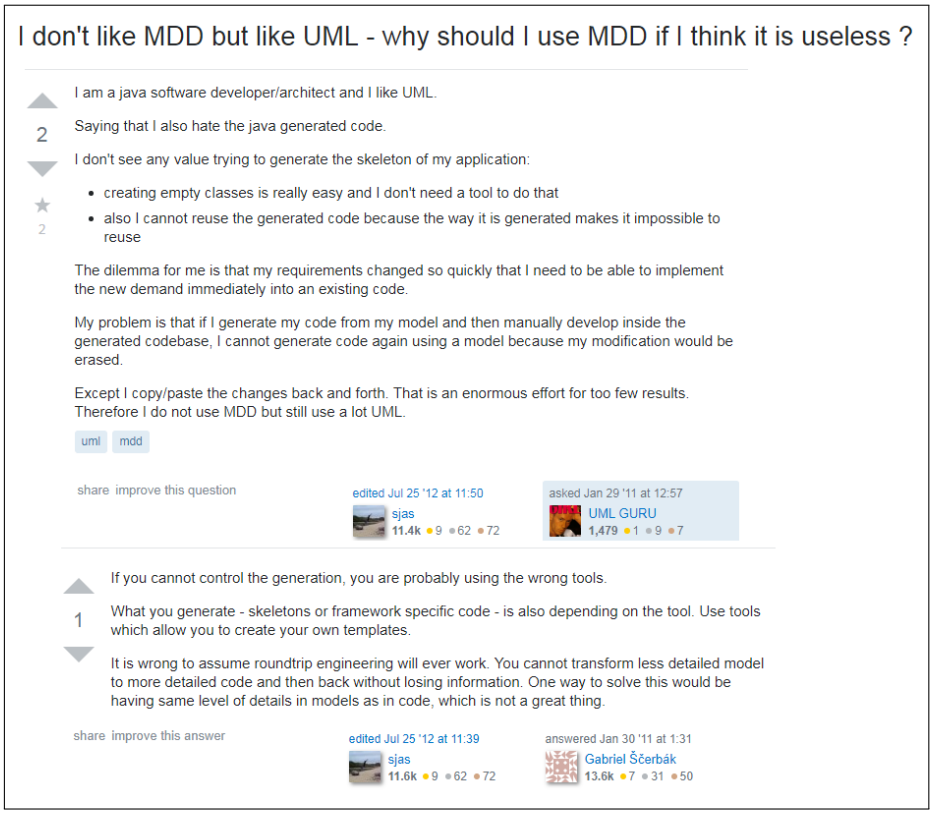
Com base nisso, a partir do primeiro conjunto de *tags* genéricas apresentado na Tabela 2, foi identificado um conjunto inicial contendo *tags* relacionadas às ferramentas MDE. Ao

Figura 19 – Exemplo de discussão técnica/conceitual



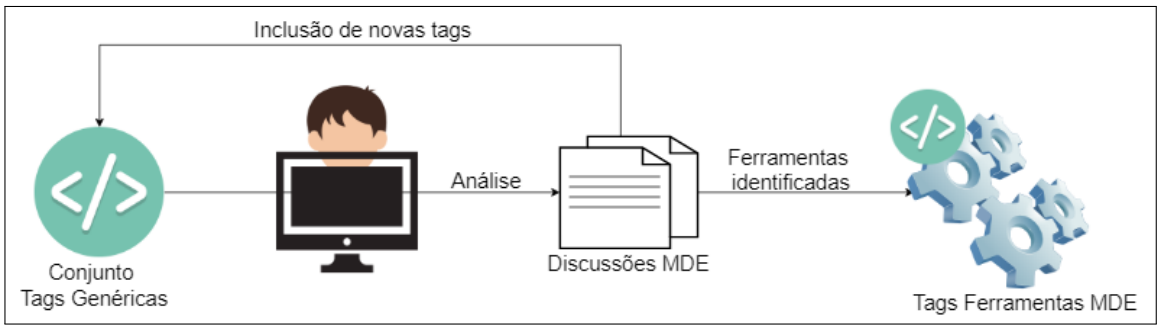
Fonte**:** https://stackoverflow.com/questions/7402003

Figura 20 – Exemplo de discussão de utilidade



Fonte**:** https://stackoverflow.com/questions/4837002

Figura 21 – Terceira Fase da Metodologia Utilizada



Fonte: Elaborado pelo autor

Tabela 4 – Conjunto de Tags Genéricas

	MDE Tags
Tags Genéricas	"mde", "mdsd", "mdd", "model-driven", "model-driven-development", "mda", "metamodel"

Fonte: Elaborado pelo autor

longo do processo, foi observado a presença de uma nova *tag* utilizada pelos desenvolvedores que poderia ser incluída ao conjunto genérico de modo a identificar novas ferramentas. Finalmente, com base no conjunto final de *tags* genéricas, novas *tags* relacionadas às ferramentas MDE foram extraídas. Por fim, a Tabela 5 apresenta o conjunto de *tags* que possuem relação com às ferramentas MDE.

Tabela 5 – Conjunto das Tags de Ferramentas

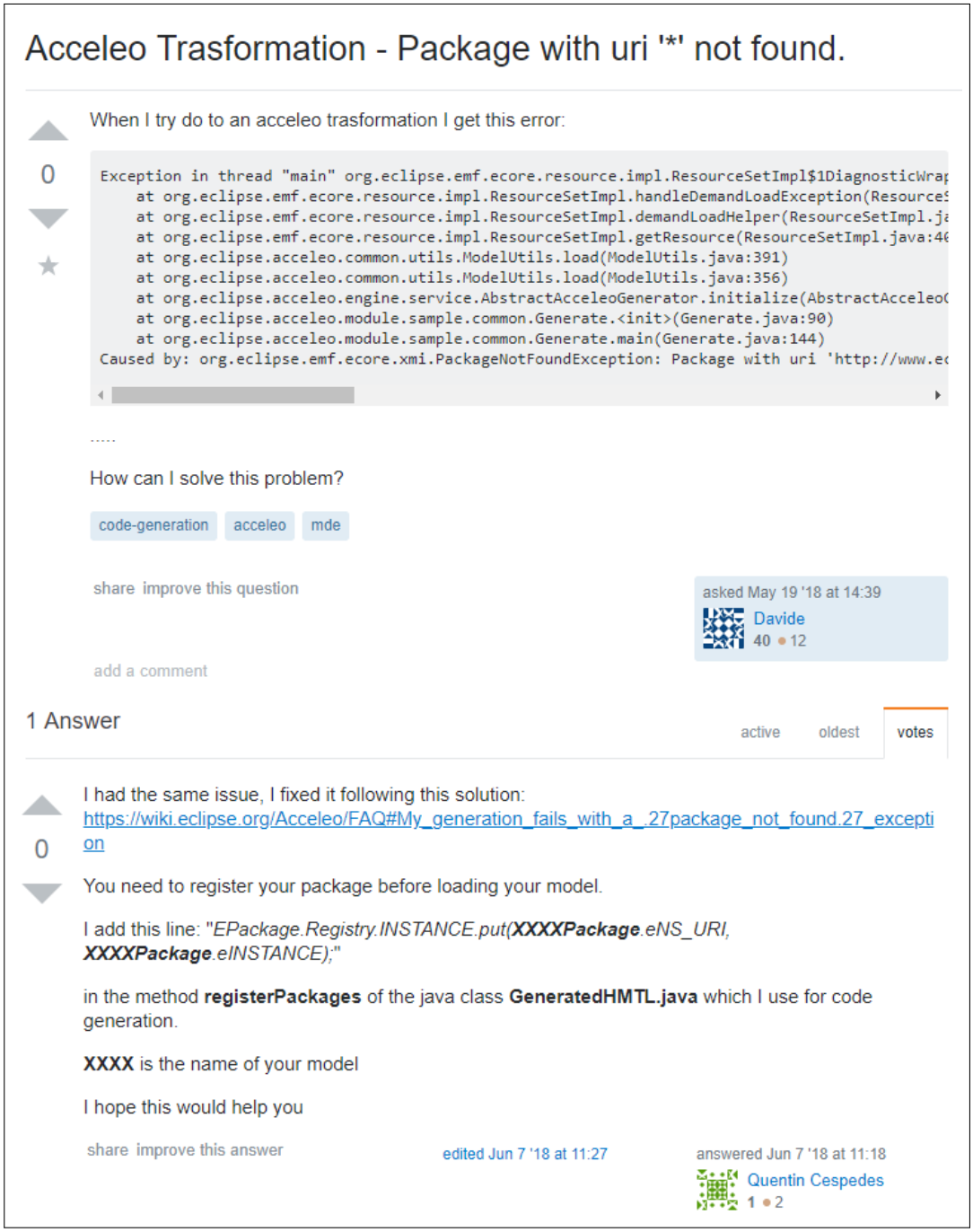
Tags Ferramentas Stack Overflow
"eclipse-emf", "eclipse-emf-ecore", "emf"
"eclipse-sirius"
"eclipse-atl"
"qvt"
"epsilon"
"xtext"
"acceleo"
"xpand"
"xtend"
"ocl"
"papyrus"

Fonte: Elaborado pelo autor

4.4 FASE 4: DISCUSSÕES SOBRE FERRAMENTAS MDE

O objetivo desta fase é identificar as discussões do *Stack Overflow* que envolvem as ferramentas MDE. A partir disso, é possível realizar uma análise nesse conjunto de discussões

Figura 22 – Discussão Identificada com Tag de Ferramenta



Acceleo Trasformation - Package with uri '*' not found.

When I try do to an acceleo trasformation I get this error:

```
Exception in thread "main" org.eclipse.emf.ecore.resource.impl.ResourceSetImpl$1DiagnosticWrap
    at org.eclipse.emf.ecore.resource.impl.ResourceSetImpl.handleDemandLoadException(ResourceS
    at org.eclipse.emf.ecore.resource.impl.ResourceSetImpl.demandLoadHelper(ResourceSetImpl.ja
    at org.eclipse.emf.ecore.resource.impl.ResourceSetImpl.getResource(ResourceSetImpl.java:46
    at org.eclipse.acceleo.common.utils.ModelUtils.load(ModelUtils.java:391)
    at org.eclipse.acceleo.common.utils.ModelUtils.load(ModelUtils.java:356)
    at org.eclipse.acceleo.engine.service.AbstractAcceleoGenerator.initialize(AbstractAcceleoG
    at org.eclipse.acceleo.module.sample.common.Generate.<init>(Generate.java:90)
    at org.eclipse.acceleo.module.sample.common.Generate.main(Generate.java:144)
Caused by: org.eclipse.emf.ecore.xmi.PackageNotFoundException: Package with uri 'http://www.ec
```

How can I solve this problem?

code-generation acceleo mde

share improve this question

asked May 19 '18 at 14:39 Davide 40 • 12

add a comment

1 Answer

active oldest votes

I had the same issue, I fixed it following this solution: https://wiki.eclipse.org/Acceleo/FAQ#My_generation_fails_with_a_.27package_not_found.27_exception

You need to register your package before loading your model.

I add this line: "*EPackage.Registry.INSTANCE.put(**XXXXPackage**.eNS_URI, **XXXXPackage**.eINSTANCE);*"

in the method **registerPackages** of the java class **GeneratedHMTL.java** which I use for code generation.

XXXX is the name of your model

I hope this would help you

share improve this answer

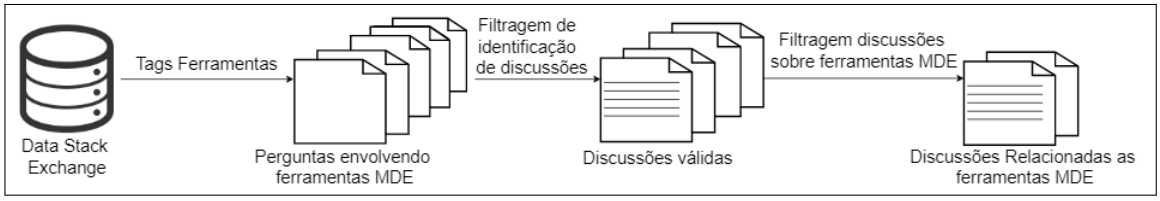
edited Jun 7 '18 at 11:27

answered Jun 7 '18 at 11:18 Quentin Cespedes 1 • 2

Fonte**:** https://stackoverflow.com/questions/50426449

visando responder as perguntas de pesquisa desta dissertação. O processo seguido nesta fase é descrito de acordo com a Figura 23. Objetivando elencar as discussões a serem analisadas, as seguintes etapas foram desenvolvidas:

Figura 23 – Quarta Fase da Metodologia Utilizada



Fonte**:** Elaborado pelo autor

4.4.1 Etapa 1: Identificação das Discussões Válidas

Nesta etapa, a extração dos dados foi obtida através de consultas SQL na base do *Stack Overflow* através do *dataset* disponibilizado pelo *Stack Exchange Data Explorer* utilizando o conjunto de *tags* (Tabela 5) extraídas do processo descrito na Seção 4.3. Após, as perguntas identificadas foram filtradas de modo a reunir as discussões consideradas válidas no contexto desta pesquisa, ou seja, discussões que apresentam pelo menos uma resposta cujo autor é diferente do autor da pergunta, e que servirão de insumo para a próxima etapa.

