UNIVERSIDADE FEDERAL DO PAMPA

Rafael dos Santos Torres

Recomendação de Assets Híbridos de Software Utilizando Técnicas de Aprendizado de Máquina

Alegrete
2021

Rafael dos Santos Torres

Recomendação de Assets Híbridos de Software Utilizando Técnicas de Aprendizado de Máquina

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Software da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Engenharia de Software.

Orientador: Prof. Dr. Fábio Paulo Basso

Alegrete
2021

Rafael dos Santos Torres

Recomendação de Assets Híbridos de Software Utilizando Técnicas de Aprendizado de Máquina

Trabalho de Conclusão de Curso apresentada ao Curso de Engenharia de Software da Universidade Federal do Pampa, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Trabalho de Conclusão de Curso defendido e aprovado em: 29 de setembro de 2021.

Banca examinadora:

_____

Prof. Dr. Fábio Paulo Basso

Orientador

UniPampa

_____

Prof. Dr. Elder de Macedo Rodrigues

UniPampa

_____

Prof. Dr. Marcelo Resende Thielo

UniPampa

Prof. Dr. Maicon Bernardino da Silveira

UniPampa

Assinado eletronicamente por **FABIO PAULO BASSO**, **PROFESSOR DO MAGISTERIO SUPERIOR**, em 29/09/2021, às 20:20, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

Assinado eletronicamente por **MAICON BERNARDINO DA SILVEIRA**, **PROFESSOR DO MAGISTERIO SUPERIOR**, em 29/09/2021, às 20:21, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

Assinado eletronicamente por **ELDER DE MACEDO RODRIGUES**, **PROFESSOR DO MAGISTERIO SUPERIOR**, em 29/09/2021, às 20:25, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

Assinado eletronicamente por **MARCELO RESENDE THIELO**, **PROFESSOR DO MAGISTERIO SUPERIOR**, em 29/09/2021, às 20:25, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

A autenticidade deste documento pode ser conferida no site https://sei.unipampa.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0622385** e o código CRC **4D7CB2FB**.

RESUMO

O reúso de software é uma alternativa efetiva para criar software de qualidade e suprir as demandas que a área de TI enfrenta, já que permite criar novos produtos de software a partir de software já existente, evitando o esforço, os custos, e os problemas originados da criação de um produto completamente novo. Graças à vasta disponibilidade de recursos disponíveis para desenvolvedores e empresas, o reúso de software se torna mais promissor, e ao mesmo tempo, mais desafiador. A fim de tornar o reúso uma abordagem mais efetiva e menos dependente do esforço humano para encontrar os *assets* corretos, as técnicas da área de aprendizado de máquina foram acrescentadas ao processo de reúso. Portanto, este trabalho relata três experimentos: dois deles executados buscando demonstrar como duas técnicas identificadas através de um mapeamento sistemático de literatura podem recomendar *assets* representados pela linguagem RAS++; o terceiro experimento compara as duas técnicas utilizadas baseando-se em duas métricas também identificadas através do mapeamento conduzido anteriormente. Os resultados acabaram sendo inconclusivos, sendo eles discutidos posteriormente junto aos fatores que levaram a tal conclusão.

Palavras-chave: Aprendizado de Máquina. DSL. *Asset*. Reúso Oportunista.

ABSTRACT

Software reuse is an effective alternative in creating software with quality. It also supports some of the Information Technology (IT) demands when creating new products from existing software, softening the effort and costs when compared to products developed from the scratch. Thanks to the vast availability of resources available to developers and to software houses, software reuse becomes more promising, yet more challenging. In order to make opportunistic reuse a more effective approach, and less dependent on human effort to find the right assets, machine learning techniques have been added to opportunistic reuse processes. This study summarizes a systematic mapping, characterizing how the machine learning area has supported the reuse of hybrid software assets. Therefore, this work reports three experiments: two of them carried out seeking to demonstrate how two techniques identified through a systematic literature mapping can recommend assets represented by the RAS++ language; the third experiment compares the two techniques used based on two metrics also identified through the mapping conducted previously. The results turned out to be inconclusive, and they are discussed later along with the factors that led to such a conclusion.

Key-words: Machine Learning. DSL. Asset. Oportunistic Reuse.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

DSL *Domain Specific Language*

MDE *Model-Driven Engineering*

SE *Software Engineering*

SUMÁRIO

1 INTRODUÇÃO

A Engenharia de Software tem como objetivo principal apoiar o desenvolvimento profissional de software (SOMMERVILLE, 2011), a fim de evitar uma nova crise no setor, como a que motivou o surgimento da área. Para tal, são propostos novos métodos e técnicas que deem suporte aos desenvolvedores e empresas, não apenas para criar software de qualidade, mas também para suprir a demanda crescente que a área enfrenta. Por esse motivo, o reúso se mostra uma alternativa sólida para lidar com os problemas citados, já que permite criar novos produtos de software a partir de software já existente (SAMETINGER, 1997), evitando o esforço, os custos, e os problemas originados da criação de um produto do zero.

Atingir o objetivo de reusar com sucesso artefatos de software demanda planejamento e, dependendo da forma que é praticado o reúso, exige certo esforço por parte dos mantenedores dos recursos que serão reutilizados. Para que esses esforços sejam minimizados e a eficiência do reuso seja intensificada, é necessário o suporte de técnicas/ferramentas para encontrar os artefatos corretos levando em consideração o problema a ser resolvido pelo desenvolvedor.

A área de *Data Mining* oferece os recursos necessários para dar suporte às necessidades da Engenharia de Software, permitindo a possibilidade de automação e otimização de processos essenciais para o processo de reúso, como busca e seleção dos artefatos corretos. Entretanto, a diversidade de métodos e tecnologias na área de *Data Mining* abre um leque para diversas abordagens que ainda devem ser exploradas.

1.1 Motivação

A MDE (*Model-Driven Engineering*) agrupa um conjunto de ativos dedicados à reutilização. Ela alcançou certa maturidade na prática e na pesquisa (MOHAGHEGHI et al., 2013), levando as pesquisas de Engenharia de Software ao desenvolvimento de vários tipos de *assets* além de modelos, como ferramentas, *Domain Specific Language* (DSL) e metodologias para necessidades específicas. Os trabalhos atuais são mais como uma orquestração de vários sistemas para *Model-Driven Engineering* (MDE) (LIEBEL et al., 2014) do que o uso de um suporte de ferramenta exclusivo, como observado no passado. Essa complexidade requer a execução de etapas de reutilização para pesquisar e integrar esses *assets* em ambientes de desenvolvimento. Essa mudança de complexidade dos processos de reutilização é observada nas primeiras contribuições para MDE em 2008 (STARY, 2000; BECKER; HOLTZ; PEREIRA, 2002; STOCQ; VANDERDONCKT, 2004; SOUZA; FALBO; GUIZZARDI, 2007; ANDERSSON; HST, 2008), em que um modelo é projetado com uma linguagem de design exclusiva e transformado por um único *script* para transformação de modelo. Nas abordagens atuais, os processos baseados em MDE orquestram vários *assets* com algumas tarefas de Engenharia de Software (semi-)automatizadas que associam sistemas para MDE (BASSO et al., 2013; BATORY; LATIMER; AZANZA,

2013; HEBIG; BENDRAOU, 2014; VARA et al., 2014).

A importância dessas abordagens para o desenvolvimento de software foi discutida recentemente por Fuggetta (FUGGETTA; NITTO, 2014), como a necessidade de investigar conceitos de computação em nuvem para a pesquisa de Processos de Software, destacando uma tendência para sistemas cooperativos que auxiliam nas tarefas de *Software Engineering* (SE) em diversas fases de desenvolvimento de *software*. Os autores concordaram que MDE é um paradigma importante para realizar essa tendência. Em 2006, Boehm também motivou a necessidade de pesquisa nesse sentido, argumentando que para competir, adaptar e sobreviver, as empresas de desenvolvimento de *software* dependerão da capacidade de integrar alguns *assets* para tarefas de engenharia de *software* (por exemplo, uma ferramenta de design e *scripts* sequenciais para geração de código) em cenários globais de reutilização feitos de *Systems of Systems* (SOS) (BOEHM, 2006).

Embora muito esforço tenha sido dedicado a essas pesquisas, oito anos depois, Fuggetta afirma que ainda é difícil tornar essa visão do futuro uma realidade na indústria (FUGGETTA; NITTO, 2014), deixando assim espaço para melhorias substanciais em termos de recomendação de *assets* (BASSO; WERNER; OLIVEIRA, 2017a).

Dessa forma, essa tendência de pesquisa para recomendação de *assets* é de especial interesse para o contexto de MDE. De acordo com essa tendência, o estado da prática de MDE (LIEBEL et al., 2014) relatou a necessidade de incluir heterogêneos (LAFI; HAMMOUDI; FEKI, 2011) e, ao mesmo tempo, sistemas colaborativos (ROCCO et al., 2016). Isso levanta uma lacuna de pesquisa para investigar como a pesquisa está adotando técnicas de mineração de dados para recomendação de *assets*.

1.2 Objetivos

O objetivo geral deste trabalho é "Aplicar técnicas de *Data Mining* para recomendação de *assets* de MDE", e assim apoiar o processo decisório de uma abordagem de aquisição de software por meio de reúso oportunista.

Objetivo específico 1: Caracterizar *assets* de MDE como recursos intensivos em conhecimento. Assim, de modo à entender o contexto de pesquisa para o TCC I, primeiramente se realizou a leitura da tese do orientador de TCC (BASSO, 2017). Em seguida, deu-se seguimento num estudo sobre a representação de recursos intensivos em conhecimento, concretizando o conhecimento inicial necessário sobre *assets* representados com RAS++ DSL.

Objetivo específico 2: Mapear as propostas da literatura da área. Assim, tratou-se de executar um mapeamento sistemático de técnicas de mineração de dados em repositórios de *assets*. Desta, um segundo artigo será escrito para o TCC 2.

Objetivo específico 3: Comparar duas técnicas de mineração de *assets*. Busca-se realizar um experimento que compare as duas melhores técnicas para a mineração de *assets* de MDE. Para tal, planejou-se a seleção de duas técnicas com base no mapeamento

conduzido na atividade anterior. Estas serão integradas no suporte ferramental disponível para o RAS++;

Objetivo específico 4: Preparar o suporte ferramental de recomendação para a integração com suporte ferramental já disponível no grupo de pequisa.

1.3 Resultados Esperados

1. Um aprofundamento dos conhecimento técnico-científicos no tema de mineração e recomendação de dados. Tal resultado é importante para o pesquisador, uma vez que visa suprir uma expectativa particular de desempenhar um perfil profissional em *data science* no futuro;

2. Fundamentação de uma pesquisa que busca a transferência automática de tecnologias de MDE. O estágio de tal pesquisa era, antes do presente trabalho, limitado para representação e transformação de *assets* descrevendo tecnologias de MDE (BASSO, 2017). Um resultado esperado pelo grupo de pesquisa é o aprofundamento da temática para um problema preliminar de levantamento e recomendação de *assets*, característica esta de pesquisas em aquisição de software e reúso oportunista;

3. Um suporte ferramental que contenha elementos centrais para apoiar o processo decisório na aquisição de software por meio da mineração e recomendação de *assets* que descrevem tecnologias de MDE;

4. Colaboração com colegas na execução de atividades de pesquisa em temas correlatos e complementares;

5. Publicação de artigos científicos com os resultados deste trabalho, bem como de pesquisas conduzidas no tema pelo grupo de pesquisa.

1.4 Contribuições

As contribuições exclusivamente derivadas deste TCC podem ser resumidas em:

1. Um estudo de mapeamento sistemático;

2. Um webcrawler em fase inicial para popular um repositório de *assets*, que pode ser continuado e expandido para outras futuras bases.

3. Um experimento comparando duas técnicas utilizáveis em um contexto de recomendação.

4. Um artigo publicado no ERES[1].

[1] <https://sol.sbc.org.br/index.php/eres/article/view/13713>.

1.5 Organização

- O Capítulo 2 apresenta a metodologia da pesquisa que detalha elementos da realização do trabalho de conclusão de curso.
- O Capítulo 3 apresenta o embasamento teórico com a descrição de conceitos importantes para o desenvolvimento desse estudo.
- O Capítulo 4 descreve o protocolo do mapeamento sistemático de literatura, também os resultados e a discussão das questões de pesquisa.
- O Capítulo 5 descreve a proposta e as atividades de planejamento de um experimento controlado.
- O Capítulo 6 apresenta os resultados obtidos em um experimento que compara duas técnicas de *data mining*.
- Finalmente, o Capítulo 7 apresenta as considerações finais.

2 METODOLOGIA DA PESQUISA

A metodologia de pesquisa associada ao trabalho de conclusão de curso é embasada em Piffers et al. Peffers et al. (2007). A seguir, apresenta-se os elementos que nortearam a realização deste trabalho de conclusão de curso.

2.1 Exemplo Motivador de Pesquisa

RAS++ foi proposta por (BASSO, 2017) para representação de *assets* híbridos de software. Trata-se de uma Linguagem Específica de Domínio que une meta-informações técnicas sobre mega modelo (B'EZIVIN; JOUAULT; VALDURIEZ, 2004) e meta-informação descritivas oriundas dos padrões *Reusable Asset Specification* (RAS) (OMG, 2005) e AMS (ASSET..., 2014).

Por ter sido concebida com o objetivo de possibilitar uma representação comum para *assets* entre diferentes repositórios e dar suporte a troca de informações entre diferentes tipos de serviços, a RAS++ torna-se uma facilitadora no processo de reúso oportunista através do agrupamento de estruturas de meta-informação que podem ser alvo de algoritmos de recomendação.

Na Figura 1, pode-se observar um exemplo de *asset* representado pela DSL RAS++, com os respectivos metadados que dão suporte à mineração dos mesmos.

Como ilustra a Figura 1, as propriedades da RAS++ de foco deste TCC para a aplicação de técnicas de *data mining* são:

- `Reusable Asset` representa um *asset* armazenado em repositório.
- `Classification` agrupa os elementos para a classificação do *asset*.
- `Descriptor Group` descreve as informações do *asset*. Na imagem, os grupos descritores são padrão do repositório de *assets* de MDE chamado `ReMoDD` (ReMoDD..., 2014). O ReMoDD é um repositório recomendado pela academia para compartilhamento de artefatos derivados de pesquisas publicadas em conferências focadas em MDE. Portanto, caracteriza uma boa fonte de informações.
- `Free Form Value` é a descrição em texto livre, sem uma estrutura específica.
- `Solution` agrupa os artefatos que compõem o *asset*.

2.2 Desenho da Pesquisa

A Figura 2 apresenta o desenho da pesquisa, um recorte da proposta em (BASSO; WERNER; OLIVEIRA, 2017b). A Figura 2 (A) apresenta as três fases onde o RAS++ DSL é útil em um processo de reúso oportunista que visa assistir processos de aquisição de software. A primeira fase visa prover informações suficientes para a especificação de

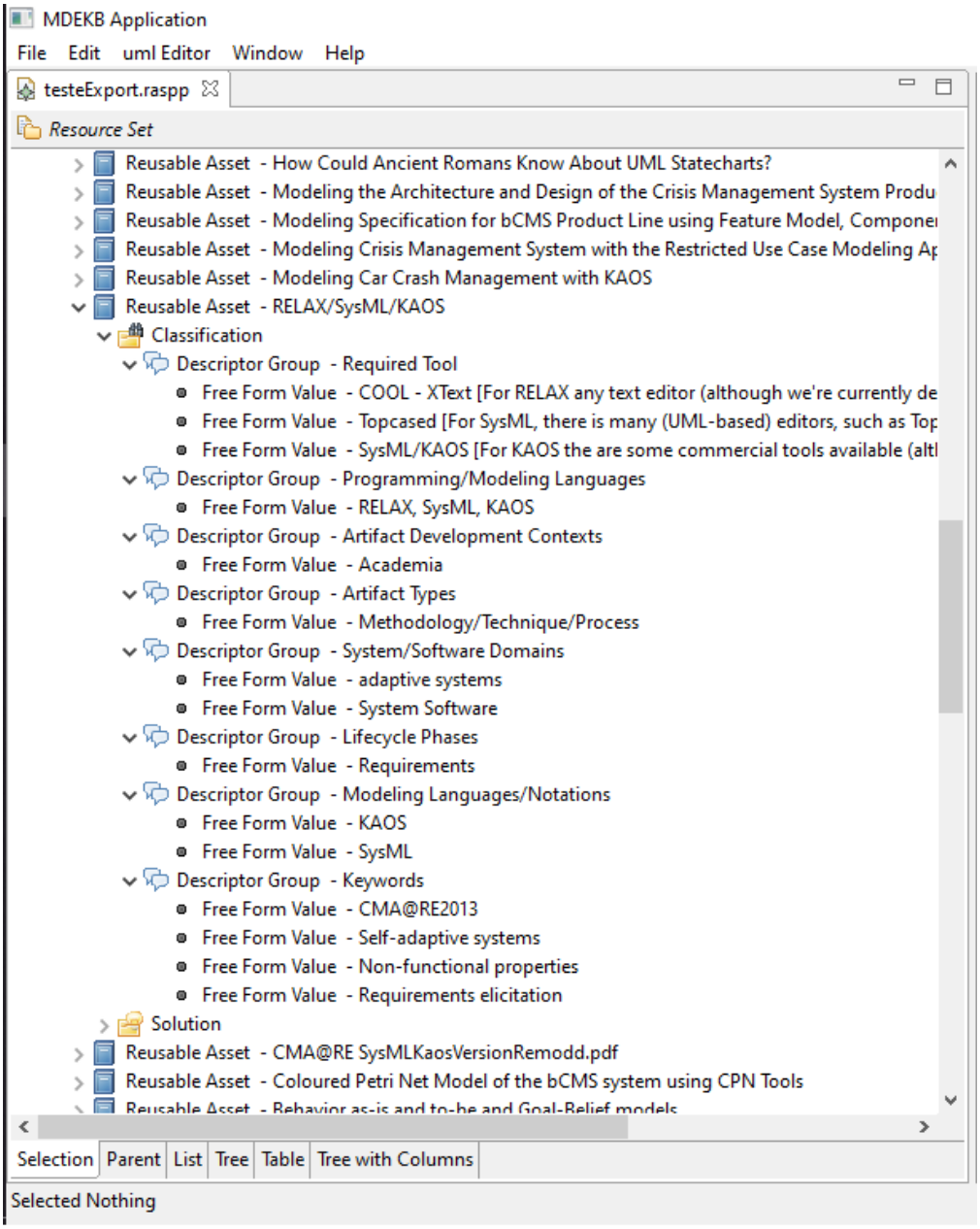

Figura 1 – *Asset* representado pela DSL RAS++.

Fonte: Próprio Autor

meta-dados tratando de tecnologias derivadas de abordagens de MDE. Ou seja, é uma fase que envolve a representação de *assets* de modo independente de repositórios. A segunda fase prevê a utilização do RAS++ em contextos onde o engenheiro de software necessita realizar a aquisição de software, portanto realizando a tomada de decisão. A terceira fase tem por objetivo transformar a informação dos *assets* em representações que são adotadas em ambientes de engenharia de software, como Ambientes de Desenvolvimento Integrado (IDE) (BASSO; WERNER; OLIVEIRA, 2017a), Linguagens de Representação de Processo (PMLs), padrões de integração de ferramentas, e linguagens específicas de domínio para o que é conhecido hoje como *mega-modeling* (ROCCO et al., 2016). A

fase 2 é uma limitação nos trabalhos anteriormente conduzidos no tema (BASSO, 2017), motivada para pesquisa em trabalho recente (NETO et al., 2019).

Como ilustra a Figura 2 (B), os *assets* das tecnologias de MDE podem ser representados com meta-informação em nível descritivo e/ou técnico. Dentro dos *assets* do grupo de pesquisa, existem quatro possíveis estruturas em *assets* que podem ser utilizadas para recomendação de artefatos de MDE: 1) recomendação com base na informação descritiva para classificação, como as ilustradas na Figura 1; 2) recomendação com base na informação estrutural de artefatos de MDE, que atualmente é permitido somente pelo RAS++ e MDE Forge (ROCCO et al., 2016); 3) recomendação com base no contexto de *tool chain* desenhado em uma linguagem específica de domínio para OSLC (FERREIRA, 2020); e 4) recomendação com base no contexto de seleção do modelo de *features* da FOMDA DSL (BASSO et al., 2017). Por motivos de conveniência e limites de tempo, optou-se pelo uso de somente uma estrutura, a descritiva, para classificação.

No contexto desse trabalho, apenas as informações de classificação (contidas nas estruturas de meta-informação `Classification`) são alvo das técnicas de *data mining* dirigidas para recomendação. Já a estrutura `Solution` é explorada pela técnica de *data mining* dirigida para coleção dos dados identificada como "Levantamento de Dados Sobre Tecnologias de MDE". Tal levantamento é feito agora de modo automático, por meio de um algoritmo para *crawling*. Mas, cabe ressaltar que o algoritmo atual serve apenas buscar as referências para arquivos físicos de um repositório alvo, e portanto sendo limitados para estruturas de meta-dados descritos em caixas na Figura 2 (D-F).

Apesar da limitação, as contribuições foco deste estudo são dirigidas para os elementos da Figura 2 (C) e (D), enquanto que os elementos da Figura 2 (E) e (F) não são abordados. Ainda, o foco representacional é em classificação e descrição de padrão de informação. Uma vez que agregam maior complexidade para extrair informações e necessitam de maior tempo de investigação, as demais meta-informações de nível técnico ainda são *gaps* de pesquisa.

Por fim, apesar dos critérios de qualidade para representação de *assets* de MDE preverem um total de seis critérios de representação (BASSO; WERNER; OLIVEIRA, 2017b) para a tomada de decisão em nível de negócio, este estudo consegue explorar uma dessas possibilidades: *C5 - Estrutura para informação descritiva.* Portanto, a contribuição para "Tomada de Decisão em Nível de Negócio" é no sentido de fornecer um primeiro suporte ferramental para assistir o processo de aquisição, e deve ser aprofundado por outros trabalhos de conclusão de curso dentro do grupo de pesquisa.

A Figura 3 apresenta o desenho da pesquisa conduzida conforme as classificações de estudos. Trata-se de um trabalho cuja natureza é aplicada, do tipo exploratória, utilizando-se de dois procedimentos científicos: Pesquisa Bibliográfica, realizado por meio de um estudo de mapeamento sistemático, e Pesquisa Experimental. Para avaliar os componentes desenvolvidos para recomendação, planejou-se Um estudo experimental quanti-

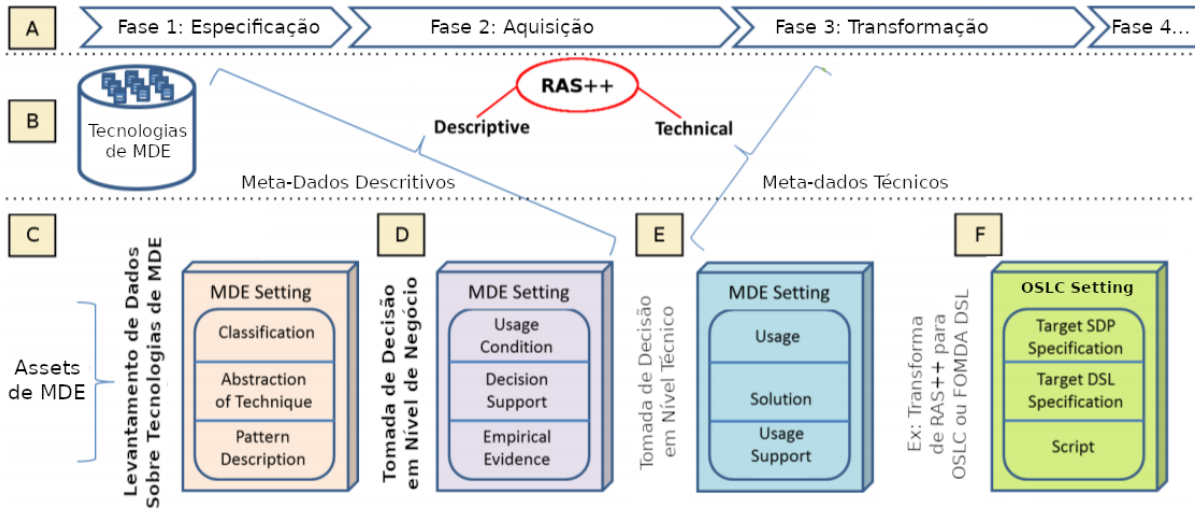

Figura 2 – Desenho da Pesquisa Conforme as Fases de Utilização da RAS++ DSL.

Fonte: Próprio Autor, adaptada de (BASSO; WERNER; OLIVEIRA, 2017b)

tativo, para avaliar o número *assets* retornados como falsos positivos e falsos negativos, determinando assim que técnica traz melhores recomendações.

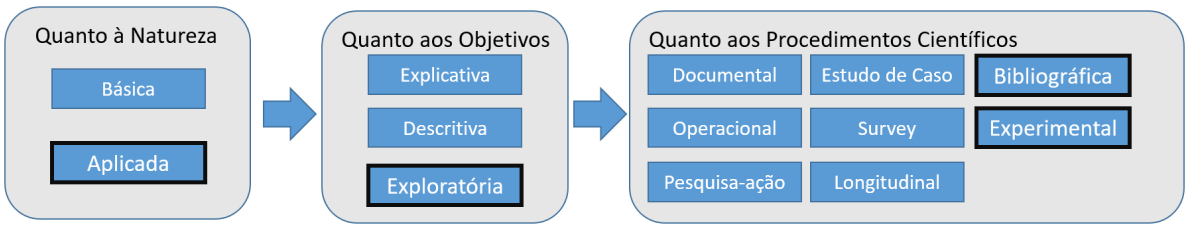

Figura 3 – Desenho da Pesquisa Conforme as Classificações de Estudos.

Fonte: Próprio Autor

2.3 Metodologia

A Figura 4 apresenta dois processos, o primeiro com atividades respectivas à realização do TCC 1, e a segunda com as atividades do TCC 2. Cada uma das atividades ilustradas é um subprocesso. Em linhas gerais, o TCC 1 contou com sub-processo para a concepção do estudo, fundamentação teórica, seleção de tecnologias para o desenvolvimento da solução e um sub-processo para a divulgação dos resultados preliminares. O TCC 2 contou com sub-processos para o refinamento do embasamento teórico e o realinhamento de contribuições para o desenvolvimento da solução de *data mining* e para a divulgação dos resultados.

As atividades para a concepção do estudo são apresentadas na Figura 5. Primeiramente, foi identificado o problema e definidos os objetivos da solução. Inicialmente, a

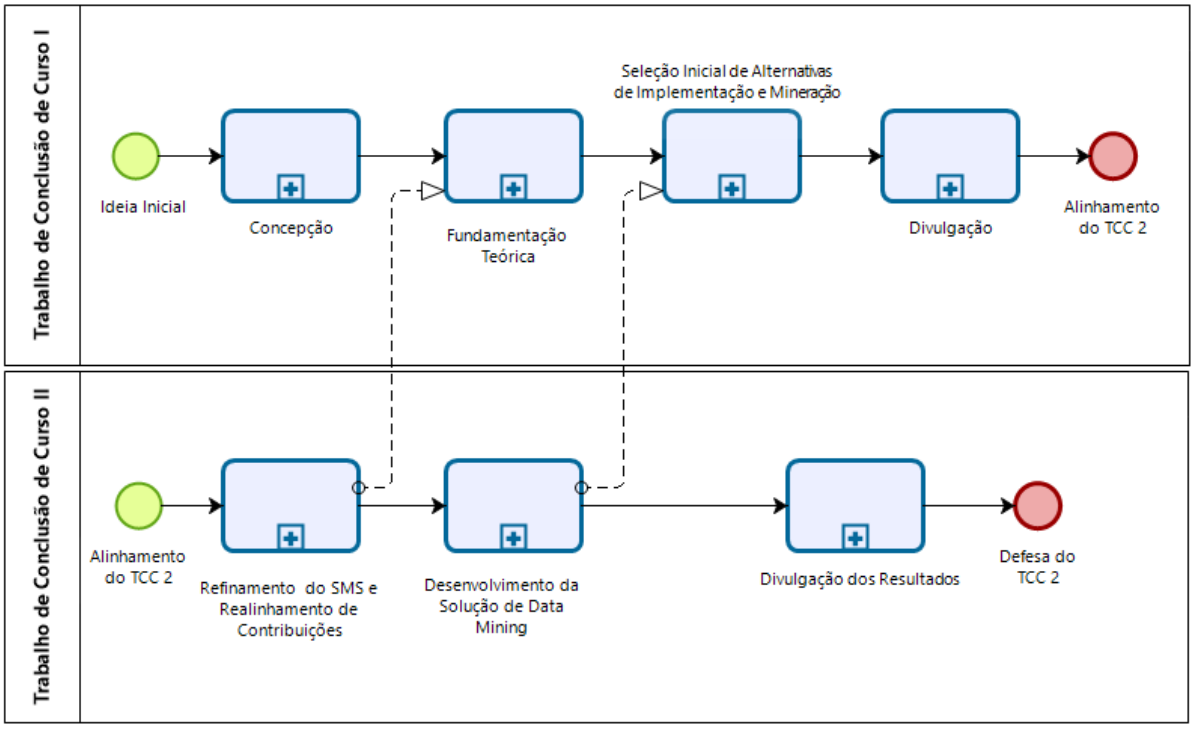

Figura 4 – Os dois Processos para Alinhamento dos Estudos Científicos ao Longo do TCC 1 e 2.