4.4.2 Etapa 2: Obtenção das Discussões que Envolvem Ferramentas MDE

De posse das discussões que utilizaram as *tags* no domínio das ferramentas, a segunda etapa visa verificar, dentro desse subconjunto de discussões, quais são aquelas que verdadeiramente possuem relação à temática MDE, isto é, excluindo discussões que os usuários utilizaram *tags* que não estão relacionadas às ferramentas MDE. Essa abordagem se faz necessária pois evita a inclusão de ruídos nos dados a serem analisados. Dessa forma, é possível obter resultados confiáveis para responder as perguntas de pesquisa deste trabalho.

4.5 CONSIDERAÇÕES FINAIS

As questões de pesquisa que serão respondidas com base na análise dos dados, bem como todo o processo seguido para realizar a mineração do repositório das plataformas *Stack Overflow* e *Software Engineering Stack Exchange* foram descritos previamente. A partir disso, os resultados gerados serão abordados com mais detalhes no próximo capítulo.

5 RESULTADOS

Neste capítulo são apresentados e discutidos os resultados alcançados a partir da mineração dos repositórios do *Stack Overflow* e do *Software Engineering Stack Exchange* levando em consideração as discussões relacionadas a MDE entre os desenvolvedores. A partir da aplicação do processo descrito no Capítulo 4, os seguintes resultados derivados deste experimento são discutidos nas próximas seções.

5.1 DISCUSSÃO DOS RESULTADOS DA PRIMEIRA FASE

A primeira fase não está ligada diretamente a uma pergunta de pesquisa. Contudo, ela possui relevância por reunir às discussões que irão compor o *corpus* que será analisado de modo a fornecer evidências para responder as perguntas de pesquisa deste trabalho. Fundamentado nisso, foram obtidas 199 questões com suas respectivas respostas nas quais o usuário utilizou as *tags* genéricas. Em seguida, foi aplicado um algoritmo para realizar uma filtragem automática objetivando verificar quais eram as publicações que caracterizavam discussões no contexto desta pesquisa. O *corpus* para análise totalizou 187 discussões válidas obtidas a partir da base do *Stack Overflow*. Por fim, a base de dados do *Software Engineering Stack Exchange*, replicando o mesmo processo aplicado ao SO, retornou um total de 11 discussões válidas para análise.

A partir das 187 discussões válidas extraídas do *Stack Overflow*, era necessário verificar quais discussões estavam relacionadas à temática deste estudo. Para isso, foram descartadas as discussões nas quais os usuários utilizaram *tags* que não possuíam ligação com MDE. Além disso, foram analisados os conteúdos de cada discussão de modo a validar se o assunto discutido entre os desenvolvedores estava associado a MDE.

Após a seleção das discussões relacionadas a MDE, o subconjunto de discussões do *Stack Overflow*, que estão incluídas no escopo dessa pesquisa, totalizou em 74. Diante do mesmo processo, a partir da plataforma *Software Engineering Stack Exchange*, foram extraídas apenas 9 discussões relevantes.

Por fim, os resultados alcançados no fim desta fase, servirão como entrada à próxima etapa.

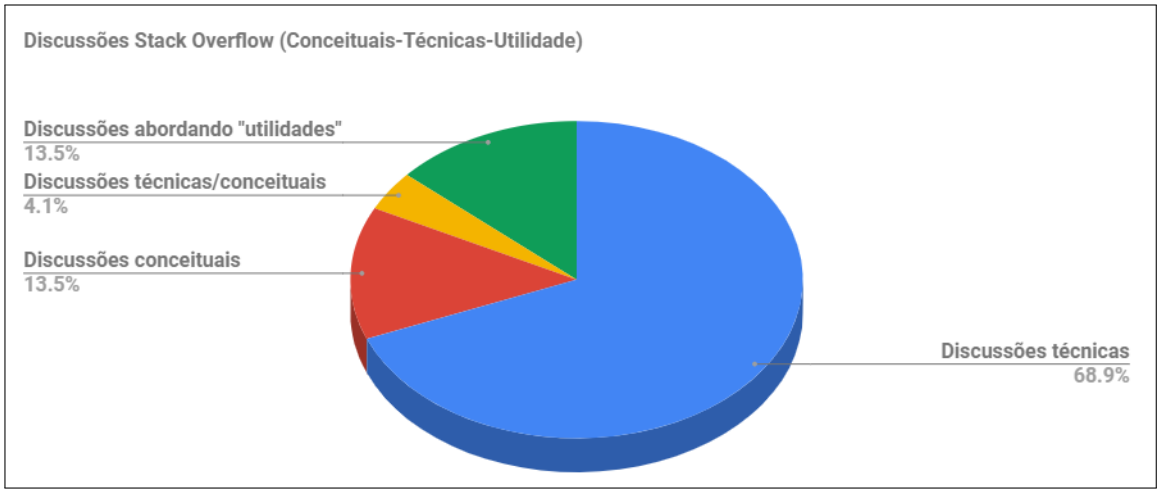
É necessário ressaltar que devido a quantidade de discussões obtidas, a abordagem por tópicos implementada pelo LDA não foi utilizada. Essa abordagem possui o potencial de melhorar a busca e a extração de temas semânticos a partir de documentos escritos em linguagem natural. Contudo, quando as coleções são constituídas por poucos documentos, a abordagem por

tópicos pode ser menos coerente e menos interpretável (BOYD-GRABER; MIMNO; NEWMAN, 2014). Além disso, essa abordagem é aplicável em grandes coleções de dados (BLEI; NG; JORDAN, 2003). Diante do cenário apresentado, as discussões neste trabalho são analisadas de forma manual. Vale salientar que trabalhos que realizam a análise manual das discussões em plataformas baseadas em perguntas e respostas são encontrados na literatura (KAHANI *et al.*, 2016; LOPEZ *et al.*, 2019; TREUDE; BARZILAY; STOREY, 2011).

5.2 DISCUSSÃO DOS RESULTADOS DA SEGUNDA FASE - QP1 E QP2

Com base no conteúdo das discussões, foram identificados debates que envolviam diferentes perspectivas sobre o assunto em estudo. A partir disso, objetivando uma melhor análise e para fornecer evidências para responder a **QP1 - *Quais temas relacionados a MDE os desenvolvedores têm discutido nas plataformas de perguntas e respostas em Engenharia de Software?* -**, as discussões foram categorizadas em: técnicas, conceituais, técnicas/conceituais e utilidade. Assim, levando em consideração a extração dos dados derivados da segunda fase (Seção 4.2), os resultados gerados a partir desta categorização são ilustrados na Figura 24.

Figura 24 – Categorização das discussões no Stack Overflow



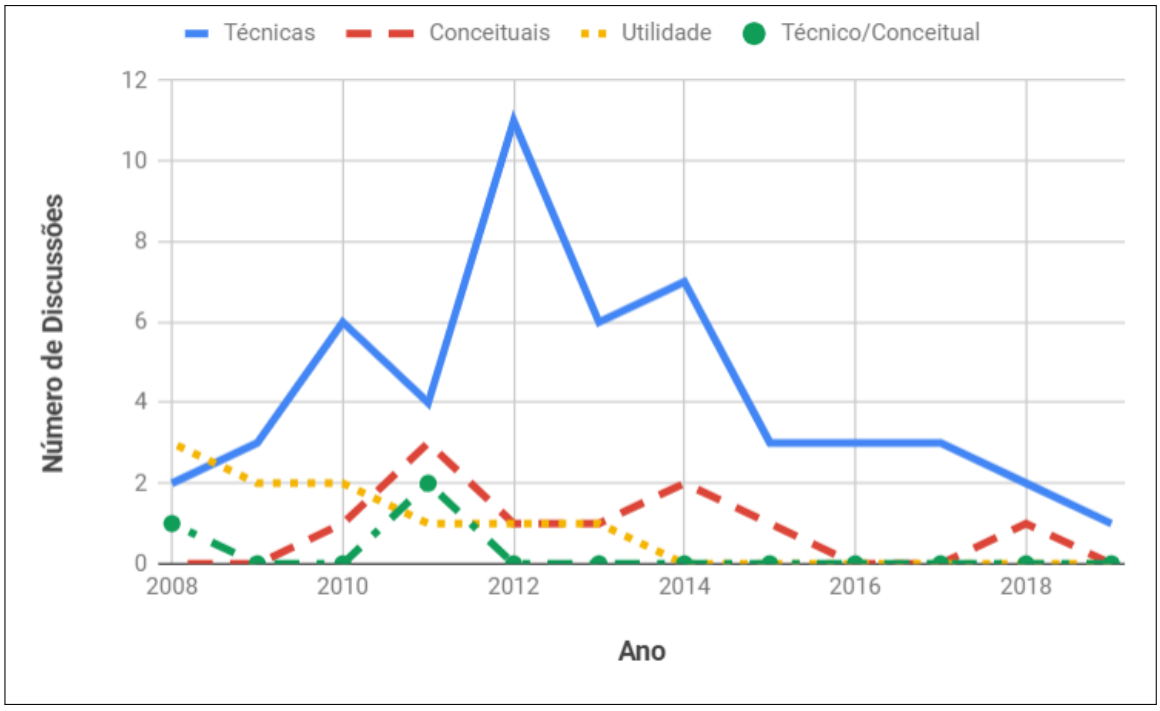
Fonte**:** Elaborado pelo autor

Apoiado nos critérios de validação, foram obtidos um total de 74 discussões em consonância com o objetivo deste estudo. Destas, aproximadamente 69% foram categorizadas como discussões técnicas, ou seja, discussões relacionadas a dúvidas específicas de alguma ferramenta utilizada no domínio da MDE. Por outro lado, apenas 13% são categorizadas como discussões conceituais, ou seja, questões que discorriam sobre os conceitos da MDE. Outra categoria, com

percentual similar ao das discussões conceituais, foi a das discussões que envolviam o tema utilidade, isto é, discussões que tratavam sobre viabilidade, benefícios, riscos ou exemplo de aplicações reais da MDE. Por fim, o grupo das discussões que tratavam tanto questões técnicas e conceituais refletiam apenas 4% de todas as discussões.

A partir de cada categoria, foram extraídas informações visando entender a evolução temporal das discussões analisadas. Acerca disso, a Figura 25 ilustra a quantidade de discussões por ano no intervalo de 2008 a 2019. Os resultados indicam que, dentro desse intervalo, as questões técnicas ainda permanecem como as discussões mais relevantes entre os usuários do *Stack Overflow*, sendo um tema recorrente nessa plataforma.

Figura 25 – Número de discussões por ano na plataforma Stack Overflow



Fonte: Elaborado pelo autor

As discussões conceituais, grande parte concentrada no intervalo 2010 a 2015, estavam mais presentes nos debates entre os desenvolvedores interessados nessa temática mais teórica da MDE. A diminuição da ocorrência desse tipo de questão pode indicar que, gradativamente, os conceitos teóricos da MDE estão mais consolidados entre a comunidade interessada nessa abordagem. Por outro lado, as discussões que envolvem a utilidade da MDE estão concentradas somente no período de 2008 a 2013. Esse fato provavelmente reforça que atualmente os desenvolvedores estejam mais maduros quanto ao papel da MDE e seus benefícios no processo de desenvolvimento de software.

Objetivando entender o que os desenvolvedores da plataforma *Stack Overflow* estão discutindo no aspecto técnico, uma análise mais detalhada das discussões foi realizada. A partir do conteúdo das discussões, foram elencadas categorias visando compreender o contexto do que estava sendo discutido pelos desenvolvedores.

As discussões técnicas foram divididas nas seguintes categorias: *metamodelagem, plugin, restrições em modelos, sugestão de ferramentas, integração/interoperabilidade entre ferramentas, versionamento de modelos, transformações M2T, transformações M2M, sintaxe textual concreta, aplicação exemplo*. Os resultados obtidos nessa categoria são ilustrados na Figura 26.

No âmbito das discussões técnicas, cerca de 31% destas abordavam a temática metamodelagem. Além disso, aproximadamente, 14% das discussões tratavam sobre transformações M2T; por outro lado, um pouco mais de 5% abordavam a temática relacionada à transformações M2M. Por essa análise, é possível destacar que, no contexto das discussões da plataforma *Stack Overflow* no qual o usuário utilizou as *tags* do grupo genérico, os desenvolvedores possuem um maior interesse na abordagem M2T em comparação à M2M. Vale salientar que aproximadamente 20% das discussões envolvem solicitações por parte dos usuários sobre sugestões de ferramentas MDE para diferentes contextos.

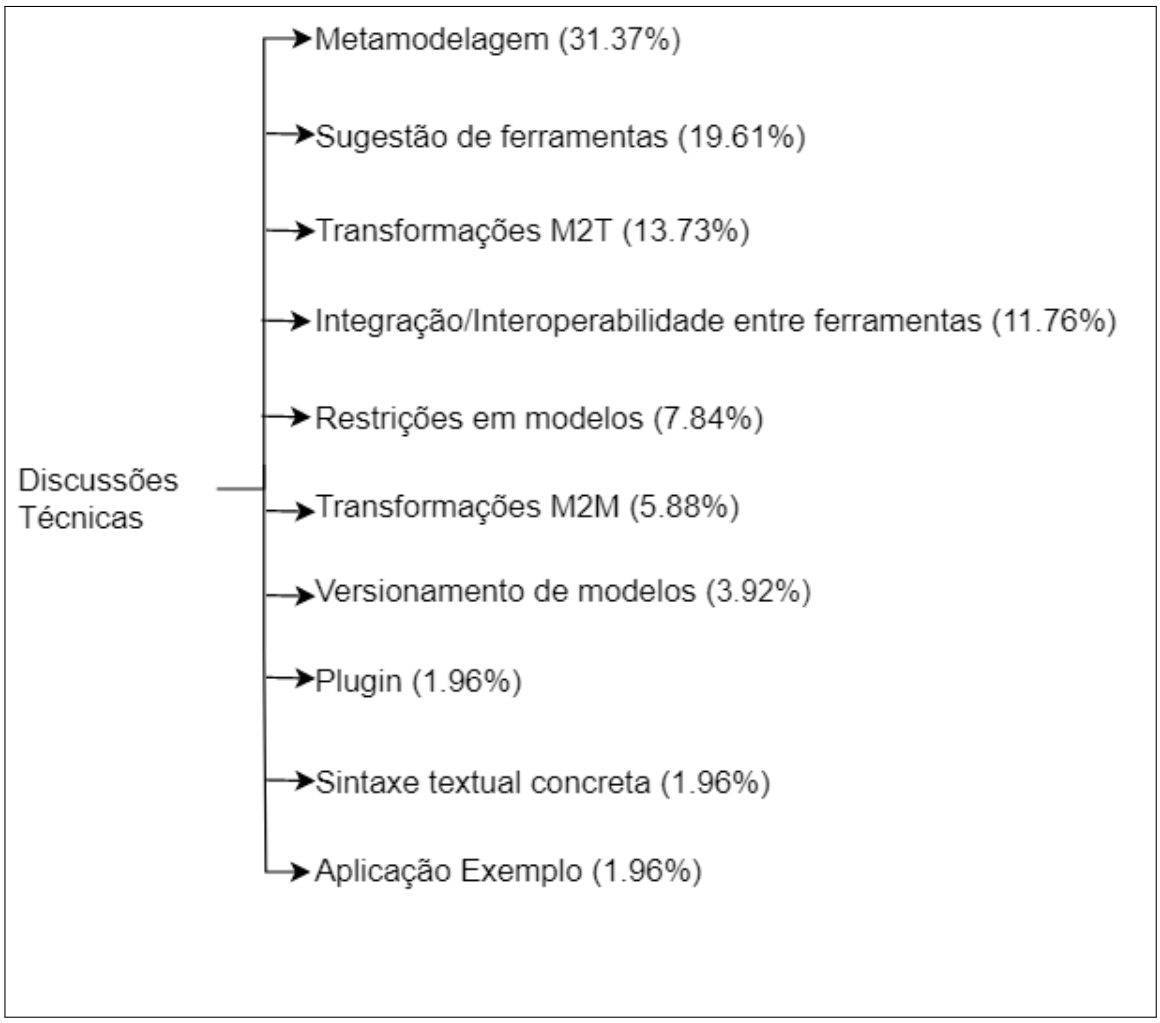
Ainda sobre essas discussões, foram extraídas as categorias tendo como objetivo investigar, dentro das discussões técnicas, quais são os assuntos mais debatidos pelos usuários no ambiente do *Stack Overflow* nos últimos quatro anos[1]. Apoiado nessa premissa, os assuntos do domínio técnico que estão sendo debatidos pelos desenvolvedores são: metamodelagem, transformações M2T, restrições em modelos e, por fim, integração/interoperabilidade entre ferramentas.

Outra análise realizada a partir das discussões técnicas, a fim de responder a **QP2 - *Quais são os domínios que MDE está sendo empregada?* -**, se deu na identificação dos domínios que os usuários estavam aplicando MDE. A Tabela 6 exibe os domínios que foram possíveis de serem extraídos com base nas discussões do *Stack Overflow* de modo a elencar o contexto que a referida abordagem estava sendo empregada. É necessário ressaltar que existem discussões nas quais os desenvolvedores deixam claro quais são os domínios que a MDE está sendo aplicada. Sendo assim, os domínios apresentados neste trabalho são oriundos das discussões que os desenvolvedores explicitaram o contexto onde MDE estava sendo aplicada.

A partir deste resultado, é possível elencar que MDE não está restrita em poucos

[1] visando identificar os assuntos mais recentes debatidos pelos desenvolvedores no contexto da MDE

Figura 26 – Categorização das discussões técnicas no Stack Overflow



Fonte**:** Elaborado pelo autor

domínios. Esse fato também é corroborado no estudo de Mohagheghi e Dehlen (2008). Além disso, os desenvolvedores têm discutido sobre a utilização de MDE em aplicações voltadas para computação em nuvem e dispositivos móveis. A utilização de MDE nesses domínios são temas de interesse em outras pesquisas acadêmicas (BRUNELIERE; CABOT; JOUAULT, 2010; FERRY *et al.*, 2014; HEITKOTTER; MAJCHRZAK; KUCHEN, 2013). Adicionalmente, trabalhos acadêmicos que exploram o uso de MDE em outros domínios como, por exemplo, Microsserviços e Internet das Coisas (*Internet of Things* - IOT), são discutidos na literatura (RADEMACHER; SORGALLA; SACHWEH, 2018; THRAMBOULIDIS; VACHTSEVANOU; SOLANOS, 2018). Contudo, os desenvolvedores nas plataformas do *Stack Overflow* ou do *Software Engineering Stack Exchange* ainda não estão debatendo sobre o uso de MDE nesses domínios. Vale ressaltar que apesar do trabalho de Kahani *et al.* (2016) realizar uma análise sobre o contexto da abordagem MDE na plataforma do Eclipse, o autor não indica os domínios

que a referida abordagem está sendo empregada. Sendo assim, os resultados desta pergunta de pesquisa elenca os domínios que os desenvolvedores possuem interesse na plataforma *Stack Overflow*.

Tabela 6 – Domínios Identificados nas Discussões Técnicas

Domínio
Sistemas Embarcados/ Sistemas real-time
Desenvolvimento para dispositivos móveis
Indústria automotiva
Aplicações Web
Aplicações Empresariais
Aplicações na nuvem

Fonte**:** Elaborado pelo autor

Finalmente, aplicando a mesma metodologia na plataforma *Software Engineering Stack Exchange*, foram obtidos os seguintes resultados a partir do conjunto das 9 discussões: 4 tratavam sobre utilidade, 3 norteavam sobre discussões conceituais e, por fim, 2 discorriam sobre aspectos técnicos. As discussões nesta plataforma se concentraram, de modo geral, no intervalo anual de 2011 a 2013, alcançando um pico no ano de 2011. Nesse cenário, as discussões abordando os aspectos técnicos da MDE debatiam sobre metamodelagem e tratamento de exceções em transformações de modelos.

5.3 DISCUSSÃO DOS RESULTADOS DA TERCEIRA FASE

A terceira fase não está ligada diretamente a uma pergunta de pesquisa. Contudo, ela fornece meios para identificar quais são as *tags* relacionadas às ferramentas MDE que estão sendo utilizadas pelos desenvolvedores na plataforma *Stack Overflow*. Para tanto, foram utilizadas as *tags* do conjunto genérico (apresentado na Tabela 2) para obter as *tags* relacionadas às ferramentas MDE necessárias à próxima fase.

Com base no que foi discutido, em uma busca inicial nas discussões relacionadas a MDE, resultantes da primeira fase, foram obtidas 8 ferramentas. Após, de modo a identificar novas ferramentas, uma nova *tag* (*"metamodel"*) foi incorporada ao conjunto genérico (apresentado na Tabela 4). Desta nova inclusão, 3 novas ferramentas foram adicionadas. Por fim, 11 ferramentas MDE foram identificadas, são elas: *Eclipse Modeling Framework - EMF*[2], *Sirius*[3],

[2] https://projects.eclipse.org/projects/modeling.emf.emf
[3] https://www.eclipse.org/sirius/

ATL Transformation Language - ATL[4], *QVT*[5], *Epsilon*[6], *Xtext*[7], *Acceleo*, *Xpand*[8], *Xtend*[9], *Object Constraint Language - OCL*[10], *Papyrus*[11]. As ferramentas identificadas e suas respctivas *tags* disponíveis na plataforma *Stack Overflow* são apresentadas na Tabela 7. Resultado similar foi obtido no estudo de Kahani *et al.* (2016). Contudo, a ferramenta *Sirius* não foi identificada pelos autores nas discussões dos fóruns na plataforma Eclipse.

Tabela 7 – Conjunto Tags Ferramentas

Ferramentas	Tags Stack Overflow
EMF	"eclipse-emf" "eclipse-emf-ecore" "emf"
Sirius	"eclipse-sirius"
ATL	"eclipse-atl"
QVT	"qvt"
Epsilon	"epsilon"
Xtext	"xtext"
Acceleo	"acceleo"
Xpand	"xpand"
Xtend	"xtend"
OCL	"ocl"
Papyrus	"papyrus"

Fonte**:** Elaborado pelo autor

5.4 DISCUSSÃO DOS RESULTADOS DA QUARTA FASE - QP3

O objetivo da quarta fase era reunir o *corpus* contendo as discussões da plataforma *Stack Overflow* que envolvem as ferramentas MDE a partir das *tags* extraídas do processo discutido na seção 5.3. Além disso, essa etapa contribuiu para alcançar os resultados que serviram de base para responder a **QP3 - *Quais são as ferramentas MDE discutidas pelos desenvolvedores?* -** e que serão discutidos a seguir. O *corpus* obtido engloba discussões excluindo as *tags* do grupo genérico, ou seja, estão presentes apenas discussões nas quais o usuário utilizou as *tags* da Tabela 7. Com isso, o somatório de cada discussão das ferramentas analisadas totalizou em 2.128 discussões. Vale ressaltar que para facilitar a compreensão em relação à responsabilidade desempenhada por cada ferramenta no processo da MDE, uma

[4] https://www.eclipse.org/atl/
[5] https://projects.eclipse.org/projects/modeling.mmt.qvt-oml
[6] https://www.eclipse.org/epsilon/
[7] https://www.eclipse.org/Xtext/
[8] https://wiki.eclipse.org/Xpand
[9] https://www.eclipse.org/xtend/
[10] https://projects.eclipse.org/projects/modeling.mdt.ocl
[11] https://www.eclipse.org/papyrus/

categorização foi criada de acordo com os conceitos expostos em Brambilla, Cabot e Wimmer (2012) e Kahani *et al.* (2016). A Tabela 8 apresenta as categorias de cada ferramenta.

Tabela 8 – Categoria das Ferramentas

Categoria	Ferramenta		
Frameworks de modelagem	EMF	Sirius	
Model to Model	ATL	QVT	Epsilon
Model to Text	Acceleo	Xpand	Xtend
Ferramentas de apoio para modelagem	OCL	Papyrus	
Ferramenta modelagem Textual	Xtext		

Fonte**:** Elaborado pelo autor

Na plataforma *Stack Overflow*, os usuários comumente utilizam várias *tags* para rotular suas questões. Com base nisso, a partir do *corpus* das discussões sobre ferramentas, foram identificadas quais destas ocorriam acompanhadas com as outras *tags* do conjunto de *tags* de ferramentas. O resultado desta análise é exibido na Tabela 9 e explicado a seguir.