Fonte: Próprio Autor

identificação do problema deu-se em tese de doutorado (BASSO, 2017), como um trabalho futuro/limitação que foi atacado neste novo estudo. Com isso, derivou-se os principais objetivos, estrutura e leitura inicial. O tema de pesquisa em *data mining* foi o foco escolhido nesta temática, também motivado após a realização de um estágio em *big data* realizado no `DTIC`. A partir e 22 de dezembro de 2018, alinhou-se um interesse de pesquisa comum, derivando objetivos para investigar as técnicas adotadas na recomendação de *assets* por meio de *data mining*, com leitura inicial no tema.

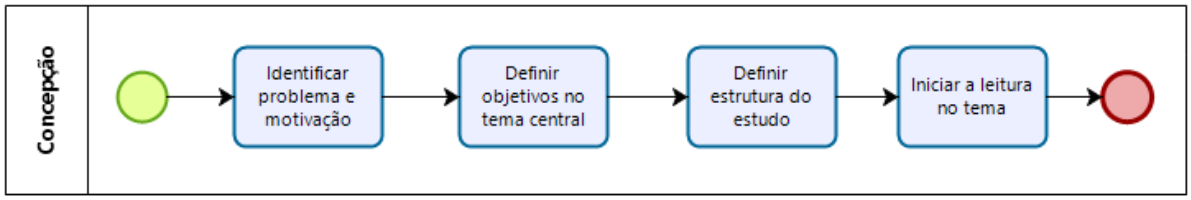

Figura 5 – Passos para a Concepção da Pesquisa.

Fonte: Próprio Autor

A Figure 6 apresenta os passos adotados para aprofundamento no tema de pesquisa. O tema dá continuidade ao que foi investigado anteriormente sobre *assets* para MDE como candidatos para a composição de *tool chains* (BASSO; WERNER; OLI-

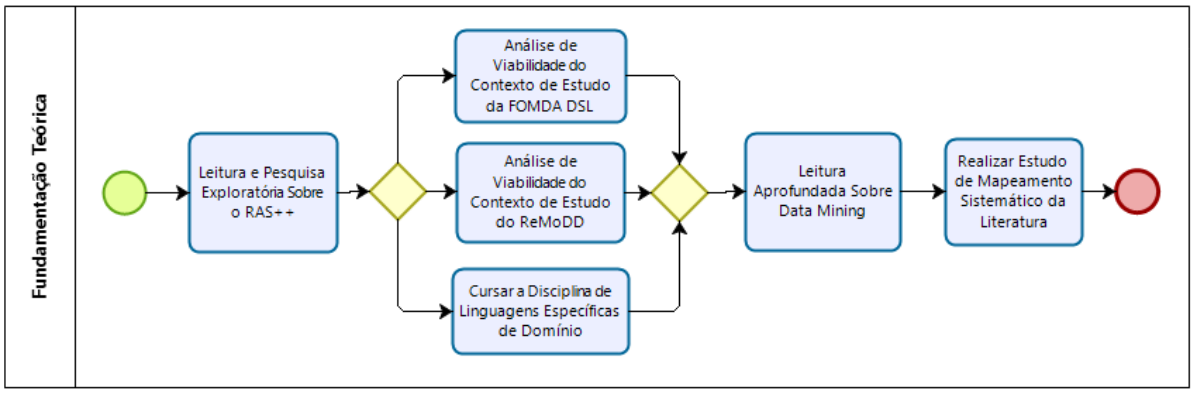

Figura 6 – Passos para o Desenvolvimento da Fundamentação Teórica.

Fonte: Próprio Autor

VEIRA, 2017a). Discutiu-se com o orientador alguns cenários onde data mining seria interessante, como para o reúso oportunista que visa fornecer um apoio para tomada de decisão em contextos de projetos de software quanto à componentes de software. Investigou-se assim, a partir de uma disciplina cursada e denominada Linguagens Específicas de Domínio, a viabilidade de aplicar conhecimento de *data mining* em contextos de integração envolvendo a FOMDA DSL e o repositório ReMoDD.

A problematização da pesquisa também trata de um interesse de pesquisa do grupo LESSE, caracterizado por um projeto de pesquisa denominado "Fundamentação para a Transferência de Tecnologia no MDE como um Serviço". Dentro deste projeto estão previstas ações que fomentem a colaboração entre fábricas de software por meio da recomendação de *assets*. Assim, buscou-se aprofundar na temática de *data mining*, realizando um estudo de mapeamento sistemático cujos trabalhos selecionados são aplicados em problemas de *assets* do interesse da Engenharia de Software.

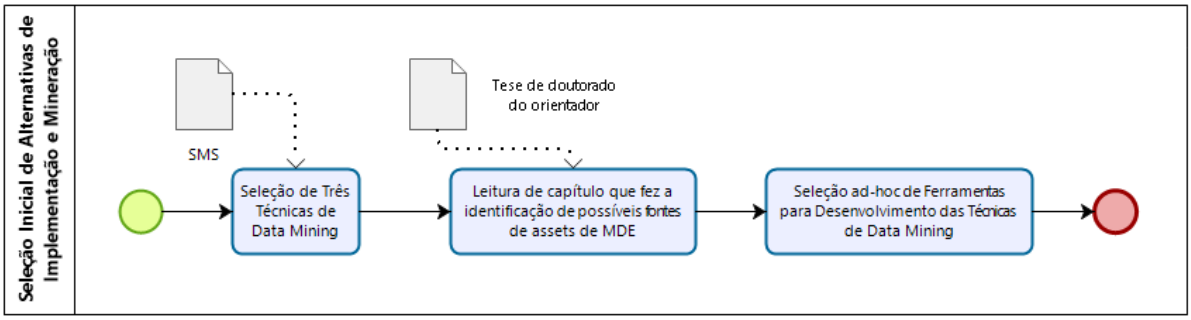

Figura 7 – Passos para a Seleção Inicial de Alternativas de Implementação da Mineração.

Fonte: Próprio Autor

O sub-processo mostrado na Figura 7 apresenta as atividades executadas para levantamento e seleção inicial de alternativas para implementar o sistema proposto. Após a execução do estudo de mapeamento sistemático, selecionou-se três técnicas de *data*

mining para identificar aquelas que melhor atendem as necessidades do projeto. Essa seleção de técnicas necessitou ser reconsiderada posteriormente, uma vez que se teve na prática uma compreensão de modo mais aprofundado das interfaces necessárias para a execução das tarefas de mineração de *assets* de MDE.

Num segundo momento, fez-se leituras do Capítulo 3.3 da tese (BASSO, 2017), que elencou cinco repositórios como fontes de dados sobre tecnologias de MDE.

Num terceiro momento, baseado nas experiências anteriores ao TCC e maior afinidade com Python, selecionou-se um conjunto de ferramentas para o desenvolvimento do suporte ferramental esperado.

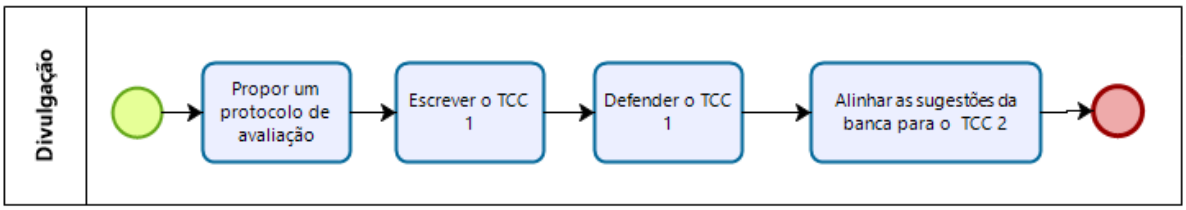

Figura 8 – Passos para a Divulgação dos Resultados do TCC 1.

Fonte: Próprio Autor

A Figura 8 apresenta as atividades para a divulgação dos resultados da pesquisa conduzida, que culminaram na defesa do TCC 1.

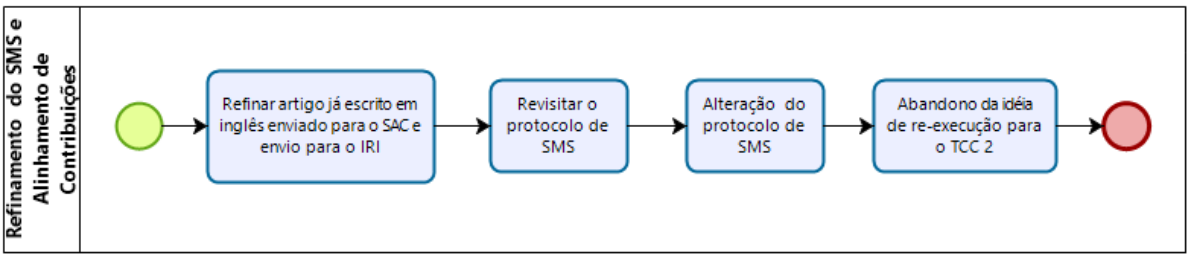

Figura 9 – Passos para o Realinhamento de Atividades da Pesquisa para o TCC 2.

Fonte: Próprio Autor

A Figura 9 apresenta as atividades respectiva ao primeiro sub-processo do TCC 2: Refinamento do SMS e Alinhamento de Contribuições. Cabe salientar que diversas atividades foram conduzidas para colaborar com trabalhos de pesquisa de colegas, estando as atividades do sub-processo restritas ao escopo deste TCC. Assim, fez-se o refinamento de um artigo em inglês tratando da revisão de literatura, este foi submetido para a conferência IRI, porém não foi aceito. De modo à suprir algumas deficiências do trabalho e tornar o material mais interessante, decidiu-se então refinar o protocolo e reexecutar o estudo. No entanto, devido aos prazos, optou-se pela reexecução após a defesa do TCC 2, sendo que este artigo foi momentaneamente suspenso.

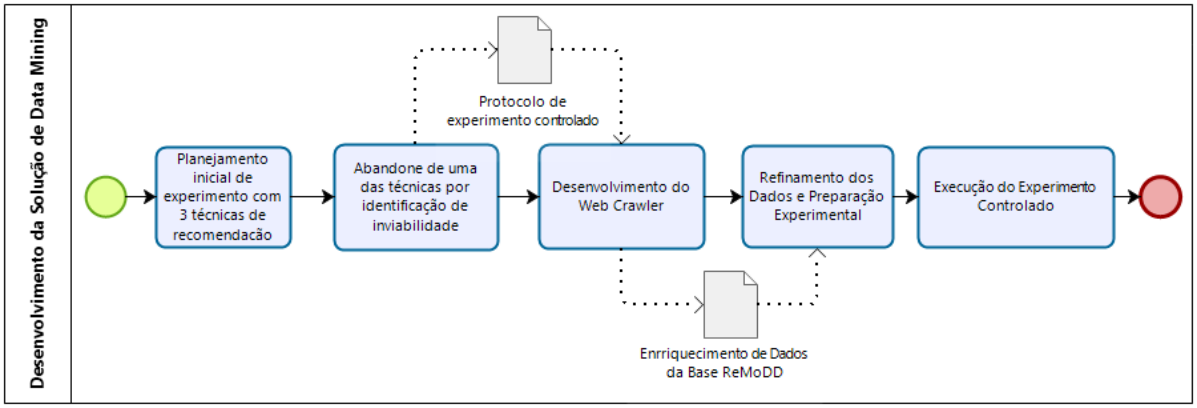

Figura 10 – Passos para o Desenvolvimento da Proposta.

Fonte: Próprio Autor

A Figura 10 apresenta o sub-processo adotado para o desenvolvimento da proposta. Inicialmente, planejou-se a comparação de três técnicas de recomendação, mas uma delas foi abandonada por limitações técnicas dos *assets* do `ReMoDD`. Isso implicou em modificações no protocolo do experimento. Etapas seguintes incluíram o desenvolvimento de um *web crawler* para coletar automaticamente as informações de *assets*. Isso resolveu um problema anterior reportado em (BASSO, 2017): a mineração manual da informação. Os dados coletados foram então preparados para o experimento, que foi executado.

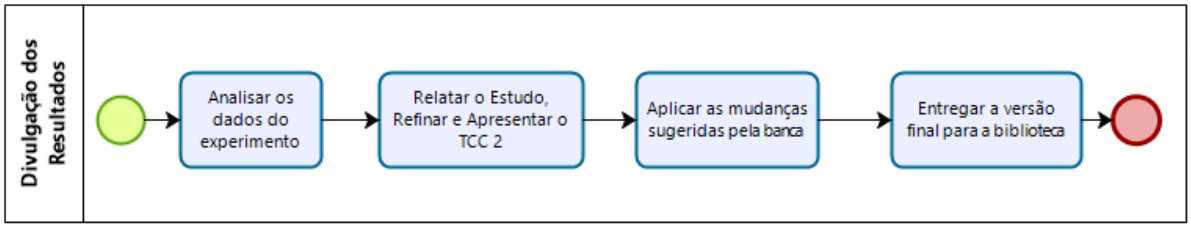

Figura 11 – Passos para a Divulgação dos Resultados do TCC 2.

Fonte: Próprio Autor

O último sub-processo mostrado na Figura 11 apresenta as atividades para a divulgação dos resultados do TCC 2.

3 EMBASAMENTO TEÓRICO

3.1 Data Mining

Data Mining pode ser definido como o processo de analisar conjuntos de dados e inferir padrões destes, através de algoritmos de aprendizado de máquina e métodos estatísticos. De acordo com Fayyad, Piatetsky-Shapiro e Smyth (1996), pode-se categorizar as diferentes abordagens para mineração de dados como:

- Classificação
- Regressão
- Aglomeração
- Sumarização
- Modelo de Dependência
- Detecção de Anomalias

3.1.1 Processo de *Data Mining*

De maneira geral, podemos descrever o processo de *Data Mining* com passos que são comuns independente do contexto do projeto:

1. **Seleção dos Dados:** primeiramente é necessário ter em mente quais dados irão compor o conjunto com que será trabalhado.

2. **Extração de Dados:** dependendo do objetivo do processo, nem todas as informações dentro do conjunto selecionado serão úteis. Para otimizar o processo, apenas as informações relevantes são consideradas.

3. **Transformação de Dados:** dados de diferentes fontes devem ser padronizados para os requerimentos do algoritmo, como exemplifica a Figura 12.

4. **Limpeza dos Dados:** algumas dados valores da amostra podem ser inválidos, como valores faltando ou no formato errado. Tais exemplos tendem a serem desconsiderados, quando não há como recuperar os valores corretos.

5. **Armazenamento Dados:** após obter e padronizar os dados, eles podem ser armazenados para uso futuro.

   O armazenamento, em contextos industriais principalmente, é feito através de *data warehouses*, que nada mais são do que bancos de dados que possuem informações relacionadas a processos e metadados.

Fonte 1

Date	Value
1965-12-31	R$ 2.869
1966-01-03	R$ 2.721
1966-01-04	R$ 2.784
1966-01-05	R$ 3.111

Fonte 2

Date	Value
06/01/66	R$ 4.207
07/01/66	R$ 8.259
10/01/66	R$ 7.386
11/01/66	R$ 5.428

Após formatação

Date	Value
31/12/65	R$ 2.869
03/01/66	R$ 2.721
04/01/66	R$ 2.784
05/01/66	R$ 3.111
06/01/66	R$ 4.207
07/01/66	R$ 8.259
10/01/66	R$ 7.386
11/01/66	R$ 5.428

Figura 12 – Exemplo de transformação de dados.