Tendo em vista uma melhor compreensão sobre o cenário apresentado, cada ferramenta foi agrupada de acordo com o papel que desempenha nos processos da MDE. Conforme descrito na Tabela 9, a classificação das ferramentas é concebida da seguinte maneira:

- **Metamodelagem (criação de metamodelos):** ferramentas *EMF* e *Sirius*;
- **Transformação M2M (transformação entre modelos):** ferramentas *ATL*, *QVT* e *Epsilon*;
- **Sintaxe textual concreta:** ferramenta *Xtext*;
- **Transformação M2T (transformação modelo para texto):** ferramentas *Acceleo*, *Xpand* e *Xtend*;
- **Ferramentas de apoio para modelagem (aplicação de restrições em modelos/ auxílio na criação-edição gráfica dos modelos):** ferramentas *OCL*[12], *Papyrus*.

Os números representam a coocorrência das *tags* do grupo das ferramentas MDE, por exemplo, as interseções entre mesmas *tags* de ferramentas exibem a quantidade de discussões que uma determinada *tag* foi utilizada de forma única, isto é, não houve uma ocorrência simultânea de outras *tags* do grupo de ferramentas MDE.

Nesse contexto, por exemplo, as *tags* que representam a ferramenta *EMF* foram usadas unicamente em 440 discussões, ou seja, excluindo o conjunto de *tags* das outras ferramentas.

[12] Apesar de OCL ser uma linguagem, foi incluída como ferramenta por uma questão de generalização de uso e para facilitar o agrupamento

Uma outra situação possível ocorre quando um usuário utilizou mais de uma *tag* em seu *post*. Diante da circunstância apresentada, a Tabela 9 informa a quantidade de vezes na qual uma determinada *tag* foi utilizada simultaneamente com outra.

Tabela 9 – Ferramenta x Ferramenta

	Metamodelagem		**Transformação M2M**			**Sintaxe textual concreta**	**Transformação M2T**			**Ferramentas de apoio para modelagem**	
	EMF	**Sirius**	**ATL**	**QVT**	**Epsilon**	**Xtext**	**Acceleo**	**Xpand**	**Xtend**	**OCL**	**Papyrus**
EMF	440	9	2	6	2	72	21	3	16	21	4
Sirius	9	14	0	0	0	4	1	0	2	1	0
ATL	2	0	0	0	1	2	0	0	0	0	0
QVT	6	0	0	4	0	0	0	0	0	0	1
Epsilon	2	0	1	0	6	0	0	0	0	1	0
Xtext	72	4	2	0	0	670	2	13	152	1	1
Acceleo	21	1	0	0	0	2	66	1	1	9	6
Xpand	3	0	0	0	0	13	1	5	6	0	0
Xtend	16	2	0	0	0	152	1	6	129	0	0
OCL	21	1	0	0	1	1	9	0	0	86	3
Papyrus	4	0	0	1	0	1	6	0	0	3	66

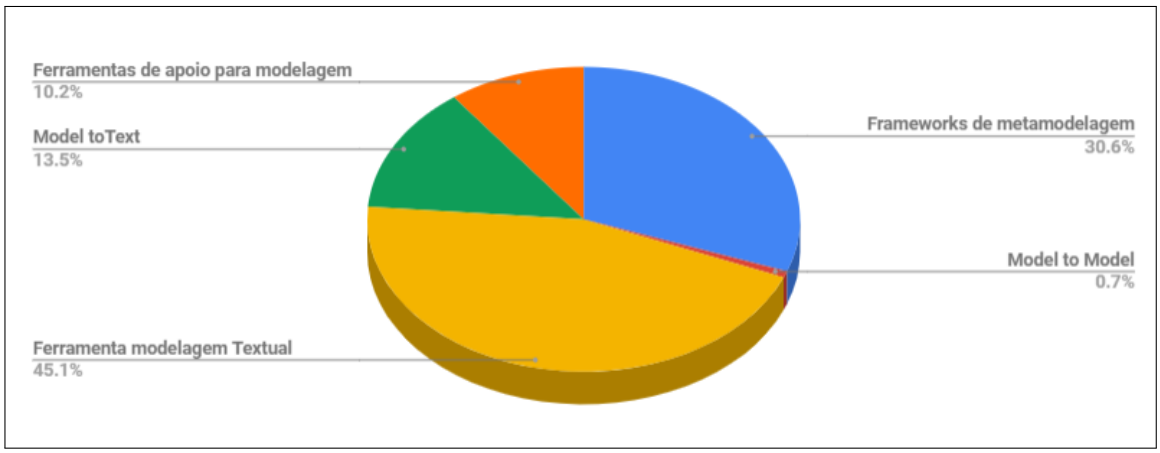
Fonte**:** Elaborado pelo autor

Assim, foram somados os resultados de cada interseção entre mesma *tag* de ferramenta por etapa (*Metamodelagem, Transformações M2M, Sintaxe textual concreta, Transformações M2T, Ferramentas de apoio para modelagem*) de modo a obter uma estimativa identificando quais dessas ferramentas e suas respectivas abordagens estão sendo mais discutidas pelos desenvolvedores. O total de discussões únicas, geradas a partir das interseções, resultam em 1486. Com base nisso, conforme mostra a Figura 27, é possível observar que pelo menos cerca de 45% discutem sobre (ou utilizam) uma ferramenta de modelagem textual, enquanto que cerca de 30% envolvem debates sobre *frameworks* de metamodelagem. Além disso, aproximadamente 13% das discussões utilizaram as *tags* das ferramentas relacionadas à transformação M2T. Por outro lado, menos de 1% utilizavam as *tags* das ferramentas ligadas à temática relacionada à transformação M2M.

Apoiado nessa análise, é possível destacar que, no contexto das discussões da plataforma *Stack Overflow* no qual o usuário utilizou as *tags* do grupo das ferramentas MDE, os desenvolvedores possuem um maior interesse nas ferramentas associadas à abordagem M2T. É necessário ressaltar que a *tag* de ferramenta que mais ocorre simultaneamente com as outras é a *tag "emf"* da categoria *Metamodelagem*. Isso pode ser um indicativo que os desenvolvedores estão aplicando metamodelagem na prática, permeando por todas as fases da MDE.

A partir dos resultados, e conforme apresentado na Tabela 9, é possível elencar quais ferramentas, dentro dos diversos processos da MDE, obtiveram um maior destaque entre os usuários da plataforma *Stack Overflow*. No âmbito das ferramentas de metamodelagem, a

Figura 27 – Categorização das discussões sobre ferramentas no Stack Overflow



Fonte**:** Elaborado pelo autor

ferramenta *EMF* possui uma maior relevância. No contexto das ferramentas que se propõem a realizar transformações entre modelos (M2M), a ferramenta *Epsilon* tem gerado mais discussões por parte da comunidade.

Por outro lado, *Xtext* ainda continua se mantendo a ferramenta mais adotada pelos desenvolvedores para geração da sintaxe textual concreta. Além disso, esta ferramenta pode ser utilizada para o desenvolvimento de linguagens específicas de domínio. O interesse da comunidade dos desenvolvedores por esse tipo de ferramenta está alinhado com os resultados obtidos por Hutchinson *et al.* (2011), mostrando que muitos desenvolvedores têm utilizado DSLs em projetos pessoais.

No que tange às ferramentas utilizadas para realizar transformações M2T, *Xtend* está se estabelecendo como uma nova opção aos desenvolvedores, tendo em vista que a ferramenta *Acceleo* é tida, conforme Brambilla, Cabot e Wimmer (2012), como protagonista em relação às ferramentas M2T.

No domínio das ferramentas de apoio para modelagem as mais utilizadas, com base no resultado da análise, são: *OCL* e *Papyrus*. Vale salientar que ambas podem ser utilizadas em conjunto (ECLIPSE, 2019).

Realizando uma comparação com o trabalho de Kahani *et al.* (2016), as ferramentas relacionadas a metamodelagem são mais discutidas nas plataformas de perguntas e respostas do Eclipse. Em seguida, ferramentas de modelagem textual (Xtext) estão entre as ferramentas que geram mais discussões nos fóruns do Eclipse. Por outro lado, no contexto do *Stack Overflow*, as ferramentas de modelagem textual estão entre as ferramentas que agregam maior interesse da comunidade. Após, as ferramentas de metamodelagem seguem como sendo outra preferência

entre os desenvolvedores.

É necessário ressaltar que além da análise apresentada ao longo deste capítulo, foi realizado um levantamento com base nas métricas *ViewCount* e *AnswerCount*, fornecidas pelo *Stack Overflow*. Conforme Bandeira *et al.* (2019), estas métricas refletem diferentes aspectos de popularidade de um *post* na referida plataforma. Além disso, essas métricas caracterizam a popularidade de uma discussão considerando dois grupos, o grupo de usuários que acessou as discussões de um tema específico e o grupo de usuários que interagiu com as discussões nas quais tiveram acesso, respectivamente. Os resultados extraídos a partir dessas métricas serviram de base para responder a **QP4 -** ***Quais temas sobre MDE atraíram mais a atenção dos desenvolvedores?***-.

Com base nas métricas analisadas a partir das discussões da plataforma *Stack Overflow*, os seguintes resultados foram extraídos:

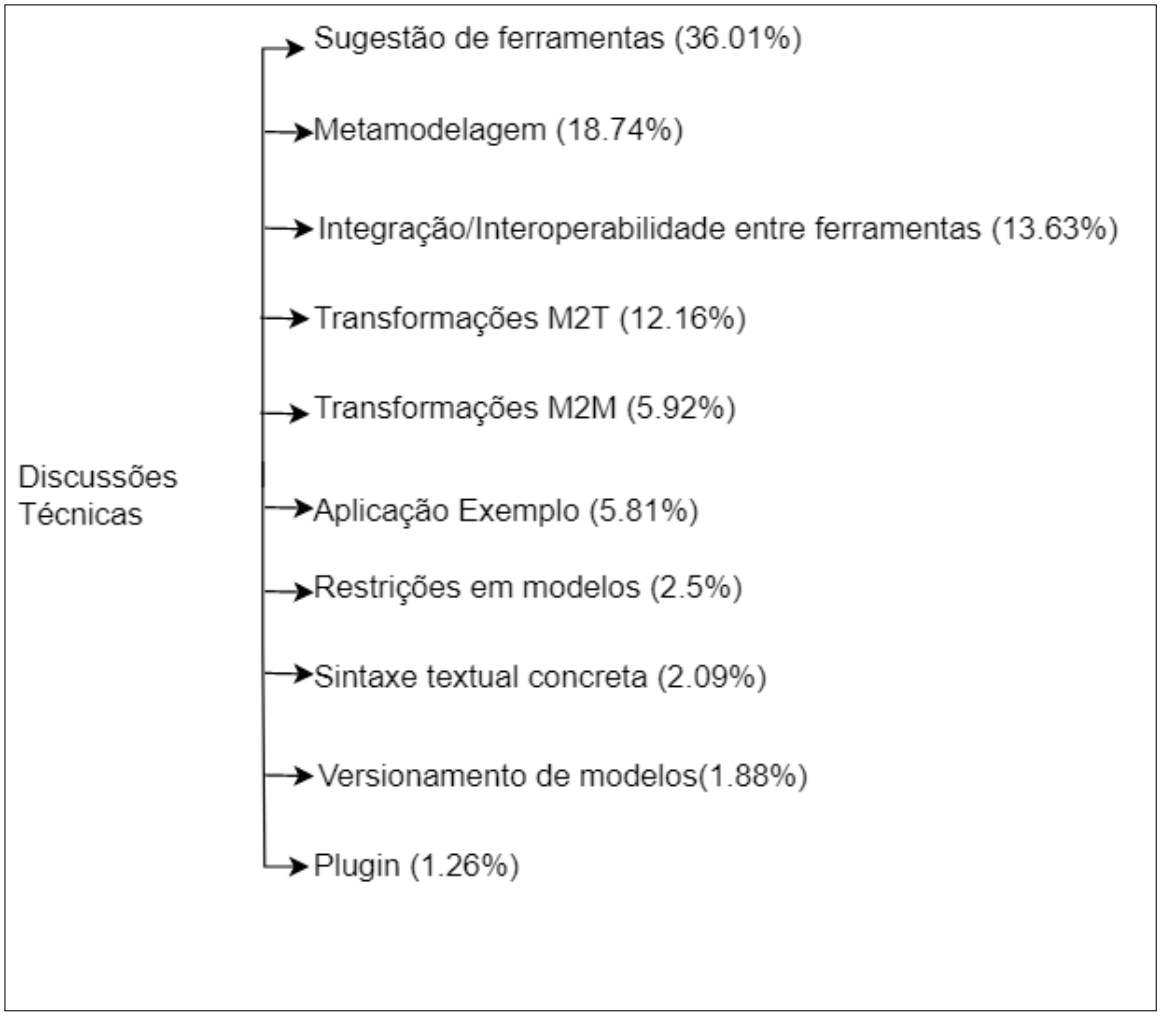
- **ViewCount:** a métrica *ViewCount* das discussões técnicas totalizou em 55.238 visualizações;
- **AnswerCount:** a métrica *AnswerCount* das discussões técnicas totalizou em 98 respostas fornecidas pelos usuários;

No grupo das discussões técnicas a categoria *"metamodelagem"* gerou mais discussões, conforme apresentado na Figura 26. Porém, tendo como base a métrica *ViewCount*, a categoria relacionada ao tema *"sugestão de ferramenta"* atraiu mais atenção da comunidade, conforme demonstra a Figura 28. Nesse contexto, uma possível justificativa, seria o fato dos desenvolvedores estarem interessados em conhecer as ferramentas que fornecem suporte à MDE que atendam às suas necessidades. Além disso, o tema que gerou mais discussões não foi o mesmo que dominou o interesse dos desenvolvedores.

Vale salientar que a categoria *"versionamento de modelos"*, apesar de possuir uma quantidade um pouco maior em números de discussões do que outras categorias, foi um dos temas que menos atraiu atenção dos desenvolvedores. Segundo Forward, Badreddin e Lethbridge (2010), é reportado que o maior impedimento no uso de modelos pelos desenvolvedores é a falta de sincronização entre modelos e códigos, ou seja, modelos tendem a ficar desatualizados em relação ao código. Logo, apesar dos resultados destacarem que este tema possui baixo interesse entre os desenvolvedores, esse assunto possui relevância e deveria ser mais discutido pela comunidade. Além disso, é notório que a comunidade possui maior interesse nas transformações M2T em relação às transformações M2M. Este aspecto também é destacado em outros estudos

presentes na literatura (TORCHIANO *et al.*, 2013).

Figura 28 – Categorização das discussões técnicas no Stack Overflow pela métrica ViewCount

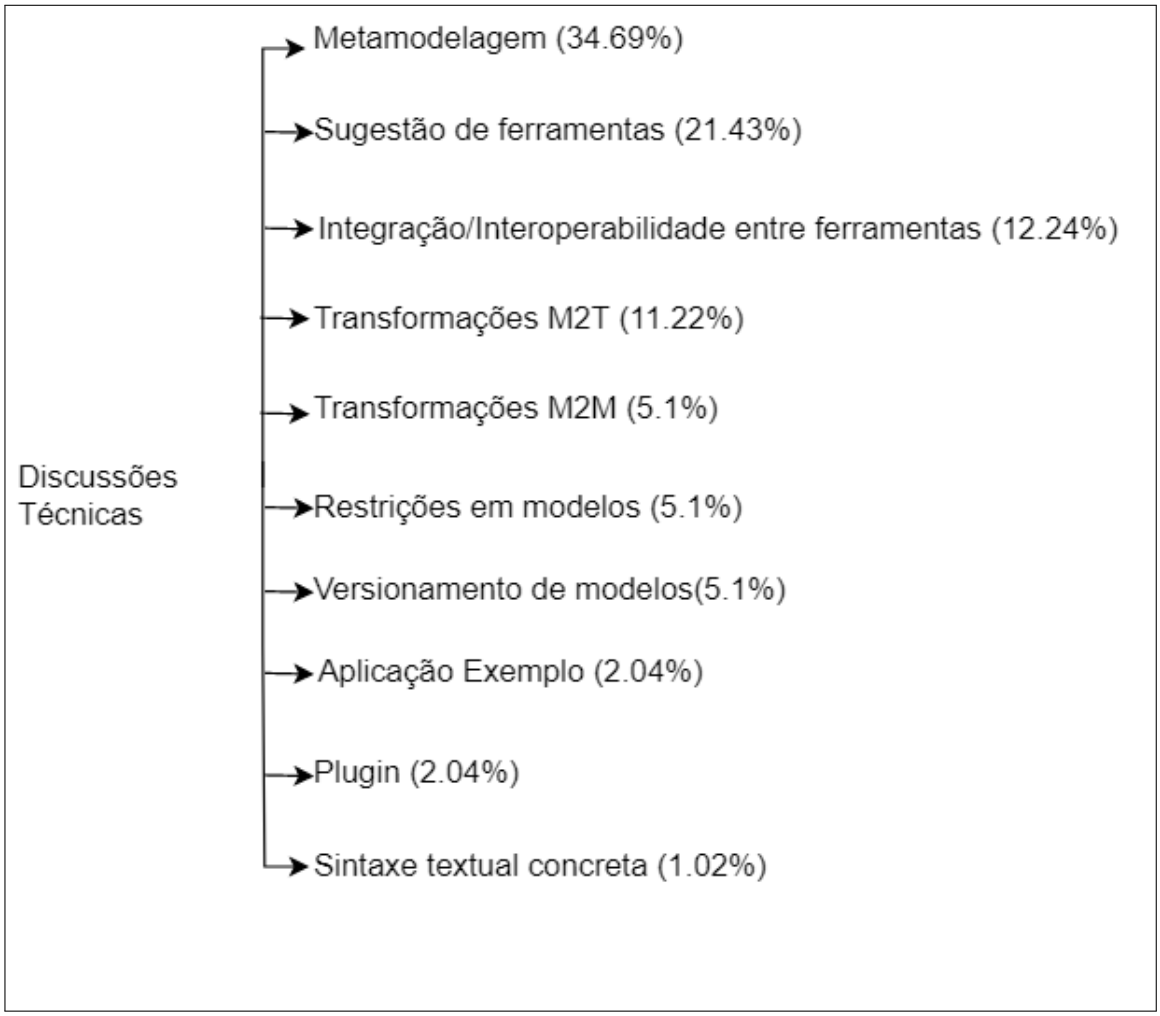


Fonte**:** Elaborado pelo autor

Em uma outra perspectiva sobre o grupo das discussões técnicas, tendo como base a métrica *AnswerCount*, a categoria relacionada ao tema *"metamodelagem"* proporcionou um maior engajamento da comunidade. Portanto, o tema que gerou mais discussões foi o mesmo que reuniu uma participação maior entre desenvolvedores. Por outro lado, a categoria relacionada à sintaxe textual concreta, foi o tema que menos agregou uma interação entre os desenvolvedores. A Figura 29 ilustra os resultados para cada categoria levando em consideração a métrica *AnswerCount*.

No contexto das discussões conceituais, os temas relacionados a UML e conceitos gerais de MDD, por exemplo, transformações entre modelos e geração de código a partir de modelos, figuram entre os principais assuntos que atraem tanto a atenção como a participação dos

Figura 29 – Categorização das discussões técnicas no Stack Overflow pela métrica AnswerCount



Fonte**:** Elaborado pelo autor

desenvolvedores. Nesse contexto, estudos no âmbito da MDE mostram que UML é a notação de modelagem mais utilizada pelos desenvolvedores (FORWARD; BADREDDIN; LETHBRIDGE, 2010; HUTCHINSON *et al.*, 2011; TORCHIANO *et al.*, 2011)

No âmbito das questões sobre utilidade da abordagem, discussões sobre benefícios, riscos, viabilidade e perspectivas futuras abordando MDA e MDD agregam tanto a participação como a atenção dos desenvolvedores. O interesse por esse tipo de questão é corroborado por Davies *et al.* (2006), que afirmam que o principal fator que influencia o uso contínuo das abordagens de modelagem é relativo as suas vantagens (desvantagens)/utilidade.

No contexto das discussões na plataforma *Software Engineering Stack Exchange*, o tema que tem atraído atenção e motivado a participação dos desenvolvedores versa sobre as razões da abordagem MDD não está sendo amplamente utilizada no processo de desenvolvimento

de software. Isso deixa explícito o interesse da comunidade em entender os fatores que estão influenciando negativamente a adoção da MDE pela indústria de software de modo mais efetivo.

5.5 AMEAÇAS À VALIDADE

Esta seção discute as ameaças à validade dos resultados desta pesquisa em relação a: (i) Validade da Conclusão; (ii) Validade Externa; (iii) Validade Interna e (iv) Validade de Construção.

Validade da Conclusão refere-se ao relacionamento entre o tratamento e o resultado do experimento (WOHLIN *et al.*, 2012). Com base nisso, todas as etapas apresentadas nesta pesquisa envolveram a participação de três pesquisadores: o autor do presente trabalho e dois professores doutores em Engenharia de Software. Deste modo, reuniões periódicas objetivando discutir e aperfeiçoar a metodologia empregada foram realizadas de modo a minimizar esta ameaça.

Validade Externa são as condições que limitam a habilidade de generalizar os resultados de um experimento (WOHLIN *et al.*, 2012). Neste caso, as análises feitas podem não representar toda a comunidade de desenvolvedores da MDE, tendo em vista que foram analisados somente dados gerados a partir de plataformas baseadas em perguntas e respostas. Para reduzir esse impacto, o estudo foi realizado utilizando as bases de dados do *Stack Overflow*, considerado a maior plataforma de perguntas e respostas da comunidade de desenvolvedores.

A plataforma *Software Engineering Stack Exchange* também foi utilizada de modo a agregar uma outra perspectiva sobre os resultados. Outro aspecto importante é o fato da mesma metodologia utilizada na plataforma *Stack Overflow* poder ser replicada na plataforma *Software Engineering Stack Exchange*.

Validade Interna refere-se à dimensão em que os resultados do estudo podem ser atribuídos aos tratamentos utilizados no próprio estudo (THOMAS; NELSON; SILVERMAN, 2009). Em outras palavras, o pesquisador deve tentar controlar os fatores ou as variáveis que possam influenciar o resultado. No contexto deste trabalho, os resultados obtidos através da análise dos dados poderiam sofrer interferência e provocar um viés gerado por parte do autor. Para minimizar o impacto desta problemática, foi desenvolvido uma metodologia rigorosa apoiada no conhecimento da literatura sobre o tema em estudo e fundamentada nas informações extraídas de ambas plataformas.

Validade de Construção é a generalização dos resultados do experimento para a teoria

por trás do experimento (WOHLIN *et al.*, 2012). Para minimizar esta ameaça, a base teórica foi adotada com base em métodos de outros estudos da literatura e na utilização de métricas de popularidade fornecidas pelas plataformas para avaliar uma discussão com base em diferentes perspectivas.

5.6 CONSIDERAÇÕES FINAIS

Neste capítulo foram apresentados e discutidos os resultados obtidos com base na análise das discussões das plataformas de perguntas e respostas *Stack Overflow* e *Software Engineering Stack Exchange* no âmbito da MDE. As discussões que serviram de base para essa análise estão disponíveis em um repositório do GitHub[13]. A Tabela 10 exibe o conjunto de dados utilizado neste trabalho.

Tabela 10 – Conjunto dos dados de cada plataforma

	Dataset	
	Stack Overflow	**Software Engineering Stack Exchange**
Discussões	74	9
Intervalo	ago/2008 a fev/2019	fev/2011 a mar/2016

Fonte**:** Elaborado pelo autor

A partir dos dados, é possível notar que os desenvolvedores possuem interesse na utilização das ferramentas de metamodelagem. Em seguida, buscam maiores informações relacionadas às transformações M2T ao invés das transformações M2M. Além disso, a comunidade dos desenvolvedores demonstram interesse na aplicação de restrições em seus modelos e como realizar integração entre ferramentas. Adicionalmente, questões envolvendo sugestões de ferramentas MDE costumam atrair à atenção dos usuários.

É importante mencionar que foram identificadas às ferramentas MDE nas quais o usuário utilizou as *tags* (Tabela 4) relacionadas aos conceitos de MDE. A partir disso, foram mapeadas as ocorrências entre cada ferramenta com base nas *tags* (Tabela 7) utilizadas pelos desenvolvedores. Além disso, os resultados mostraram que os usuários do *Stack Overflow* possuem interesses em ferramentas que divergem de outros desenvolvedores que utilizam uma outra plataforma. Vale ressaltar que uma análise mais detalhada deverá ser realizada de modo a entender o contexto de uso das ferramentas identificadas.