6. **Analisar e Minerar:** com os dados armazenados e padronizados, pode-se utilizar as ferramentas de mineração por um cientista de dados, que (preferencialmente) conhece os processos e o contexto em que as informações estão inseridas.

7. **Visualização dos Dados:** por fim, é importante que os resultados possam ser facilmente compreendidos pelos interessados. Para isso costuma-se usar gráficos e infográficos.

3.1.2 Classificação

Classificação utiliza algoritmos de aprendizado supervisionado para identificar padrões em um conjunto de dados, cujas características já são bem conhecidas, a fim de poder inferir como uma nova informação se encaixaria dentro do conjunto já existente, podendo citar como exemplo a Regressão Logística, baseada na Função Logística, uma função *sigmóide* definida de forma genérica como 3.1.

$$f(x) = \frac{1}{1 + \mathrm{e}^{-x}} \tag{3.1}$$

Dado qualquer valor de x, a função logística retornará um valor de 0 à 1 para representar a probabilidade da variável x pertencer a uma classificação específica (no caso, o valor se aproximaria de 1), ou pertencer à outra classificação (no caso, o valor se aproximaria de 0).

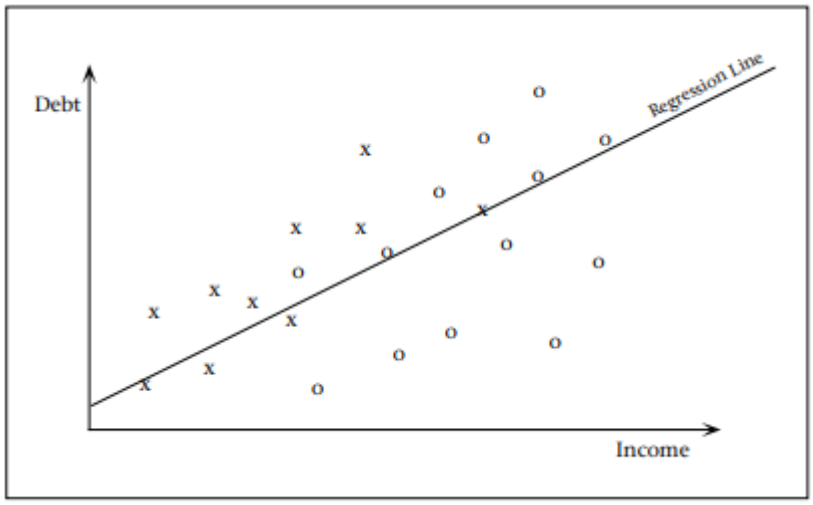

Figura 13 – Exemplo de resultado de uma regressão linear.

Fonte: Fayyad, Piatetsky-Shapiro e Smyth (1996)

3.1.3 Regressão

Regressão, assim como as abordagens de classificação, busca identificar padrões em conjuntos de dados, porém, estimando a relação entre variáveis numéricas contínuas.

Como exemplo, pode-se citar a amplamente difundida Regressão Linear Simples. Nela, objetiva-se representar o conjunto com um modelo linear 3.2. A Figura 13 ilustra o resultado de uma regressão.

$$Y_i = aX_i + \beta + \epsilon \tag{3.2}$$

3.1.4 Aglomeração

Comumente chamada de *Clustering* ou *Cluster Analysis*, trata-se de agrupar objetos em grupos, de modo a identificar um conjunto de categorias, baseando-se na semelhança entre os objetos agrupados, como mostra a Figura 14. Uma maneira comum de verificar a semelhança entre dois objetos, é calcular a distância entre eles. Uma das distâncias mais utilizadas é a Euclidiana (3.3).

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{3.3}$$

3.2 Assets

Um *asset*, no contexto da Engenharia de Software, pode ser caracterizado como qualquer artefato produzido durante o ciclo de vida do desenvolvimento de software e que pode ser reutilizado.

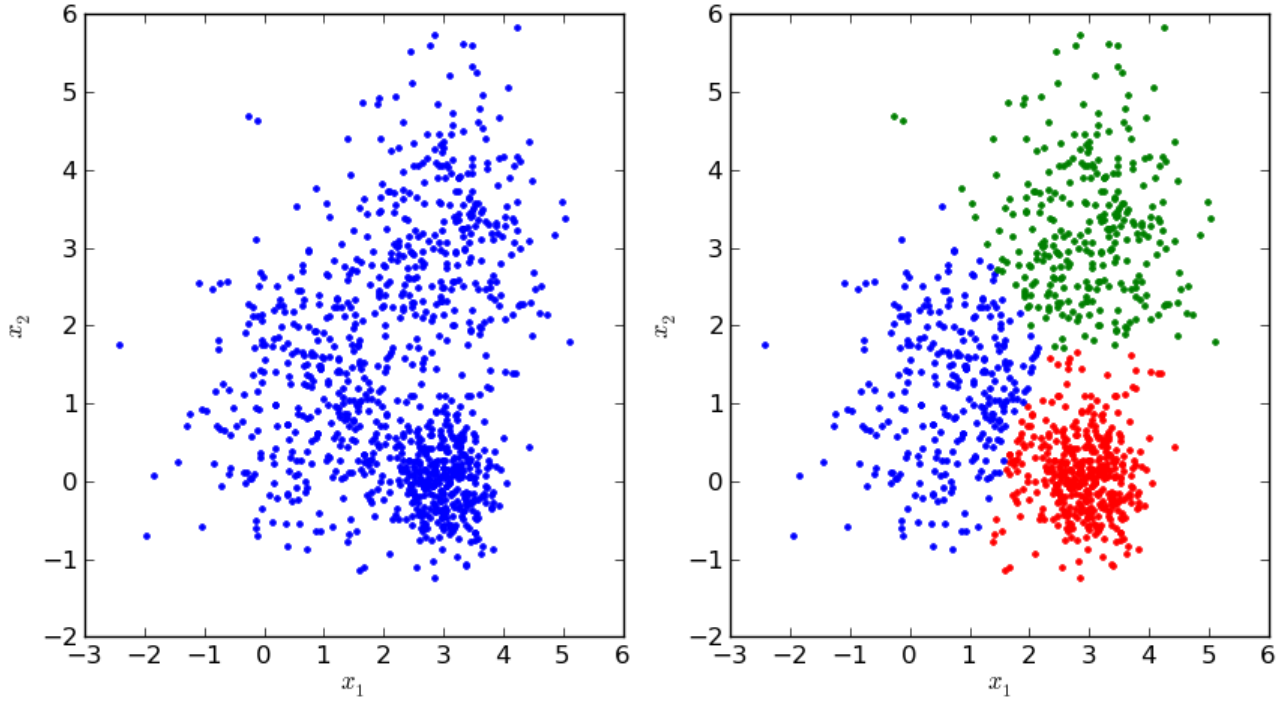

Figura 14 – Exemplo de clusterização com K-Means.

Fonte: https://mubaris.com/posts/kmeans-clustering/

De acordo com a definição dada pela (OSLC. . . , 2012), *asset* é qualquer coisa que pode ser mantida a fim de gerar valor. No contexto de software, *assets* podem ser código-fonte (funções, classes, módulos, etc), modelos, ferramentas e componentes.

3.3 Técnicas

3.3.1 Algoritmo Genético

Algoritmo genético pertence a uma classe de meta-heurísticas de otimização baseadas no comportamento da natureza e mais especificamente, no processo evolutivo.

A ideia do algoritmo é simples: iniciar com uma população de indivíduos (possíveis soluções) gerados aleatoriamente, e então gerar inúmeras versões de uma população a partir dos indivíduos mais adaptados da geração anterior, até que uma condição de parada seja alcançada. Em nosso caso, uma população seria composta por indivíduos, que por sua vez são grupos de *assets* representando componentes de software que são armazenados neste repositório. Podemos ter vários repositórios com diferentes *assets*.

Indivíduo: cada indivíduo carrega o material genético que representa uma possível solução para o problema alvo. Em nosso caso, cada indivíduo é uma possível lista de *assets* retornados em uma busca no repositório por determinadas informações descritivas. Ou seja, é cada possível combinação, não importa se melhor ou pior, de *assets* que

atendam à determina busca.

Função *Fitness*: é a função que avalia o quão bem adaptado um indivíduo está, ou seja, o quão satisfatória é a solução que ele representa. Ou seja, uma vez que os indivíduos não caracterizam a solução ideal de retorno, a função *fitness* apresenta uma pontuação para o contexto da busca. Com isso, é possível determinar aqueles indivíduos que deveriam ser retornados na busca.

Reprodução: o processo de criação de uma nova geração de indivíduos acontece através da reprodução entre indivíduos da geração anterior. O resultado é um novo indivíduo cujo material genético é uma mistura do material genético dos pais. Em nosso contexto, a reprodução geraria duas novas seleções compostas de assets contidos em dois grupos/indivíduos, cada um contendo assets diferentes.

Para esse processo de reprodução, é necessário primeiramente que os pais sejam selecionados entre os indivíduos da população. Uma maneira comum de proceder com o processo de seleção é utilizar um algoritmo de roleta ou de torneio, em que os pais serão selecionados aleatoriamente, com os indivíduos mais bem adaptados tendo mais chances de serem selecionados, porém, não excluindo os menos adaptados, a fim de manter a diversidade da população. Ou seja, nesta parte do algoritmo a geração dos novos indivíduos é feita de modo probabilístico.

Durante a geração de novos indivíduos também é comum permitir, com uma chance pequena, que um dos genes do novo indivíduo sofra uma mutação, também a fim de ampliar a diversidade das soluções. Isso também é feito de modo aleatório.

Por fim, quando uma certa condição for alcançada, o algoritmo terá uma população de onde será selecionado o indivíduo mais bem adaptado, sendo esse a solução final do algoritmo.

Por se tratar de uma meta-heurística não determinística, é importante ressaltar que a solução encontrada pode não ser a solução ótima para o problema.

3.3.2 Rede Neural Artificial

Redes Neurais Artificiais são sistemas baseados na estrutura de redes neurais artificiais, com funcionamento similar ao de um cérebro. São utilizados para aprenderem e realizarem atividades de maneira autônoma após serem treinados para tal.

Em relação ao treino de uma rede neural, este pode ser supervisionado ou não supervisionado. **Aprendizado Supervisionado** ocorre quando a rede neural é treinada com dados conhecidos e categorizados previamente. **Aprendizado Não Supervisionado** ocorre quando a rede neural é treinada com dados não categorizados previamente, cabendo a própria rede neural identificar padrões nas informações. É útil quando é necessário encontrar padrões desconhecidos.

3.3.3 Regra de Associação

Regra de Associação é um método de aprendizado de máquina proposto por (AGRAWAL; IMIELIńSKI; SWAMI, 1993), primariamente para encontrar relações entre vendas de produtos, sendo portanto, útil para identificar padrões entre variáveis em grandes volumes de dado.

De acordo com (AGRAWAL; IMIELIńSKI; SWAMI, 1993), o problema pode ser definido como:

Seja $I = I_1, I_2, ..., I_n$ um conjunto de n itens e T um conjunto de transações, e cada transação t é representada por um vetor binário, com $t_k = 1$ caso o item I_k faça parte da transação, ou 0 caso não faça parte. Sendo X um subconjunto de I, busca-se inferir uma relação $X \Rightarrow \mathrm{I}_j$.

Em outras palavras significa que, em uma compra em uma loja por exemplo, um cliente que compra os itens pertencentes ao conjunto X, tem grandes chances de também se interessar pelo item I_j. Outro exemplo, agora dentro do contexto de *assets* de software, dois ou mais *assets* podem ser complementados por um terceiro dentro de uma cadeia de transformação.

3.4 Domain-Specific Language

Uma linguagem de domínio específico pode ser definida como uma linguagem feita com o objetivo de resolver um problema específico. Portanto, ao contrário de linguagens de propósito geral (como C++ e Python), DSLs tendem a ser limitadas, tendo utilidade somente dentro do domínio para qual foram projetadas. (FOWLER, 2010) classifica DSLs em três categorias:

- Uma **DSL externa** é uma linguagem independente da aplicação em que está inserida, sendo normalmente analisada pela linguagem original da aplicação para que tenha uso. Uma consulta SQL a partir de uma aplicação em Python é um exemplo de uso de DSL externa. Nesse exemplo, seria utilizada uma biblioteca escrita em Python para iniciar a consulta em SQL, e para converter os resultados da consulta para serem utilizados na aplicação.

- **DSL interna** pode ser um script ou biblioteca escrita com uma linguagem de propósito geral, porém, com estilo único e destinada para resolver problemas específicos. O *framework* Rails, da linguagem Ruby, é um exemplo conhecido de DSL interna.

- **DSL *workbench*** é uma ferramenta específica para criação a criação de DSLs e de ambientes em que a DSL possa ser utilizada. MPS da JetBrain e Eclipse Xtext podem ser citados como exemplos.

Seu uso abrange várias áreas dentro do área de tecnologia, mas não se restringe apenas aos profissionais da área, já que por apresentarem uma sintaxe própria para um domínio, pessoas de qualquer área do conhecimento podem se tornar familiares com a DSL em questão e utilizá-la diariamente, como as utilizadas pelo Excel e PowerBI, ambas aplicações da Microsoft.

Outra vantagem que encoraja o uso de DSLs é a capacidade de representar e modelar um problema de domínio, bem como ter o desenvolvimento de uma solução orientada por ele, sendo esse inclusive o objetivo da área de *Domain Driven Design* (ou Projeto Orientado à Domínio). Neste trabalho, foi utilizada a DSL RAS++ para a representação conjunta dos *assets* utilizados.

4 MAPEAMENTO SISTEMÁTICO DE LITERATURA (SMS)

Uma vez que este Trabalho de Conclusão de Curso explora técnicas de *data mining* para um contexto aplicado para a reutilização de software, precisamente na recomendação de especificações de ativos (ou *assets*) que descrevem ferramentas, uma revisão de literatura foi executada para encontrar técnicas utilizadas neste contexto. Com a finalidade de classificar as possíveis técnicas e mapeá-las para possíveis tipos de assets, elaborou-se um Estudo de Mapeamento Sistemático de literatura, do Inglês *Systematic Mapping Study (SMS)*. Este capítulo apresenta o protocolo de SMS e sua execução.

4.1 Protocolo do SMS

4.1.1 Questões de pesquisa

Para especificar o objetivo dessa pesquisa, foi utilizada a abordagem GQM (CALDIERA; ROMBACH, 1994), apresentada na Tabela 1.

Objetivo	Com o objetivo de identificar e classificar
Questão	técnicas, metodologias, desafios, tendências
Objeto	mineração de dados de repositórios de assets, componentes e código
Viewpoint	do ponto de vista do pesquisador

Tabela 1 – Objetivo da pesquisa.

A partir do objetivo especificado, foram concebidas as questões de pesquisa as quais objetivamos responder neste trabalho.

- **Q1**: Quais são as propostas de técnicas/metodologias/*guidelines* propostas e aplicadas na área de mineração de dados úteis à aquisição/reúso de software/desenvolvimento?
- **Q2**: Quais os desafios atuais da área de mineração de dados quando aplicada neste contexto?
- **Q3**: Que mecanismos de recomendação são utilizados?
- **Q4**: Quais são as tendências das pesquisas na área?
- **Q5**: Quais são as formas de avaliação discutidas nos estudos?
  - **Q5.1**: Quais estudos são aplicados em casos reais em análises de viabilidade?
  - **Q5.2**: Quais foram os métodos de avaliação empregados no estudo?
  - **Q5.3**: Quais foram os critérios de aceitação do produto final?
  - **Q5.4**: Quais foram seus pontos positivos e negativos observados no estudo?
  - **Q5.5**: Que desafios existem na execução de estudos de viabilidade?

4.1.2 Estratégia de Busca

A pesquisa foi feita em 4 bases digitais de artigos amplamente conhecidas e utilizadas pela comunidade científica (CHEN; BABAR; ZHANG, 2010). As bases utilizadas são apresentadas na Tabela 2.

Scopus	https://www.scopus.com/home.uri
IEEE	https://ieeexplore.ieee.org
Springer	https://www.springer.com
ACM	https://dl.acm.org/

Tabela 2 – Bases utilizadas na pesquisa.

4.1.3 String de Busca

A *string* de busca foi pensada através da estratégia PICo (PREGUNTA, 2007), como mostra a Tabela 3.

P	"data mining"OR "big data"OR "data extraction"OR “knowledge management” OR “knowledge discovery”
I	"asset management specification"OR ams OR "reuse repository"OR "component repository"OR "mde repository"OR "code recommender"OR "recommender system"OR "asset recommender"OR "software asset"OR "component asset"OR "reusable asset"OR RAS
Co	“technology transfer” OR “software acquisition” OR “integration” OR “software engineering” OR “software development” OR “software reuse”

Tabela 3 – Aplicação da estratégia PICo

Vale ressaltar que foram necessárias algumas modificações específicas para algumas das bases pesquisadas. A combinação de cada item da estratégia PICo se da através do operador lógico AND. As *strings* definitivas são apresentadas na Tabela 4.

4.1.4 Critérios de Seleção

A seguir, são apresentados os critérios de inclusão que os estudos devem seguir a fim de serem considerados relevantes para esta pesquisa, bem como os critérios de exclusão, que desqualificarão os estudos que se enquadrem em algum deles.

4.1.4.1 Critérios de Inclusão

1. Estudos que apresentem técnicas/metodologias/guidelines para mineração de dados.

2. Estudos que relatam estudos de caso sobre técnicas/metodologias/guidelines para mineração de dados.

Base	String
Scopus	(TITLE-ABS-KEY (("data mining"OR "big data"OR "data extraction"OR "knowledge management"OR "knowledge discovery") AND ("asset management specification"OR ams OR "reuse repository"OR "component repository"OR "mde repository"OR "code recommender"OR "software asset"OR "component asset"OR "reusable asset"OR ras OR "recommender system"OR "asset recommender") AND ("technology transfer"OR "software acquisition"OR "integration"OR "software engineering"OR "software development"OR "software reuse")))
IEEE	("data mining"OR "big data"OR "data extraction"OR "knowledge management"OR "knowledge discovery") AND ("asset management specification"OR ams OR "reuse repository"OR "component repository"OR "mde repository"OR "code recommender"OR "software asset"OR "component asset"OR "reusable asset"OR ras OR "recommender system"OR "asset recommender") AND ("technology transfer"OR "software acquisition"OR "integration"OR "software engineering"OR "software development"OR "software reuse")
Springer	("data mining"OR "big data"OR "data extraction"OR "knowledge management"OR "knowledge discovery") AND ("asset management specification"OR "reuse repository"OR "component repository"OR "mde repository"OR "code recommender"OR "software asset"OR "component asset"OR "reusable asset"OR "asset recommender") AND ("technology transfer"OR "software acquisition"OR "software engineering"OR "software development"OR "software reuse")
ACM	("data mining"OR "big data"OR "data extraction"OR "knowledge management"OR "knowledge discovery") AND ("asset management specification"OR ams OR "reuse repository"OR "component repository"OR "mde repository"OR "code recommender"OR "software asset"OR "component asset"OR "reusable asset"OR ras OR "recommender system"OR "asset recommender") AND ("technology transfer"OR "software acquisition"OR "integration"OR "software engineering"OR "software development"OR "software reuse")

Tabela 4 – *Strings* de busca utilizadas.

3. Estudos que tratem de sistemas de recomendação de *assets*/componentes/código em cenários de aquisição de software.

4.1.4.2 Critérios de Exclusão

1. Estudos que não envolvam mineração de dados
2. Estudos que não estejam em inglês
3. Estudos duplicados
4. Estudos que não estejam disponíveis para download

5. Estudos que sofreram de retratação ou foram invalidados
6. Estudos que não são primários
7. Estudos que não tratem de transferência de tecnologia do domínio da engenharia de software nem de aquisição ou reúso de software.

4.1.5 Critérios de Qualidade

Os critérios de qualidade foram definidos com o propósito de verificar a completude dos trabalhos que serão discutidos nas próximas seções. Por se tratar de um mapeamento, nenhum trabalho foi desconsiderado baseando-se nos critérios de qualidade.

1. O desafio/problema que motivou o trabalho é bem definido? S: O estudo apresenta o problema que o motivou. P: O estudo apenas cita o problema, mas não disserta sobre ele. N: A motivação do estudo não é apresentada.
2. A forma de validação é bem especificada?
   - Valor 3 = O estudo é do tipo *evaluation research* (Controlled Experiment, Real-world case study, Real-world action research).
   - Valor 2 = O estudo é do tipo *validation research* (Analytical study, Survey, Simulation, Focus group).
   - Valor 1 = O estudo é do tipo *solution proposal* (Proof of concept, Conceptual demonstration).
   - Valor 0 = O estudo não apresenta uma avaliação. Ou seja, o estudo ou é ou apenas do tipo *philosophical paper*, ou do tipo *experience report*, ou do tipo *opinion paper*.
3. O trabalho deixa clara a sua contribuição para a área? S: O estudo deixa clara a sua contribuição para a área. P: O estudo cita superficialmente como a sua proposta contribui para a área. N: O estudo não discute suas contribuições.

4.1.6 Formulário de Extração de Dados

4.2 Execução do SMS

Nessa seção são apresentados os resultados da execução do Estudo de Mapeamento Sistemático de literatura.

4.2.1 Resultados das Bases

Inicialmente, foram obtidos 888 artigos, como mostra a Tabela 6, entre eles. A busca foi feita considerando título, resumo e palavras-chave.

Dados	Descrição	Questão de busca
Ano	Ano de publicação.	Q2/Q4
Objeto de *Asset*	Tipo de *asset* relacionado ao trabalho.	Q1
Proposta/Contribuição	Solução aplicada no artigo relacionada à *data mining*.	Q1/Q4
Método de validação	Método de validação da proposta apresentada no artigo.	Q5

Tabela 5 – Dados à serem extraídos dos estudos.

Base	Resultados
IEEE	78
Scopus	201
Springer	315
ACM	294
	888

Tabela 6 – Artigos retornados

4.2.2 Aplicação dos Critérios de Seleção

Após obtermos os resultados das bases, foram aplicados os critérios de exclusão. A Tabela 7 apresenta a quantidade de estudos desconsiderados por se encaixarem nos critérios.

Critério / Base	CE1	CE6	CE7	Total
IEEE	4	4	61	69
ACM	14	7	262	281
Scopus	12	29	147	185
Springer	71	81	145	297
				833

Tabela 7 – Resultado da aplicação dos critérios de exclusão.

Além disso, 55 estudos estavam repetidos entre as bases, sendo que destes, 44 estavam entre os descartados, e 11 entre os aceitos, resultando em 42 estudos aprovados após a leitura dos titulos, *abstract* e palavras-chave, e 19 após a leitura do texto completo, como mostra a Figura 15.

O processo de seleção pode ser visto na Figura 16.

4.2.3 Aplicação dos Critérios de Qualidade

No geral, foram obtidos bons artigos, que em sua maioria, apresentaram todos os aspectos esperados, como pode-se observar na Tabela 8.

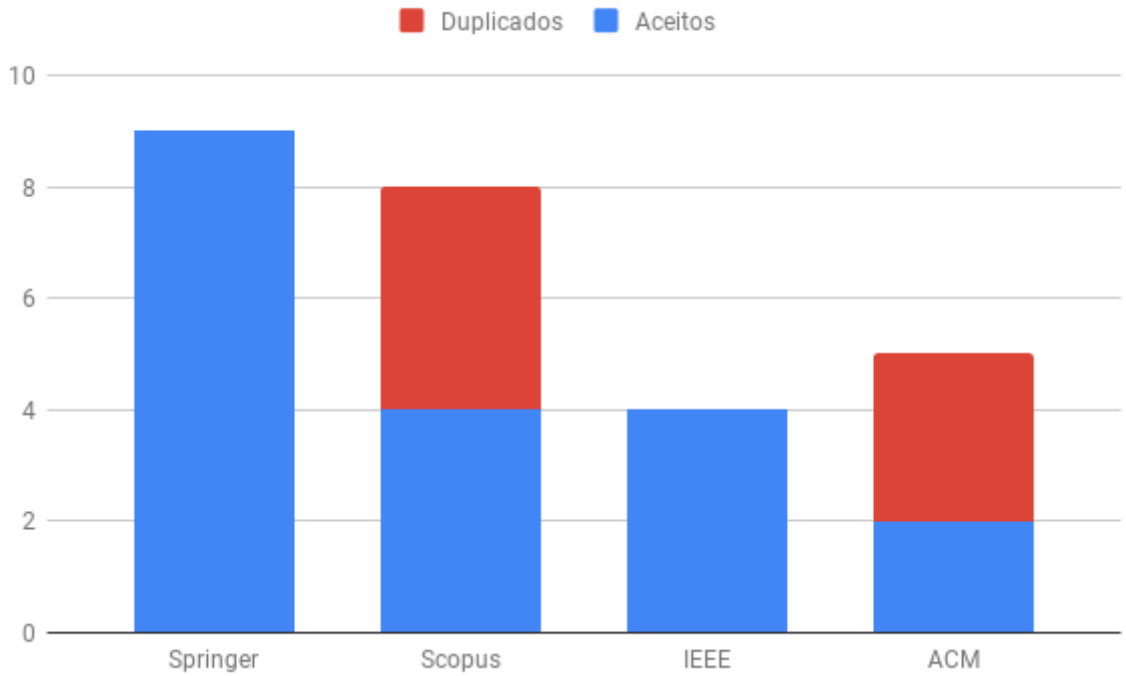

Figura 15 – Artigos aceitos após leitura do texto completo.

4.3 Resultados

Nesta seção serão discutidos os resultados da pesquisa e as questões que a motivaram.

4.3.1 Técnicas Aplicadas (Q1)

Pode-se observar duas abordagens diferentes em relação às propostas dos trabalhos obtidos: utilizar técnicas de clusterização e organização de repositórios de *assets*; e utilizar técnicas de filtragem e exploração de dados a fim de recomendar *assets* que possam ser uteis dependendo do contexto. Além disso, alguns trabalhos também buscam recursos em ambientes web. A relação de fases abordadas está representada na Tabela 16.

Na Tabela 15 é apresentada a relação de quais técnicas foram citadas nos trabalhos e em quais conjuntos de *assets* foram aplicadas.

4.3.2 Desafios da área de mineração de dados aplicada em aquisição e reúso de *assets*. (Q2)

O trabalho de (YE; LO, 2000) cita como os maiores desafios para a ampla adoção do reuso de software o armazenamento e recuperação de software. Além disso, (YE; LO, 2000) também aponta a dificuldade de obter uma boa precisão em técnicas de criação automatizada de bibliotecas.

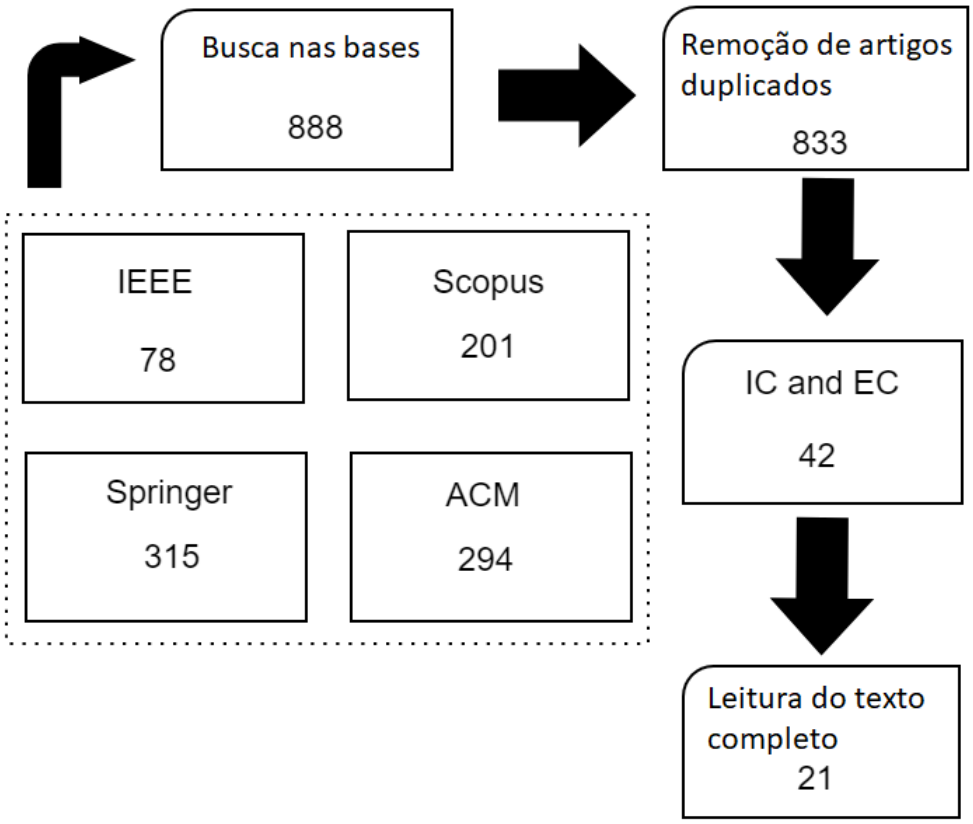

Figura 16 – Processo de seleção.

O restante dos trabalhos, por mais que não citem abertamente esse desafio, buscam contribuir com uma nova abordagem para mitigar o mesmo problema, resultando na diversidade de propostas distintas encontradas.