Por fim, é perceptível um baixo interesse por parte da comunidade de desenvolve-

[13] https://github.com/betomedeiros/stackoverflow-software-engineering-discussion

dores por assuntos que envolvam a abordagem MDE. Isto é corroborado pelo baixo número de discussões identificadas durante a condução deste trabalho, tendo em vista que o *Stack Overflow* possui aproximadamente 18 milhões de questões e o *Software Engineering Stack Exchange* possui aproximadamente 54.000 questões (STACKEXCHANGE, 2019).

6 CONCLUSÃO

Este trabalho apresentou os resultados obtidos a partir da análise de discussões entre desenvolvedores em plataformas de perguntas e respostas de Engenharia de Software no âmbito da Engenharia Dirigida por Modelos. Os dados utilizados foram obtidos a partir do *dataset* fornecido pelo *Stack Exchange Data Explorer*, que reúne os dados das plataformas *Stack Overflow* e *Software Engineering Stack Exchange*. Tais plataformas são providas no modelo de perguntas e respostas, servindo como uma base de conhecimento no âmbito da Engenharia de Software.

De modo a alcançar os objetivos deste trabalho de forma satisfatória, uma metodologia foi desenvolvida e aplicada em ambas plataformas. O processo que norteou esta pesquisa foi dividido em quatro fases: (i) mineração dos repositórios do *Stack Overflow* e *Software Engineering Stack Exchange*; (ii) classificação das discussões; (iii) identificação das principais ferramentas MDE; (iv) por fim, discussões sobre ferramentas MDE.

A partir dos dados analisados, é possível observar que há um grande interesse da comunidade pelas ferramentas de metamodelagem. Isso contrasta com resultados de outros estudos da literatura como, por exemplo, o estudo de Forward e Lethbridge (2008), que aponta que poucos desenvolvedores utilizavam linguagens de modelagem e linguagens de geração de código.

Conforme Hutchinson *et al.* (2011), em seu estudo empírico avaliando a MDE na indústria, o uso de MDE implica em uma carga de treinamento direcionado às ferramentas MDE. Corroborando esse fato, de acordo com Vetro, Bohm e Torchiano (2015), existe um esforço requerido para aplicação de técnicas e ferramentas de MDE, bem como a falta de competência e suporte ferramental. Com base nisso, os fóruns de discussões que abordam temas diversos no âmbito da Engenharia de Software, especificamente o *Stack Overflow* e o *Software Engineering Data Exchange*, podem ser mais um recurso visando uma melhor compreensão dos conceitos tanto teóricos como técnicos a respeito da MDE a partir das diversas discussões presentes nessas plataformas.

Além disso, os resultados direcionam para uma utilização mais baixa das ferramentas de transformações de modelos, fato este que é também identificado em outros estudos acadêmicos. Por exemplo, no estudo de Tomassetti *et al.* (2012), com base na análise dos resultados de um *survey*, os autores concluíram que poucos desenvolvedores utilizavam transformações entre modelos.

A presença de perguntas envolvendo aspectos teóricos da abordagem MDE valida que os conceitos ainda não estão totalmente elucidados na comunidade dos desenvolvedores. Alinhado a isso, Agner *et al.* (2013) afirmam que no contexto da indústria, poucos desenvolvedores conhecem e usam, parcialmente ou não, as abordagens dirigidas por modelo.

6.1 CONTRIBUIÇÕES

Com a pesquisa relatada nesta dissertação, as principais contribuições foram alcançadas:

- Categorização das discussões sobre MDE a partir das discussões no *Stack Overflow* e *Software Engineering Stack Exchange*;
- Evidências aos desenvolvedores, pesquisadores ou praticantes da abordagem MDE sobre quais temas da referida abordagem têm sido discutido nos fóruns sobre Engenharia de Software;
- Informações sobre quais temas da MDE têm atraído a atenção dos desenvolvedores nos fóruns de Engenharia de Software;
- Identificação dos domínios nos quais alguns desenvolvedores estão aplicando MDE;
- Apresentação das principais ferramentas MDE que estão sendo discutidas no *Stack Overflow*;
- Identificação das principais ferramentas que se destacaram em cada fase da MDE com base nas discussões do *Stack Overflow*;
- Criação de uma metodologia para identificar discussões relacionadas a MDE na base de dados do *Stack Overflow* e *Software Engineering Stack Exchange*.

6.2 LIMITAÇÕES

Apesar dos objetivos deste trabalho terem sido alcançados pode-se citar algumas limitações:

- Utilizar apenas dois repositórios de software baseados em perguntas e respostas. A análise de outros repositórios poderia agregar novas evidências sobre o tema em estudo;
- Foram utilizadas apenas duas métricas de popularidade: *ViewCount* e *AnswerCount*. Outras métricas como *Score* e *FavoriteCount* poderiam ter sido utilizadas;

6.3 TRABALHOS FUTUROS

Como trabalho futuro, pretende-se explorar novas bases de dados relacionadas ao tema de modo a ampliar e agregar novas perspectivas nas discussões em relação ao assunto estudado. Além disso, almeja-se realizar pesquisas a partir de outros trabalhos acadêmicos, confrontando a relação que estes possuem em consideração sobre o que se é discutido nos fóruns que abordam a temática MDE.

Por fim, espera-se também criar um *website* que congregue as principais dúvidas identificadas durante esta pesquisa, fornecendo respostas mais completas que apresentem, por exemplo, trechos de código e discussões mais aprofundadas.

REFERÊNCIAS

AGNER, L. T. W.; SOARES, I. W.; STADZISZ, P. C.; O, J. M. S. A brazilian survey on uml and model-driven practices for embedded software development. **J. Syst. Softw.,** New York, NY, USA, v. 86, n. 4, p. 997–1005, abr. 2013. Disponível em: <http://dx.doi.org/10.1016/j.jss.2012.11.023>. Acesso em: 5 jul. 2019.

ANTONIOL; CANFORA; CASAZZA; LUCIA, D. Information retrieval models for recove-ring traceability links between code and documentation. In: PROCEEDINGS INTERNATIONAL CONFERENCE ON SOFTWARE MAINTENANCE, 2., 2000. [S.l.], **Anais...** [S.l.:s.n.], 2000. p. 40–49.

ANVIK, J.; MURPHY, G. C. Determining implementation expertise from bug reports. In: PROCEEDINGS OF THE FOURTH INTERNATIONAL WORKSHOP ON MINING SOFTWARE REPOSITORIES, 2., 2007. Washington, **Anais...** Washington, DC, USA: IEEE Computer Society, 2007. (MSR '07), p. 2–. Disponível em: <https://doi.org/10.1109/MSR.2007.7>. Acesso em: 5 jul. 2019.

ATKINSON, C.; KUHNE, T. Model-driven development: a metamodeling foundation. **IEEE Software**, v. 20, n. 5, p. 36–41, set. 2003.

BAJAJ, K.; PATTABIRAMAN, K.; MESBAH, A. Mining questions asked by web developers. In: WORKING CONFERENCE ON MINING SOFTWARE REPOSITORIES, 11., 2014. New York, **Anais...** New York, NY, USA: ACM, 2014. (MSR 2014), p. 112–121. Disponível em: <http://doi.acm.org/10.1145/2597073.2597083>. Acesso em: 5 jul. 2019.

BALTES, S.; DUMANI, L.; TREUDE, C.; DIEHL, S. Sotorrent: Reconstructing and analyzing the evolution of stack overflow posts. In: INTERNATIONAL CONFERENCE ON MINING SOFTWARE REPOSITORIES, 15., 2018. New York, **Anais...** New York, ew York, NY, USA: ACM, 2018. p. 319–330. Disponível em: <http://doi.acm.org/10.1145/3196398.3196430>. Acesso em: 5 jul. 2019.

BANDEIRA, A.; MEDEIROS, C. A.; PAIXAO, M.; MAIA, P. H. We need to talk about microservices: An analysis from the discussions on stackoverflow. In: INTERNATIONAL CONFERENCE ON MINING SOFTWARE REPOSITORIES, 16., 2019. Piscataway, **Anais...** Piscataway, NJ, USA: IEEE Press, 2019. p. 255–259. Disponível em: <https://doi.org/10.1109/MSR.2019. 00051>. Acesso em: 5 jul. 2019.

BARUA, A.; THOMAS, S. W.; HASSAN, A. E. What are developers talking about? an analysis of topics and trends in stack overflow. **Empirical Software Engineering,** v. 19, n. 3, p. 619–654, Jun 2014. Disponível em: <https://doi.org/10.1007/s10664-012-9231-y>. Acesso em: 5 abr. 2019.

BEGEL, A.; BOSCH, J.; STOREY, M. Social networking meets software development: Perspectives from github, msdn, stack exchange, and topcoder. **IEEE Software,** v. 30, n. 1, p. 52–66, jan. 2013.

BETTENBURG, N.; PREMRAJ, R.; ZIMMERMANN, T.; KIM, S. Extracting structural in-formation from bug reports. In: INTERNATIONAL WORKING CONFERENCE ON MINING SOFTWARE REPOSITORIES, 1., 2008. New York, **Anais...** New York, NY, USA: ACM, 2008. p. 27–30. Disponível em: <http://doi.acm.org/10.1145/1370750.1370757>. Acesso em: 5 abr. 2019.

BIRD, C.; RIGBY, P. C.; BARR, E. T.; HAMILTON, D. J.; GERMAN, D. M.; DEVANBU, P. The promises and perils of mining git. In: INTERNATIONAL WORKING CONFERENCE ON MINING SOFTWARE REPOSITORIES, 2., 2009. [S.l.], **Anais...** [S.l.:s.n.], 2009. p. 1–10.

BLEI, D. M.; NG, A. Y.; JORDAN, M. I. Latent dirichlet allocation. **Journal of machine Learning research,** v. 3, n. 2, p. 993–1022, 2003.

BLUMBERG, R.; ATRE, S. The problem with unstructured data. **Powell Publishing Inc,** v. 13, n. 42-49, p. 62, 2003.

BOYD-GRABER, J.; MIMNO, D.; NEWMAN, D. Care and feeding of topic models: Pro-blems, diagnostics, and improvements. Handbook of mixed membership models and their applications. **CRC Press**, v. 225255, 2014.

BRAMBILLA, M.; CABOT, J.; WIMMER, M. Model-driven software engineering in practice. **Synthesis Lectures on Software Engineering, Morgan & Claypool Publishers,** v. 1, n. 1, p. 1–182, 2012.

BRUNELIERE, H.; CABOT, J.; JOUAULT, F. Combining Model-Driven Engineering and Cloud Computing. In: EUROPEAN CON-FERENCE ON MODELLING FOUNDATIONS AND APPLICATIONS, 4., 2010. Paris, **Anais...** Paris, France: [s.n.], 2010. Disponível em: <https://hal.archives-ouvertes.fr/hal-00539168>. Acesso em: 5 abr. 2019.

CANFORA, G.; CERULO, L.; PENTA, M. D. Identifying changed source code lines from version repositories. In: INTERNATIONAL WORKSHOP ON MINING SOFTWARE REPOSITORIES, 5., 2007. [s.l.]. **Anais**... [S.l.]: ICSE Workshops, 2007. p. 14–14.

CARLSSON, E. **Mining Git Repositories:** An introduction to repository mining. [S.l.:s.n.], 2013.

CETINKAYA, D.; VERBRAECK, A. Metamodeling and model transformations in modeling and simulation. In: WINTER SIMULATION CONFERENCE, 1., 2011. [S.l.]. **Anais...** [S.l.: s.n.], 2011. p. 3043–3053.

CHEN, T.-H.; THOMAS, S. W.; HASSAN, A. E. A survey on the use of topic models when mining software repositories. **Empirical Software Engineering,** v. 21, n. 5, p. 1843–1919, Oct 2016. Disponível em: <https://doi.org/10.1007/s10664-015-9402-8>. Acesso em: 5 abr. 2019.

CICCOZZI, F.; CRNKOVIC, I.; RUSCIO, D. D.; MALAVOLTA, I.; PELLICCIONE, P.; SPA-LAZZESE, R. Model-driven engineering for mission-critical iot systems. **IEEE Software**, v. 34, n. 1, p. 46–53, jan 2017.

CORLEY, C. S.; DAMEVSKI, K.; KRAFT, N. A. Changeset-based topic modeling of software repositories. **IEEE Transactions on Software Engineering,** p. 1–1, 2018.

CUADRADO, J. S.; IZQUIERDO, J. L. C.; MOLINA, J. G. Applying model-driven engineering in small software enterprises. **Science of Computer Programming,** v. 89, p. 176 – 198, 2014. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167642313001056>. Acesso em: 5 abr. 2019.