4.3.3 Mecanismos de Recomendação (Q3)

Com relação aos mecanismos de recomendação identificados, não obtivemos nenhum indício de que há uma técnica que se destaque. Como pode ser observado na Tabela 17, a única técnica de aprendizado de máquina que foi aplicada em mais de um trabalho foi algoritmo evolutivo.

4.3.4 Tendências de Pesquisa (Q4)

Pode-se observar uma homogeneidade entre as técnicas identificadas, como ilustra a Figura 17, o que pode ser um indicador de que ainda não foi encontrada/concebida uma abordagem próxima ao considerado ideal para a mineração e recomendação de *assets* de *software*.

Em relação aos tipos de *assets* que eram foco nos respectivos estudos, 42% dos artigos trabalhavam com componentes de software, enquanto o restante focava em um

	CQ1	CQ2	CQ3
(WANG; REN, 2011)	S	2	S
(YE; LO, 2000)	S	2	S
(VESCAN, 2015)	S	2	S
(NAKKRASAE; SOPHATSATHIT, 2004)	S	2	S
(ELKAMEL; GZARA; BEN-ABDALLAH, 2016)	S	3	S
(LI et al., 2004)	S	2	S
(BASCIANI et al., 2016)	S	2	S
(DIAMANTOPOULOS; KARAGIANNOPOULOS; SYMEONIDIS, 2018)	S	2	S
(HEINEMANN, 2012)	S	2	S
(DUMITRU et al., 2011)	S	2	S
(MCCAREY; CINNÉIDE; KUSHMERICK, 2005)	S	2	S
(SAYYAD; AMMAR; MENZIES, 2012)	S	2	S
(BAJRACHARYA; OSSHER; LOPES, 2009)	S	2	S
(Martins et al., 2009)	S	2	S
(LE et al., 2018)	S	1	S
(WANG; LIU; FENG, 2004)	S	2	S
(KöGEL, 2017)	S	2[1]	S
(VODITHALA; PABBOJU, 2015)	S	0	P
(WU et al., 2007)	P	0	P
(BAWA; KAUR, 2017)	P	0	N
(KUMAR; BHATIA; KUMAR, 2011)	P	0	P

Tabela 8 – Aplicação dos critérios de qualidade.

tipo específico de artefato. A Figura 18 mostra em detalhes a proporção de *assets* focados nos estudos.

4.3.5 Formas de validação dos estudos (Q5)

Os estudos identificados não foram suficientes para responder às subquestões Q5.1, Q5.3, Q5.4, Q5.5, portanto decidimos responder apenas a questão principal através da categorização proposta por (KITCHENHAM; LINKMAN; LAW, 1997).

- Evaluation research = O estudo é do tipo evaluation research (Controlled Experiment, Real-world case study, Real-world action research).

  (ELKAMEL; GZARA; BEN-ABDALLAH, 2016) foi o único trabalho a apresentar um experimento com a participação de pessoas. O grupo de participantes foi composto por 10 doutorandos em ciência da computação de 10 mestrandos em engenharia de *software*. Durante o experimento, os participantes foram instruídos a modelar um modulo de sistema, com e sem o suporte da ferramenta apresentada pelo autor, e os resultados obtidos foram comparados.

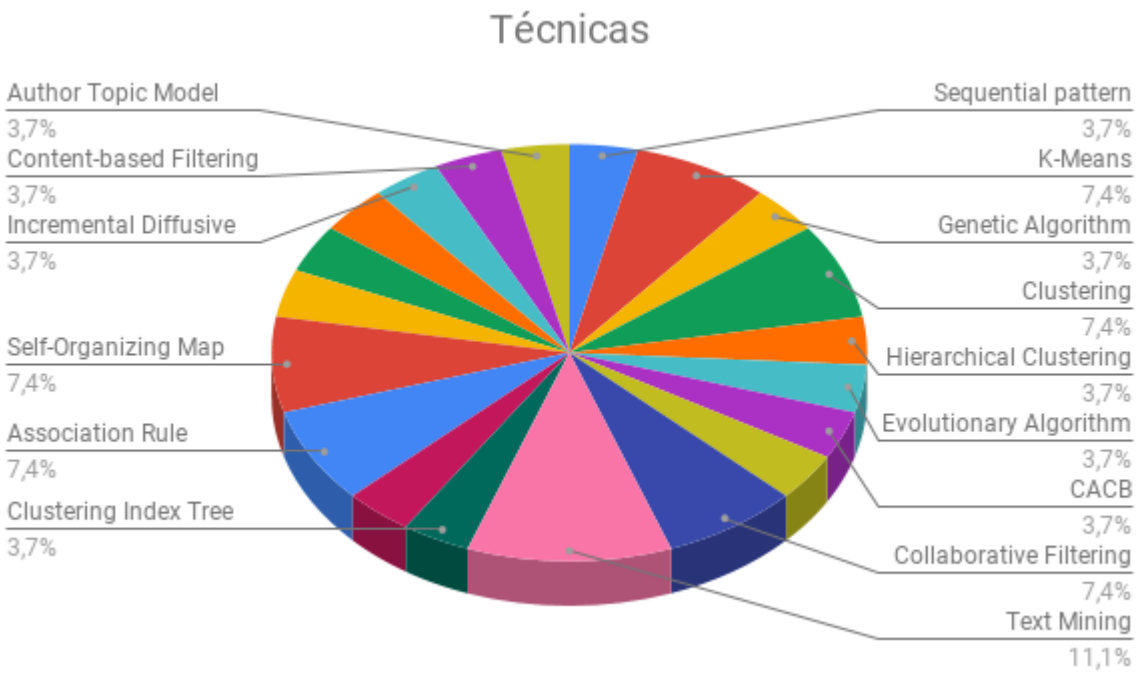

Figura 17 – Proporção de técnicas identificadas.

Fonte: Próprio Autor.

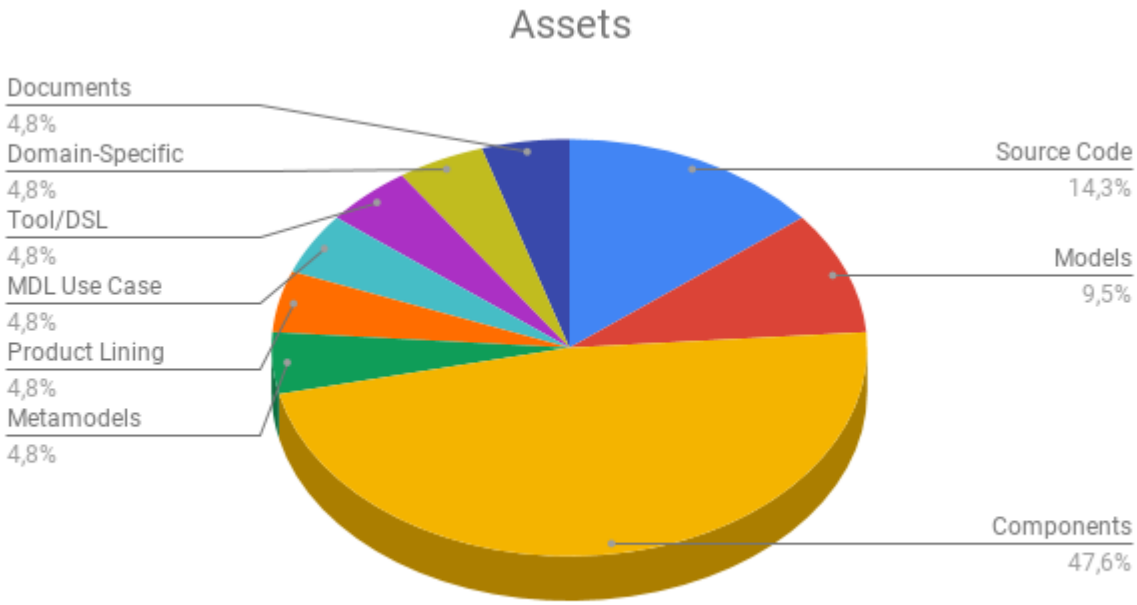

Figura 18 – Proporção entre os tipos de *assets* identificados.

Fonte: Próprio Autor.

Referência do Estudo	Busca	Organização	Recomendação
(YE; LO, 2000)		x	
(NAKKRASAE; SOPHATSATHIT, 2004)		x	
(LI et al., 2004)			x
(WANG; LIU; FENG, 2004)		x	
(MCCAREY; CINNÉIDE; KUSHMERICK, 2005)			x
(WU et al., 2007)		x	
(BAJRACHARYA; OSSHER; LOPES, 2009)	x	x	
(Martins et al., 2009)			x
(WANG; REN, 2011)		x	
(DUMITRU et al., 2011)	x		x
(HEINEMANN, 2012)			x
(KUMAR; BHATIA; KUMAR, 2011)		x	
(SAYYAD; AMMAR; MENZIES, 2012)			x
(VODITHALA; PABBOJU, 2015)		x	
(VESCAN, 2015)			x
(ELKAMEL; GZARA; BEN-ABDALLAH, 2016)		x	x
(BASCIANI et al., 2016)		x	
(BAWA; KAUR, 2017)		x	
(KöGEL, 2017)			x
(LE et al., 2018)			x
(DIAMANTOPOULOS; KARAGIANNOPOULOS; SYMEONIDIS, 2018)	x	x	

Tabela 9 – Fases identificadas nos trabalhos obtidos.

- Validation research = O estudo é do tipo validation research (Analytical study, Survey, Simulation, Focus group).

  Em (DIAMANTOPOULOS; KARAGIANNOPOULOS; SYMEONIDIS, 2018), se propôs uma ferramenta que busca pedaços de código-fonte na internet. A abordagem utilizada pelo autor para valida-la foi comparar resultados obtidos por buscas feitas pela ferramenta com o motor de busca Google.

  Trabalhos como os de (YE; LO, 2000), (WANG; DAGLI, 2011), (NAKKRASAE; SOPHATSATHIT, 2004), (LI et al., 2004), (BASCIANI et al., 2016), (HEINEMANN, 2012), (WANG; LIU; FENG, 2004), (DUMITRU et al., 2011), (MCCAREY; CINNÉIDE; KUSHMERICK, 2005), (KöGEL, 2017), (SAYYAD; AMMAR;

Referência do Estudo	Recomendação
(LI et al., 2004)	Information Entropy Theory
(MCCAREY; CINNÉIDE; KUSHMERICK, 2005)	Collaborative Filtering
(Martins et al., 2009)	Association Rule
(DUMITRU et al., 2011)	Association Rule e k-Nearest-Neighbor
(HEINEMANN, 2012)	Collaborative Filtering e Association Rule
(SAYYAD; AMMAR; MENZIES, 2012)	Range Ranking
(VESCAN, 2015)	Algoritmo Evolutivo
(ELKAMEL; GZARA; BEN-ABDALLAH, 2016)	CACB
(KöGEL, 2017)	Algoritmo Evolutivo
(LE et al., 2018)	Sequential Pattern

Tabela 10 – Técnicas usadas para recomendação.

MENZIES, 2012), (BAJRACHARYA; OSSHER; LOPES, 2009), (Martins et al., 2009) e (VESCAN, 2015) aplicaram a(s) técnica(s) proposta(s) em conjuntos de *assets* e repositórios, em sua maioria de fontes da *internet*.

- Proof of concept = O estudo é do tipo *solution proposal* (Proof of concept, Conceptual demonstration).

  (LE et al., 2018) demonstra o uso da sua proposta através de um exemplo ilustrativo.

- Philosophical paper = O estudo é do tipo mapa conceitual tratando da temática da área. Nenhum estudo se enquadrou nessa classificação.

- Experience report = O estudo é do tipo relato de uso de uma determinada tecnologia para recomendação. Nenhum estudo se enquadrou nessa classificação.

- Position paper = O estudo é do tipo opinião sobre uma temática da área. Nenhum estudo se enquadrou nessa classificação.

Já nos trabalhos de (VODITHALA; PABBOJU, 2015), (BAWA; KAUR, 2017), (KUMAR; BHATIA; KUMAR, 2011) e (WU et al., 2007) não se apresentou nenhuma forma de validação nos seus artigos.

Na Figura 19, fica claro que a abordagem de validação é a mais utilizada para propostas na área de recomendação de *assets*. Esse achado é importante para fins de comparação com os estudos futuros derivados deste TCC, podendo direcionar a melhor escolha para a avaliação de nossa proposta.

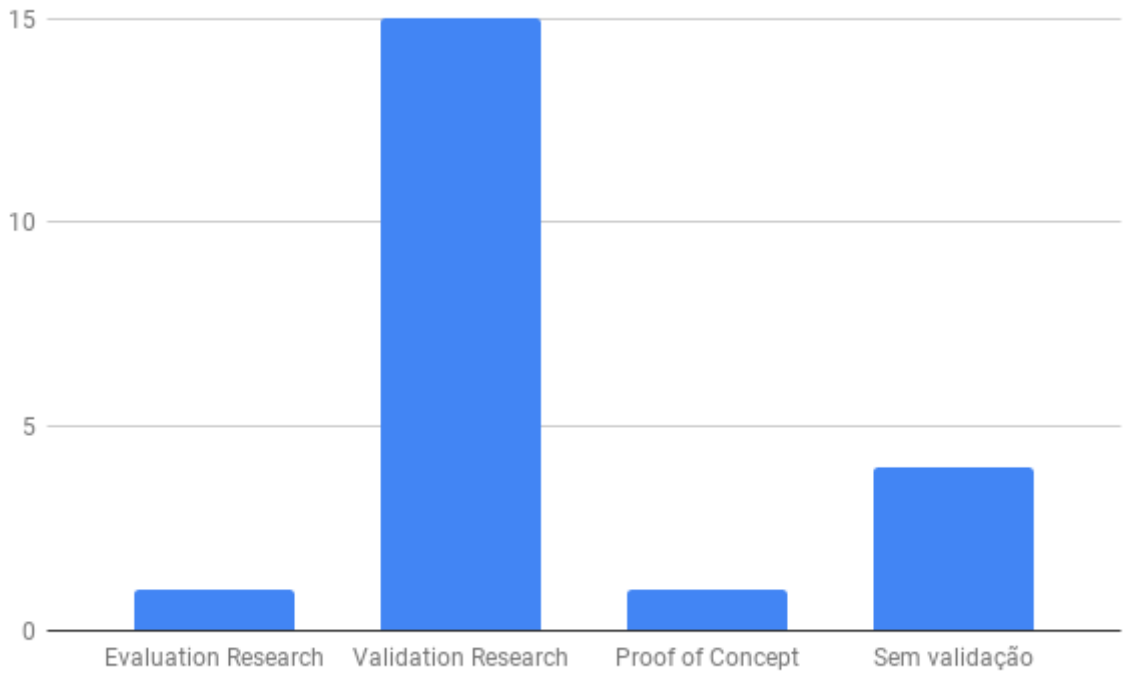

Figura 19 – Formas de validação.

4.3.6 Ameaças à Validade

Esta seção apresenta as ameaças a validade do nosso estudo, classificadas em 4 categorias, de acordo com o trabalho de (WOHLIN et al., 2012).

Validade de Construção: A categorização não foi feita de maneira sistemática, tendo sido realizada *ad hoc*.

Validade Interna: A fase de classificação pode ter sido influenciada pelo viés do autor, pelo fato da pesquisa ter sido executada somente por ele. Para tentar mitigar esse problema, mais dois autores participaram da análise e da discussão dos trabalhos.

Validade Externa: dado o número reduzido de trabalhos resultantes da busca na literatura, o resultado pode não refletir precisamente a realidade, o que levanta a possibilidade da execução de um *snowballing* no futuro com o objetivo de colher mais informação, e portanto, mitigar essa ameaça ao trabalho.

Validade da Conclusão: nós consideramos que esse estudo carece de uma análise estatística mais aprofundada, também reflexo do baixo número de resultados da busca. Assim como já citado, um futuro processo de *snowballing* pode ser levado em consideração a fim de apresentarmos uma análise mais próxima da realidade.

4.4 Considerações do Capítulo

O estudo de mapeamento sistemático foi desenhado para focar em técnicas de recomendação dedicadas para *assets* de software. Isso foi um ponto negativo para publicar os

Referência do Estudo	Ano	Técnica	Tipo de *Asset*
(YE; LO, 2000)	2000	Self-Organizing Map	Componentes
(NAKKRASAE; SOPHATSATHIT, 2004)	2004	RPCL(Neural Network)	Componentes
(LI et al., 2004)	2004	Information Entropy Theory	Componentes
(WANG; LIU; FENG, 2004)	2004	Self-Organizing Map	Componentes
(MCCAREY; CINNÉIDE; KUSHMERICK, 2005)	2005	Collaborative filtering, Content-based Filtering	Componentes (em contexto agil)
(WU et al., 2007)	2007	Ontologia, text mining	Documentos
(BAJRACHARYA; OSSHER; LOPES, 2009)	2009	Topic Model e Author Topic Model	Source Code
(Martins et al., 2009)	2009	Association Rule	Componentes
(WANG; REN, 2011)	2011	Clustering Index Tree	Componentes
(DUMITRU et al., 2011)	2011	Text Mining, K-Nearest-Neighbor, Incremental Diffusive Clustering	Domain-Specific Features
(HEINEMANN, 2012)	2012	Collaborative Filtering e Association Rules	Tool/DSL
(KUMAR; BHATIA; KUMAR, 2011)	2012	K-Means	Casos de uso em MDL (Rational Rose)
(SAYYAD; AMMAR; MENZIES, 2012)	2012	Range Ranking Method	Configuração de linha de produtos
(VODITHALA; PABBOJU, 2015)	2015	Clustering	Componente (Funções)
(VESCAN, 2015)	2016*	Algoritmo evolutivo	Componentes
(ELKAMEL; GZARA; BEN-ABDALLAH, 2016)	2016	CACB	Modelo (Classes UML)
(BASCIANI et al., 2016)	2016	Hierarchical clustering	Metamodelos
(BAWA; KAUR, 2017)	2017	Clustering	Componentes
(KöGEL, 2017)	2017	Algoritmo evolutivo	Modelos
(LE et al., 2018)	2018	Sequential pattern	Source code
(DIAMANTOPOULOS; KARAGIANNOPOULOS; SYMEONIDIS, 2018)	2018	K-Means	Source code

Tabela 11 – Técnicas aplicadas aos tipos de *assets*.

resultados em conferências e ou revistas da área, uma vez que limita bastante o escopo de aplicação das técnicas. No entanto, forneceu um conjunto de técnicas bastante aplicadas

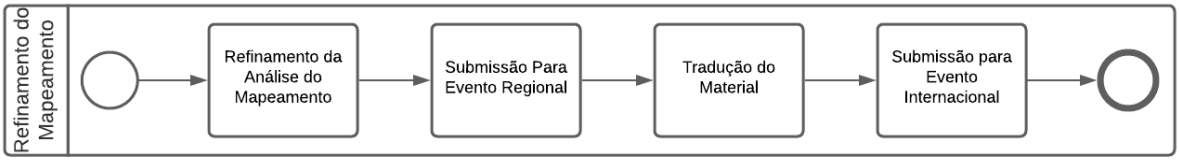

Figura 20 – Processo de Refinamento e Submissão do Mapeamento.

ao tema, o que é importante para a tomada de decisão para seleção inicial daquelas que podem contribuir de melhor forma para o problema de pesquisa investigado.

Ao longo da execução do trabalho de conclusão de curso, identificou-se a necessidade de tornar o material mais abrangente. Isso se deu nos meses que antecederam a retomada das atividades do TCC 2, em que foi conduzido um árduo processo de refinamento do material apresentado durante a defesa do TCC 1 a fim de submetê-lo para revistas/eventos, tal processo sendo apresentado na Figura 20. Como resultado deste processo obtivemos o mapeamento refinado publicado em evento regional, porém não obtivemos relevância em conferências e revistas de maiores proporções.

Como alternativa, surge a alternativa de complementar futuramente o estudo de mapeamento sistemático. O atual visa encontrar mais propostas e evidências relacionadas ao tema deste trabalho (técnicas de *data mining*) aplicadas ao reúso oportunista de *assets*. Ao mesmo tempo, este trabalho faz parte de um projeto de pesquisa mais abrangente, que inclui outras propostas relacionadas que, em conjunto, servem para um mesmo propósito: tornar o reúso oportunista de software mais eficiente com uma ferramenta de apoio para a etapa de aquisição de *assets*.

Com isso em mente, como um trabalho futuro, foi decidido reexecutar o mapeamento com alterações no protocolo, repensando o foco do mesmo de *data mining* para ser mais genérico, e consequentemente, fugindo também do escopo deste trabalho. Por isso, a suspensão das atividades relacionadas com a revisão, uma vez que desfocariam o estudo atual da sua proposta original.

5 PROPOSTA NA TEMÁTICA DE ATIVOS HÍBRIDOS DE SOFTWARE

Este capítulo apresenta a proposta de estudo numa das temáticas da área de pesquisa em reutilização de software: a reutilização com base em especificação de ativos, ou *asset specifications*. Em especial, busca-se explorar possibilidades de mineração de dados associados com um tipo específico de ferramenta, que é devotada para auxiliar em contextos de produção de software dirigido por modelos. Estas ferramentas são classificadas em repositórios de *assets* e podem ser alvo de técnicas de recomendação aplicadas em cenários de reúso oportunista, discutido no capítulo anterior por meio de um estudo de mapeamento sistemático da literatura.

5.1 Demonstração Conceitual

Este trabalho faz parte de um projeto de pesquisa que busca auxiliar o reúso oportunístico de software através da integração contínua entre *assets* durante todas as fases do desenvolvimento. Como ilustra a Figura 21, nossa pesquisa se encaixa nesse contexto através da análise de duas possíveis técnicas que poderiam ser utilizadas para a recomendação de *assets* que fossem complementares uns aos outros.

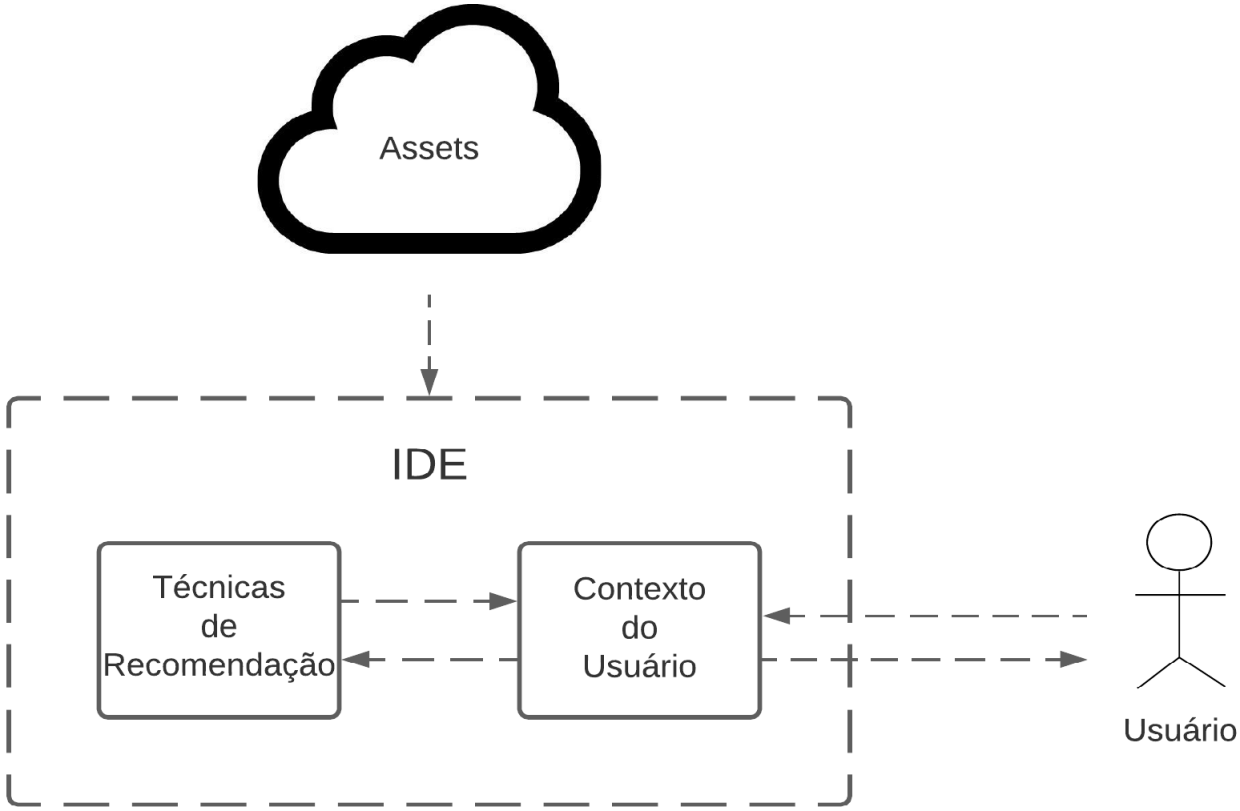

Figura 21 – Desenho arquitetural que engloba a proposta.

Os *assets* que temos até o momento estavam separados em 3 conjuntos diferentes. Com o código apresentado na Figura 22, os 3 conjuntos foram organizados em um único repositório a fim de facilitar as buscas de seu conteúdo.

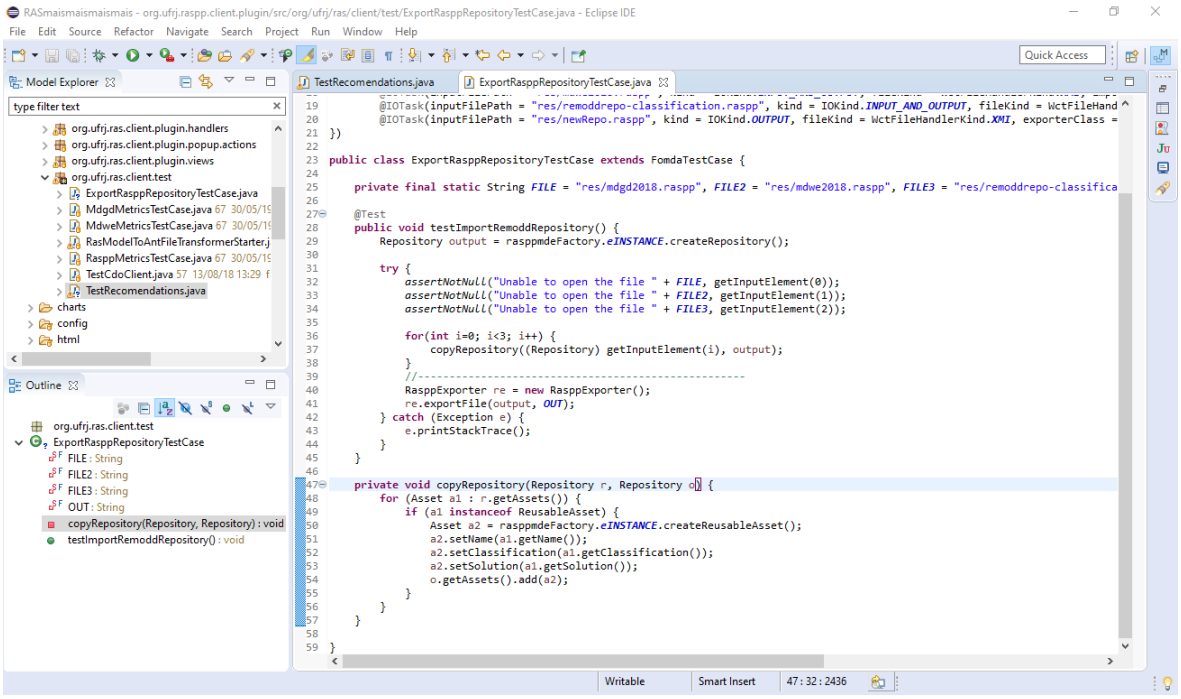

Figura 22 – Mescla dos repositórios.

Já na Figura 23, fica ilustrado como pode-se comparar dois *assets*. A semelhança entre os dados representados pela RAS++ será a chave durante a aplicação dos algoritmos.

Cada um dos grupos descritores (*Descriptor Group*) também pode influenciar no desempenho das buscas dependendo do contexto. O grupo *Required Tool* por exemplo, pode ter um peso maior numa busca por ferramentas que complementem umas as outras, assim como o grupo *Lifecycle Phases* pode ter um peso maior quando o desenvolvedor puder reaproveitar um *asset* específico para uma fase de desenvolvimento.

5.2 Preparação do Armazém de Dados

A Figura 21 apresenta um cenário aparentemente muito simples de recomendação, mas que na prática é bastante complexo e envolve a execução de um processo completo de *data mining*. A seção apresenta as atividades de *data mining* necessárias para a criação de um armazém de dados, que foram executadas após a defesa do TCC 1 e que implicaram em contribuições na preparação do ambiente experimental, bem como limitações deste estudo que serão exploradas futuramente. As atividades permitiram se chegar em dados comuns em um armazém de dados em formato RAS++, que posteriormente foram alvo de aplicação de técnicas de *Machine Learning* para recomendação.

5.2.1 Seleção dos Dados

Uma vez que estudos anteriores ofereceram 81 assets híbridos de software, fez-se leituras do Capítulo 3.3 da tese (BASSO, 2017) para identificar possíveis fontes de dados.