DAVIES, I.; GREEN, P.; ROSEMANN, M.; INDULSKA, M.; GALLO, S. How do practitioners use conceptual modeling in practice? **The Netherlands**, v. 58, n. 3, p. 358–380, set. 2006. Disponível em: <http://dx.doi.org/10.1016/j.datak.2005.07.007>. Acesso em: 5 abr. 2019.

ECLIPSE. **OCL Constraint Examples for UML (using Papyrus).** 2019. Disponí-vel em: <https://help.eclipse.org/oxygen/index.jsp?topic=\%2Forg.eclipse.ocl.doc\%2Fhelp\ %2FOCLExamplesforUML.html>.Acesso em: 8 abr. 2019.

ENGINEERING, S. **Welcome to Software Engineering Stack Exchange.** 2019. Disponível em: <https://softwareengineering.stackexchange.com/tour>. Acesso em: 5 abr. 2019.

FERRY, N.; SONG, H.; ROSSINI, A.; CHAUVEL, F.; SOLBERG, A. Cloudmf: Applying mde to tame the complexity of managing multi-cloud applications. In: INTERNATIONAL CONFERENCE ON UTILITY AND CLOUD COMPUTING, 7., 2014. [S.l.. **Anais**... [S.l.: s.n.], 2014. p. 269–277.

FORWARD, A.; BADREDDIN, O.; LETHBRIDGE, T. C. Perceptions of software modeling: a survey of software practitioners. In: WORKSHOP FROM CODE CENTRIC TO MODEL CENTRIC: EVALUATING THE EFFECTIVENESS OF MDD, 5., 2010. [S.l.]. **Anais...** [S.l.: s.n.], 2010.

FORWARD, A.; LETHBRIDGE, T. C. Problems and opportunities for model-centric versus code-centric software development: A survey of software professionals. In: INTERNATIONAL WORKSHOP ON MODELS IN SOFTWARE ENGINEERING, 1., 2008. New York, **Anais...** New York, NY, USA: ACM, 2008. p. 27–32. Disponível em: <http://doi.acm.org/10.1145/1370731.1370738>. Acesso em: 5 abr. 2019.

FRANCE, R.; RUMPE, B. Model-driven development of complex software: A research roadmap. In: FUTURE OF SOFTWARE ENGINEERIN, 7., 2007. [S.l.]. **Anais...** [S.l.: s.n.], 2007. p. 37–54.

FREITAS, F. J. G. d. **JustModeling:** Uma abordagem MDE para desenvolvimento de aplicações negociais para Android. Dissertação (Mestrado em Ciência da Computação) — Universidade Estadual do Ceará, Fortaleza, 2016.

GANSSLE, J. **A trillion lines of code?** [S.l.:s.n.], 2008.

GIERE, R. N. **How models are used to represent physical reality**. [S.l.:s.n.], 2002.

GODFREY, M. W.; HASSAN, A. E.; HERBSLEB, J.; MURPHY, G. C.; ROBILLARD, M.; DEVANBU, P.; MOCKUS, A.; PERRY, D. E.; NOTKIN, D. Future of mining software archives: A roundtable. **IEEE Software,** v. 26, n. 1, p. 67–70, Jan 2009.

GRIMES, S. Unstructured data and the 80 percent rule. **Carabridge Bridgepoints,** p. 10, 2008.

GUTTMAN, M.; PARODI, J. **Real-Life MDA:** Solving Business Problems with Model Dri ven Architecture. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2007.

HASSAN, A. E. The road ahead for mining software repositories. In: FRONTIERS OF SOFTWARE MAINTENANCE, 1., 2008. [S.l.] **Anais...** [S.l.: s.n.], 2008. p. 48–57.

HEITKOTTER, H.; MAJCHRZAK, T. A.; KUCHEN, H. Cross-platform model-driven develop-ment of mobile applications with md2. In: ANNUAL ACM SYMPOSIUM ON APPLIED COMPUTING, 28., 2013. New York, **Anais...** New York, NY, USA: ACM, 2013. (SAC '13), p. 526–533. Disponível em: <http://doi.acm.org/10.1145/2480362.2480464>. Acesso em: 9 maio 2019.

HUTCHINSON, J.; WHITTLE, J.; ROUNCEFIELD, M. Model-driven engineering practices in industry: Social, organizational and managerial factors that lead to success or failure. **Science of Computer Programming,** v. 89, p. 144 – 161, 2014.. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167642313000786>. Acesso em: 9 maio 2019.

HUTCHINSON, J.; WHITTLE, J.; ROUNCEFIELD, M.; KRISTOFFERSEN, S. Empirical assessment of mde in industry. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 3., 2011. New York, **Anais...** New York, NY, USA: ACM, 2011. (ICSE '11), p. 471–480. Disponível em: <http://doi.acm.org/10.1145/1985793.1985858>. Acesso em: 9 maio 2019.

JOUAULT, F.; BÉZIVIN, J.; KURTEV, I. Tcs:: A dsl for the specification of textual concrete syntaxes in model engineering. In: INTERNATIONAL CONFERENCE ON GENERATIVE PROGRAMMING AND COMPONENT ENGINEERING, 5., 2006. New York, **Anais...** New York, NY, USA: ACM, 2006.

KAGDI, H.; COLLARD, M. L.; MALETIC, J. I. A survey and taxonomy of approaches for mining software repositories in the context of software evolution. **J. Softw. Maint. Evol., John Wiley & Sons, Inc.,** New York, NY, USA, v. 19, n. 2, p. 77–131, mar. 2007. Disponível em: <http://dx.doi.org/10.1002/smr.344>. Acesso em: 9 maio 2019.

KAHANI, N.; BAGHERZADEH, M.; DINGEL, J.; CORDY, J. R. The problems with eclipse modeling tools: A topic analysis of eclipse forums. In: INTERNATIONAL CONFERENCE ON MODEL DRIVEN ENGINEERING LANGUAGES AND SYSTEMS, 19., 2016. New York, **Anais...** New York, NY, USA: ACM, 2016. p. 227–237. Disponível em: <http://doi.acm.org/10.1145/2976767.2976773>. Acesso em: 9 maio 2019.

KIM, S.; ERNST, M. D. Prioritizing warning categories by analyzing software history. In: INTERNATIONAL WORKSHOP ON MINING SOFTWARE REPOSITORIES, 5., 2007. [S.l.]. **Anais...** [S.l.: s.n.], 2007. p. 27–27.

KLEPPE, A. G.; WARMER, J.; BAST, W. **MDA Explained:** The Model Driven Architecture: Practice and Promise. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2003.

KOCHHAR, P. S. Mining testing questions on stack overflow. In: INTERNATIONAL WORKSHOP ON SOFTWARE MINING, 5., 2016. New York, **Anais...** New York, NY, USA: ACM, 2016. p. 32–38. Disponível em: <http://doi.acm.org/10.1145/2975961.2975966>. Acesso em: 9 maio 2019.

KRAMER, J. Is abstraction the key to computing? Commun. **ACM,** New York, v. 50, n. 4, p. 36–42, abr. 2007. Disponível em: <http://doi.acm.org/10.1145/1232743.1232745>. Acesso em: 9 maio 2019.

KRIOUILE, A.; ADDAMSSIRI, N.; GADI, T.; BALOUKI, Y. Getting the static model of pim from the cim. In: IEEE INTERNATIONAL COLLOQUIUM IN INFORMATION SCIENCE AND TECHNOLOGY, 3., 2014. [S.l.]. **Anais...** [S.l.: s.n.], 2014. p. 168–173.

KUHN, A.; DUCASSE, S.; GîRBA, T. Semantic clustering: Identifying topics in source code. **Information and Software Technology,** v. 49, n. 3, p. 230 – 243, 2007.. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0950584906001820>. Acesso em: 9 maio 2019.

KüHNE, T. Matters of (meta-) modeling. **Software & Systems Modeling,** v. 5, n. 4, p. 369–385, 2006. Disponível em: <https://doi.org/10.1007/s10270-006-0017-9>. Acesso em: 9 maio 2019.

LANDAUER, T. K.; MCNAMARA, D. S.; DENNIS, S.; KINTSCH, W. **Handbook of latent semantic analysis.** [S.l.]: Psychology Press, 2013.

LAWRIE, D.; MORRELL, C.; FEILD, H.; BINKLEY, D. What's in a name? a study of identifiers. In: IEEE INTERNATIONAL CONFERENCE ON PROGRAM COMPREHENSION, 14., 2006. [S.l.]. **Anais...** [S.l.: s.n.], 2006. p. 3–12.

LAWRIE, D.; MORRELL, C.; FEILD, H.; BINKLEY, D. Effective identifier names for com-prehension and memory. **Innovations in Systems and Software Engineering,** v. 3, n. 4, p. 303–318, Dec 2007. Disponível em: <https://doi.org/10.1007/s11334-007-0031-2>. Acesso em: 9 maio 2019.

LI, H.; CHEN, T.-H. P.; SHANG, W.; HASSAN, A. E. Studying software logging using topic models. **Empirical Softw. Engg., Kluwer Academic Publishers,** Norwell, MA, USA, v. 23, n. 5, p. 2655–2694, out. 2018. Disponível em: <https://doi.org/10.1007/s10664-018-9595-8>. Acesso em: 9 maio 2019.

LINARES-VáSQUEZ, M.; DIT, B.; POSHYVANYK, D. An exploratory analysis of mobile development issues using stack overflow. In: WORKING CONFERENCE ON MINING SOFTWARE REPOSITORIES, 10., 2013. [S.l.]. **Anais...** [S.l.: s.n.], 2013. p. 93–96.

LINSTEAD, E.; BALDI, P. Mining the coherence of gnome bug reports with statistical topic models. In: IEEE INTERNATIONAL WORKING CONFERENCE ON MINING SOFTWARE REPOSI-TORIES, 6., 2009. [S.l.]. **Anais....** [S.l.: s.n.], 2009. p. 99–102.

LOPEZ, T.; TUN, T.; BANDARA, A.; LEVINE, M.; NUSEIBEH, B.; SHARP, H. An anatomy of security conversations in stack overflow. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING: SOFTWARE ENGINEERING IN SOCIETY, 41., 2019. Piscataway, **Anais...** Piscataway, NJ, USA: IEEE Press, 2019. (ICSE-SEIS ’10), p. 31–40. Disponível em: <https://doi.org/11.1109/ICSE-SEIS. 2019.00012>. Acesso em: 9 maio 2019.

MA, Z.; HE, X. A model-driven approach for model transformations. In: COMPUTING CONFERENCE, 1., 2016. [S.l.]. **Anais...** [S.l.: s.n.], 2016. p. 1199–1205.

MADANI, N.; GUERROUJ, L.; PENTA, M. D.; GUEHENEUC, Y.; ANTONIOL, G. Recog-nizing words from source code identifiers using speech recognition techniques. In: EUROPEAN CONFERENCE ON SOFTWARE MAINTENANCE AND REENGINEERING, 14., 2010. [S.l.]. **Anais...** [S.l.: s.n.], 2010. p. 68–77.

MAIA, P.; GADELHA, F.; BORGES, M.; MUNIZ, L. L.; SILVA, A. S. D.; DE, J. N. Práticas e experiências no ensino de engenharia dirigida por modelos**. iSys - Revista Brasileira de Sistemas de Informação,** v. 9, p. 106–135, 08 2016.

MALETIC, J. I.; MARCUS, A. Using latent semantic analysis to identify similarities in source code to support program understanding. In: IEEE INTERNATIONALS CONFE-RENCE ON TOOLS WITH ARTIFICIAL INTELLIGENCE, 15., 2000. [S.l.]. **Anais...** [S.l.: s.n.], 2000. p. 46–53.

MAMYKINA, L.; MANOIM, B.; MITTAL, M.; HRIPCSAK, G.; HARTMANN, B. Design lessons from the fastest q&a site in the west. In: CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 1., 2011. New York, **Anais...** New York, NY, USA: ACM, 2011. (CHI ’11), p. 2857–2866. Disponível em: <http://doi.acm.org/10.1145/1978942.1979366>. Acesso em: 9 maio 2019.

MANDELIN, D.; XU, L.; BODÍK, R.; KIMELMAN, D. Jungloid mining: Helping to navigate the api jungle. **SIGPLAN Not.,** ACM, New York, NY, USA, v. 40, n. 6, p. 48–61, jun. 2005. Disponível em: <http://doi.acm.org/10.1145/1064978.1065018>. Acesso em: 9 maio 2019.

MARTíNEZ, Y.; CACHERO, C.; MELIá, S. Mdd vs. traditional software development: A practitioner's subjective perspective. **Information and Software Technology,** v. 55, n. 2, p. 189– 200, 2013. Special Section: Component-Based Software Engineering (CBSE), 2011. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0950584912001309>. Acesso em: 9 maio 2019.

MICHAIL, A.; XIE, T. Helping users avoid bugs in gui applications. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 27., 2005. [S.l.]. **Anais...** [S.l.: s.n.], 2005. p. 107–116.

MOHAGHEGHI, P.; DEHLEN, V. Where is the proof? - a review of experiences from applying mde in industry. In: SCHIEFERDECKER, I.; HARTMAN, A. (Ed.). **Model Driven Architec-ture:** Foundations and Applications. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. p. 432–443.

MOODY, D. The "physics" of notations: Toward a scientific basis for constructing visual notations in software engineering. **IEEE Transactions on Software Engineering,** v. 35, n. 6, p. 756–779, Nov 2009.

MUSSBACHER, G.; AMYOT, D.; BREU, R.; BRUEL, J.-M.; CHENG, B. H. C.; COLLET, P.; COMBEMALE, B.; FRANCE, R. B.; HELDAL, R.; HILL, J.; KIENZLE, J.; SCHÖTTLE, M.; STEIMANN, F.; STIKKOLORUM, D.; WHITTLE JON", e. J.; SCHULTE, W.; RAMOS, I.; ABRAHÃO, S.; INSFRAN, E. The relevance of model-driven engineering thirty years from now. In: _____. **Model-Driven Engineering Languages and Systems**. Cham: Springer International Publishing, 2014. p. 183–200.

NIKIFOROVA, O.; CERNICKINS, A.; PAVLOVA, N. Discussing the difference between model driven architecture and model driven development in the context of supporting tools. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING ADVANCES, 5., 2009. [S.l.]. **Anais...** [S.l.: s.n.], 2009. p. 446–451.

PONZANELLI, L.; BACCHELLI, A.; LANZA, M. Leveraging crowd knowledge for software comprehen-sion and development. In: EUROPEAN CONFERENCE ON SOFTWARE MAINTENANCE AND REENGINEERING, 17., 2013. [S.l.]. **Anais...** [S.l.: s.n.], 2013. p. 57–66.

POSNETT, D.; WARBURG, E.; DEVANBU, P.; FILKOV, V. Mining stack exchange: Expertise is evident from initial contributions. In: INTERNATIONAL CONFERENCE ON SOCIAL INFORMATICS, 12., 2012. [S.l.]. **Anais...** Washington, DC, USA: IEEE Computer Society, 2012. (SOCIALINFOR- MATICS '12), p. 199–204. Disponível em: <https://doi.org/10.1109/SocialInformatics.2012>. Acesso em: 9 maio 2019.

QUINTERO, J. B.; ANAYA, R. MDA y el papel de los modelos en el proceso de desarrollo de software. **Revista EIA, scieloco**, p. 131 – 146, 12 2007. Disponível

em: <http://www.scielo.org.co/scielo.php?script=sci_arttext&pid= S1794-12372007000200011&nrm=iso>. Acesso em: 9 maio 2019.

RADEMACHER, F.; SORGALLA, J.; SACHWEH, S. Challenges of domain-driven microser-vice design: A model-driven perspective. **IEEE Software**, v. 35, n. 3, p. 36–43, May 2018.

RAHMAN, M. Mitigating information disclosure attacks in the cloud by blocking invalid user and figure out problems to solve ddos by analyzing stackoverflow questions. In: INTERNATIONAL CONFERENCE ON ELECTRICAL ELECTRONIC ENGINEERING, 2., 2017. [S.l.]. **Anais...** [S.l.: s.n.], 2017. p. 1–4.

RIGBY, P. C.; HASSAN, A. E. What can oss mailing lists tell us? a preliminary psychometric text analysis of the apache developer mailing list. In: INTERNATIONAL WORKSHOP ON MINING SOFTWARE REPOSITORIES, 5., 2007. [S.l.]. **Anais...** [S.l.: s.n.], 2007. p. 23–23.

SAMPAIO, A.; RASHID, A. Mining early aspects from requirements with ea-miner. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 30., 2008. New York, **Anais...** New York, NY, USA: ACM, 2008. p. 911–912. Disponível em: <http://doi.acm.org/10.1145/ 1370175.1370183>. Acesso em: 9 maio 2019.

SAYÃO, L. F. Modelos teóricos em ciência da informação–abstração e método científico. **Ciência da informação,** v. 30, n. 1, 2001.

SEIDEWITZ, E. What models mean. **IEEE Software,** v. 20, n. 5, p. 26–32, set. 2003.

SELIC, B. The pragmatics of model-driven development. **IEEE Software,** v. 20, n. 5, p. 19–25, set. 2003.

SERRANO, N.; CIORDIA, I. Bugzilla, itracker, and other bug trackers. **IEEE Software,** v. 22, n. 2, p. 11–13, March 2005.

SHIHAB, E.; BETTENBURG, N.; ADAMS, B.; HASSAN, A. E. On the central role of mailing lists in open source projects: An exploratory study. In: NAKAKOJI, K.; MURAKAMI, Y.; MCCREADY, E. (Ed.). **New Frontiers in Artificial Intelligence.** Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 91–103.

SHIRANI, A.; XU, B.; LO, D.; SOLORIO, T.; ALIPOUR, A. Question relatedness on stack overflow: The task, dataset, and corpus-inspiredmodels. **arXiv preprint,** v. 2, , 2019.

SILVA, A. R. da. Model-driven engineering: A survey supported by the unified conceptual model. **Computer Languages, Systems Structures,** v. 43, p. 139 – 155, 2015. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1477842415000408>. Acesso em: 9 maio 2019.

SILVA, M. A. G.; BARBOSA, E. F.; MALDONADO, J. C. Model-driven development of learning objects. In: FRONTIERS IN EDUCATION CONFERENCE, 2., 2011. [S.l.]. **Anais...** [S.l.: s.n.], 2011. p. F4E–1–F4E–6.

STACKEXCHANGE. **All sites stackexchange.** 2019. Disponível em: <https://stackexchange.com/sites?view=list#traffic>. Acesso em: 9 maio 2019.

SUN, X.; LIU, X.; HU, J.; ZHU, J. Empirical studies on the nlp techniques for source code data preprocessing. In: INTERNATIONAL WORKSHOP ON EVIDENTIAL ASSESSMENT OF SOFTWARE TECHNOLOGIES, 3., 2014. New York, **Anais...** New York, NY, USA: ACM, 2014. p. 32–39. Disponível em: <http://doi.acm.org/10.1145/2627508.2627514>. Acesso em: 9 maio 2019.

THOMAS, J. R.; NELSON, J. K.; SILVERMAN, S. J. **Métodos de pesquisa em atividade física**. [S.l.]: Artmed, 2009.

THOMAS, S. W. Mining software repositories using topic models. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 3., 2011. [S.l.] **Anais...** New York, NY, USA: ACM, 2011. p. 1138–1139. Disponível em: <http://doi.acm.org/10.1145/1985793.1986020>. Acesso em: 9 maio 2019.

THOMAS, S. W.; HASSAN, A. E.; BLOSTEIN, D. Mining Unstructured Software Repositories. In: MENS, T.; SEREBRENIK, A.; CLEVE, A. (Ed.). **Evolving Software Systems**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014. p. 139–162.

THRAMBOULIDIS, K.; VACHTSEVANOU, D. C.; SOLANOS, A. Cyber-physical microser-vices: An iot-based framework for manufacturing systems. In: IEEE INDUSTRIAL CYBER-PHYSICAL SYSTEMS, 2., 2018. [S.l.]. **Anais...** [S.l.: s.n.], 2018. p. 232–239.

TICHY, W. An interview with prof. andreas zeller: Mining your way to software reliability. **Ubiquity, ACM,** New York, NY, USA, v. 20, nov. 2010. Disponível em: <http://doi.acm.org/10.1145/1880066.1883621>. Acesso em: 9 maio 2019.

TOMASSETTI, F.; Torchiano, M.; Tiso, A.; Ricca, F.; Reggio, G. Maturity of software mo-delling and model driven engineering: A survey in the italian industry. In: INTERNATIONAL CONFERENCE ON EVALUATION ASSESSMENT IN SOFTWARE ENGINEERING, 16., 2012. [S.l.]. **Anais...** [S.l.: s.n.], 2012. p. 91–100.

TORCHIANO, M.; TOMASSETTI, F.; RICCA, F.; TISO, A.; Reggio, G. Preliminary findings from a survey on the md state of the practice. In: INTERNATIONAL SYMPOSIUM ON EMPIRICAL SOFTWARE ENGINEERING AND MEASUREMENT, 1., 2011. [S.l.]. **Anais...** [S.l.: s.n.], 2011. p. 372–375.

TORCHIANO, M.; TOMASSETTI, F.; RICCA, F.; TISO, A.; REGGIO, G. Relevance, benefits, and problems of software modelling and model driven techniques-a survey in the italian industry. **J. Syst. Softw.,** New York, NY, USA, v. 86, n. 8, p. 2110–2126,

ago. 2013. Disponível em: <http://dx.doi.org/10.1016/j.jss.2013.03.084>. Acesso em: 9 maio 2019.

TREUDE, C.; BARZILAY, O.; STOREY, M. How do programmers ask and answer questions on the web?: Nier track. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 33., 2011. [S.l.]. **Anais...** [S.l.: s.n.], 2011. p. 804–807.

VALLECILLO, A. On the industrial adoption of model driven engineering. is your company ready for mde? **International Journal of Information Systems and Software Engineering for Big Companies (IJISEBC),** v. 1, n. 1, p. 52–68, 2015.

VERMA, A.; SARDANA, N.; LAL, S. Comprehensive analysis of trends in software engineering q a site. In: INTERNATIONAL CONFERENCE ON CLOUD COMPUTING, DATA SCIENCE ENGINEERING, 9., 2009. [S.l.]. **Anais...** [S.l.: s.n.], 2019. p. 648–653.

VETRO, A.; BOHM, W.; TORCHIANO, M. On the benefits and barriers when adopting soft-ware modelling and model driven techniques - an external, differentiated replication. In: IEEE INTERNATIONAL SYMPOSIUM ON EMPIRICAL SOFTWARE ENGINEERING AND MEASURE-MENT, 1., 2015. [S.l.]. **Anais...** [S.l.: s.n.], 2015. p. 1–4.

VILLANES, I. K.; ASCATE, S. M.; GOMES, J.; DIAS-NETO, A. C. What are software engineers asking about android testing on stack overflow? In: BRAZILIAN SYMPOSIUM ON SOFTWARE ENGINEERING, 31., 2017. New York, **Anais...** New York, NY, USA: ACM, 2017. (SBES'17), p. 104– 113. Disponível em: <http://doi.acm.org/10.1145/3131151.3131157>. Acesso em: 9 maio 2019.

VöLTER, M. Md* best practices. **Journal of Object Technology**, v. 6, p. 79–102, 2009.

WHITTLE, J.; HUTCHINSON, J.; ROUNCEFIELD, M. The state of practice in model-driven engineering. **IEEE Software,** v. 31, n. 3, p. 79–85, May 2014.

WOHLIN, C.; RUNESON, P.; HÖST, M.; OHLSSON, M. C.; REGNELL, B.; WESSLÉN, A. **Experimentation in software engineering.** [S.l.]: Springer Science & Business Media, 2012.

XIE, T.; PEI, J. Mapo: Mining api usages from open source repositories. In: INTERNATIONAL WORKSHOP ON MINING SOFTWARE REPOSITORIES, 2., 2006. New York, **Anais...** New York, NY, USA: ACM,2006. , p. 54–57. Disponível em: <http://doi.acm.org/10.1145/1137983.1137997>. Acesso em: 9 maio 2019.

ZHANG, T.; UPADHYAYA, G.; REINHARDT, A.; RAJAN, H.; KIM, M. Are code examples on an online q a forum reliable?: A study of api misuse on stack overflow. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 40., 2018. [S.l.]. **Anais...** [S.l.: s.n.], 2018. p. 886–896.

ZIMMERMANN, T.; ZELLER, A.; WEISSGERBER, P.; DIEHL, S. Mining version histories to guide software changes. **IEEE Transactions on Software Engineering,** v. 31, n. 6, p. 429–445, 2005.

ZOU, J.; XU, L.; GUO, W.; YAN, M.; YANG, D.; ZHANG, X. Which non-functional require-ments do developers focus on? an empirical study on stack overflow using topic analysis. In: WORKING CONFERENCE ON MINING SOFTWARE REPOSITORIES, 12., 2015. [S.l.]. **Anais...** [S.l.: s.n.], 2015. p. 446–449.