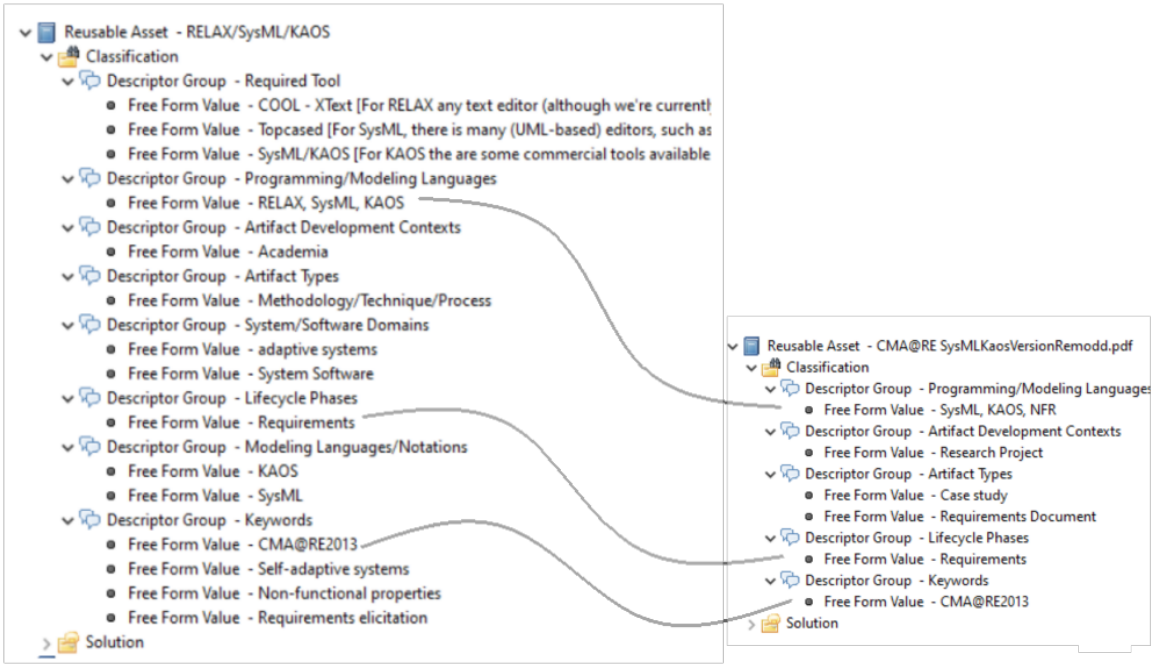

Figura 23 – Comparação entre *assets*.

Identificado que este número de *assets* poderia ser aumentado, planejou-se aumentar o corpo de dados disponíveis a fim de obter resultados com maior qualidade para a mineração e recomendação. Buscou-se manualmente os dados de cinco repositórios que tratam de informação híbrida sobre tecnologias de MDE.

Posteriormente, descobriu-se que, das cinco fontes levantadas, apenas uma era viável para mineração: o ReMoDD[1]. Outras bases como MDE Forge e SEMAT apresentam restrições quanto ao uso, o GEMOC na verdade utiliza o ReMoDD como base, a base SHARE foi descontinuada. Portanto, o repositório ReMoDD foi o único disponível com *assets* publicados sem restrições de acesso e com os devidos metadados, sendo a única opção viável como seleção cujos dados pudessem ser obtidos sem restrições.

Por fim, para viabilizar a pesquisa, delimitou-se o tipo de dado extraído para informações descritivas e representadas em estruturas como classificação. Neste ponto, vale ressaltar que a contribuição deste TCC é limitada em termos de escopo. Pelo menos outros cinco critérios de representação da informação, que levam à boas tomadas de decisões pelas fábricas de software, não são consideradas (BASSO; WERNER; OLIVEIRA, 2017a). Tais critérios necessitam da identificação de padrões de dados para viabilizar a extração, e, portanto, não são atacados por este trabalho.

5.2.2 Extração de Dados

O enriquecimento da base de *assets* se deu de modo automático, diferente do estudo (BASSO, 2017) que aplicou a extração manual. Por conta da falta de metadados

[1] <https://www.cs.colostate.edu/remodd/v1/>

em outras fontes, não foi possível aplicar a extração de dados através de *web crawler* para as bases SHARE, GEMOC, SEMAT e MDE Forge.

Dado que um objetivo futuro no grupo de pesquisa é o de automatizar o processo de extração e armazém de dados, este estudo apresenta uma contribuição importante: uma possibilidade da obtenção automática de *assets* através de *web crawlers*. Trabalhos futuros podem explorar outras alternativas e vasculhar repositórios variados, extrair e representar tais *assets* com a linguagem RAS++ e dar seguimento para este TCC.

Para desenvolvimento do *web crawler* foi utilizado o *framework* Scrapy[2]. O repositório com o código gerado se encontra em <https://github.com/torresrafa22/remodd-crawler>.

5.2.3 Transformação de Dados

Uma vez que o web crawler retorna um conjunto de assets em formato XML que segue o padrão do Scrapy, e que o formato do armazém de dados é em RAS++, uma transformação entre os formatos foi necessária. Esta transformação foi feita de modo automática, com o desenvolvimento e utilização de um algoritmo de *parser*.

5.2.4 Limpeza dos Dados

Além da transformação de dados, o estudo em (BASSO, 2017) identificou inconsistências entre informações do repositório ReMoDD. Uma delas diz respeito à ambiguidade de termos utilizados em *free form values*, que são os alvos das técnicas de aprendizado de máquina propostos. Assim, a limpeza dos dados é necessária para desambiguizar os dados.

A Figura 24 mostra um algoritmo dedicado a relatar os dados extraídos. Ele é construído em uma estrutura de suporte ao processo de mineração de dados, atualmente restrito a auxiliar Engenheiros de Software nas tarefas `Classificação` e `Clustering`. Na linha 11 é ilustrado o método sobrescrito `getSynonyms()`, permitindo um agrupamento de dados conforme ambiguidades identificadas. O método sobrescrito mostrado na Linha 45 formata os grupos de descritores para uma fonte de dados específica, ou seja, para os metadados do repositório ReMoDD.

Isso significa que a intenção da estrutura é permitir a personalização para realizar um formato de metadados específico encontrado em qualquer provedor de ativos. Finalmente, o método sobrescrito mostrado na Linha 56 caracteriza os `tipos` como valores de forma livre encontrados em repositórios baseados em RAS e AMS/OSLC. Isso permitirá representar valores para elementos técnicos associados a artefatos, independentemente de uma abordagem de megamodelo.

Apesar da existência desta *feature*, que serve para desambiguizar os termos, este TCC não considerou a limpeza dos dados por meio dela. Isso porque a ferramenta utiliza

[2] <https://scrapy.org>

```
5  public class RemoddRasppMetrics extends AbstractRasppMetrics {
6
7      /**
8       * The set of grouped synonyms found in free form values
9       */
10     private static ArrayList<Synonym> synonyms = null;
11     protected ArrayList<Synonym> getSynonyms() {
12         if (synonyms == null) {
13             synonyms = new ArrayList<Synonym>();
14             Synonym sl = new Synonym("UML",
15                     new String[] { "UML", "UML2", "UML 2" });
16             synonyms.add(sl);
17             sl = new Synonym("Embedded Systems",
18                     new String[] { "Embedded Software",
19                             "Real-time embedded systems",
20                             "Embedded Systems" });
21             synonyms.add(sl);
22             sl = new Synonym("Information Systems",
23                     new String[] { "Information Systems",
24                             "Business/Information Systems",
25                             "Enterprise Systems" });
26             synonyms.add(sl);
27             sl = new Synonym("Crisis Management",
28                     new String[] { "Crisis Management Systems",
29                             "Crisis Management",
30                             "Crisis Management System" });
31             synonyms.add(sl);
32         }
33         return synonyms;
34     }
35
36     /**
37      * The set of descriptor groups analyzed
38      */
39     private static final String[] GROUP_ARGS =
40             new String[] { "Artifact Development Contexts",
41             "Artifact Types", "System/Software Domains",
42             "Lifecycle Phases", "Modeling Languages/Notations",
43             "Required Tool", "Programming/Modeling Languages"};
44     @Override
45     protected String[] getDescriptorGroupNames() {
46         return GROUP_ARGS;
47     }
48
49     /**
50      * The set of artifact types analyzed
51      */
52     private static final String[] TYPE_ARGS =
53             new String[] { "OCL Script", "Metamodel", "Model",
54                     "Ecore Diagram" };
55     @Override
56     protected String[] getTypes() {
57         return TYPE_ARGS;
58     }
```

Figura 24 – Captura de tela de uma métrica de personalização da estrutura descritiva para limpeza de dados derivados da extração automática do ReMoDD.

dos agrupamentos em sinônimos com a finalidade de gerar gráficos de barra para quantificar a base conforme grupos descritores e valores. Assim, um trabalho futuro pretende introduzir *features* para realizar esta limpeza, assistindo o processo por meio de suporte ferramental adaptado.

5.2.5 Armazenamento de Dados

Os dados são armazenados em formato XMI, seguindo o padrão Ecore (STEINBERG et al., 2008). Portanto, trata-se de um armazém de dados orientado à modelos. Isto permite com que ferramentas como CDO[3] e EMFStore[4], que tratam da persistência automática de modelos em associação com Sistemas Gerenciadores de Banco de Dados (SGBD) relacionais, armazenem automaticamente estas informações em um banco de dados.

5.2.6 Analisar e Minerar

Nesta atividade é onde está o foco de estudo deste TCC. Ou seja, prover ao aparato ferramental algoritmos de *data mining* que viabilizem a recomendação de *assets*. Esta contribuição é detalhada no próximo capítulo.

5.2.7 Visualização dos Dados

Por fim, os dados precisam ser analisados utilizando ferramentas de análise estatística para a tomada de decisão. Este trabalho também explorou algumas alternativas, porém aplicadas na análise dos dados experimentais, também apresentadas no próximo capítulo.

5.3 Planejamento Experimental

Nesta seção é apresentado os dados gerais que são comuns para o planejamento dos três experimentos, que focam na análise e mineração de dados do ReMoDD.

5.3.1 Objetivo

O principal objetivo deste estudo é comparar duas técnicas das identificadas a partir do mapeamento descrito no Capítulo 4. Para tal, dividiu-se os objetivos em metas específicas: 1) Desenvolver e testar o algoritmo K-Medoid; 2) Desenvolver e testar o algoritmo genético, e 3) Comparar os resultados num estudo de agrupamento.

[3] https://www.eclipse.org/cdo/documentation/
[4] https://www.eclipse.org/emfstore/

5.3.2 Seleção de *Dataset*

O experimento é executado utilizando um conjunto de *assets*, manualmente representados a partir da informação contida no repositório ReMoDD (81) [5], MDGD (10) e MDWE (38). Estes repositórios foram minerados, resultando em *assets* representados utilizando a RAS++ DSL, como em (BASSO, 2017).

5.3.3 Variáveis Independentes

Variáveis independentes são passíveis de serem controladas e mudadas durantes o experimento, além de ter influência sobre as variáveis independentes.

No contexto deste experimento, podemos apontar como variáveis independentes a o conjunto de *assets* em que as técnicas serão aplicadas e a quantidade de iterações na execução de cada.

5.3.4 Variáveis Dependentes

Variáveis dependentes são variáveis que dependem das variáveis independentes, e que não podem ser manipuladas diretamente.

Experimentos envolvendo busca de informação comumente utilizam Precisão e Sensibilidade, mais conhecidos pelos termos em inglês *Precision* e *Recall*, respectivamente.

Em uma recomendação feita por um algoritmo, *Precision* representa quantos itens dentro do conjunto de resultados são de fato relevantes (verdadeiros positivos). Já o *Recall* representa quantos itens relevantes retornaram da busca, como a quantidade de falsos negativos e verdadeiros positivos.

Em termos gerais, pode-se representar essas métricas como:

$$Precision = \frac{Verdadeiro\ Positivo}{Verdadeiro\ Positivo\ + Falso\ Positivo} \tag{5.1}$$

$$Recall = \frac{Verdadeiro\ Positivo}{Verdadeiro\ Positivo\ + Falso\ Negativo} \tag{5.2}$$

Utilizando o contexto de *assets*, pode-se exemplificar o uso dessas métricas da seguinte maneira: suponha que uma busca para recomendar um *asset* obteve 10 resultados de um repositório com 100 *assets*. Desses 10, 8 eram verdadeiros positivos, enquanto os outros 2 eram falso positivos. Nesse caso, a **precisão** seria de 8/10. Imaginemos também que a quantidade total de verdadeiros positivos dentro do repositório seria de 20, com isso, a **sensibilidade** seria 8/20.

Quanto às técnicas selecionadas, adotou-se duas por serem comparáveis e bastante utilizadas em cenários semelhantes ao motivado neste estudo:

1. *K-Medoids*, que agrupa *assets* de acordo com sua inter-semelhança.

[5] <https://www.cs.colostate.edu/remodd/v1/>

2. Algoritmo Genético, que assim como *K-Medoids*, agrupa *assets* com características parecidas, porém baseado em probabilidade.

A técnica *Association Rule*, que gera sugestões baseadas na relação apresentada entre *assets* baseada em histórico, foi cogitada no TCC 1, porém foi descartada na execução do TCC 2 pela falta de dados que viabilizassem a sua aplicação.

Devido à natureza dos dados utilizados (*assets* representados através de DSL) e ao método de comparação elaborado, optamos pela técnica de *K-Medoids* no lugar da *K-Means*. As duas possuem a mesma aplicação, porém, *K-Medoids* se diferencia por utilizar um objeto do *cluster* como centro, enquanto que *K-Means* utiliza a média dos objetos como centro.

5.3.5 Configuração de Hardware

O experimento será conduzido em uma máquina com processador Intel Core i5-4210U de 1.70GHz e 8GB de memória RAM.

Também foi utilizada uma maquina virtual no ambiente AWS.

5.3.6 Análise de Possíveis Ameaças

- Como o teste de hipótese é baseado em análise estatística e probabilística, o tamanho da amostra utilizada tem grande impacto nos resultados e no nível de confiança do experimento.
  Plano de mitigação: o procedimento para diminuir essa ameaça é obter um corpo de amostras maior e reexecutar o experimento.

- A natureza estocástica de alguns algoritmos e meta-heurísticas podem influenciar no seu resultado e eficiência em uma amostra reduzida.
  Plano de mitigação: múltiplas execuções de algoritmo com as mesmas configurações tende a revelar os resultados mais constantes da execução, permitindo uma análise mais precisa.

5.3.7 Ferramentas de Análise Estatística

- Dado o objetivo de apresentar os resultados de um trabalho quantitativo, serão utilizados gráficos que facilitem a avaliação e comparação dos resultados. Os softwares utilizados são: a biblioteca Matplotlib para linguagem Python e o ambiente RStudio para a linguagem R.

- Para os testes de hipótese: ambiente RStudio para linguagem R.

Para o teste de hipóteses é adotada a técnica do teste-t (*t-test*)(WOHLIN et al., 2012).

6 EXECUÇÃO DOS EXPERIMENTOS

Este documento apresenta três experimentos controlados. O primeiro aplica o programa de *Machine Learning* desenvolvido para suportar o agrupamento de dados conforme o algoritmo genético. O segundo aplica o programa desenvolvido para o algoritmo de agrupamento *K-Medoids*. O terceiro experimento foi executado com o objetivo de comparar as duas técnicas de *Machine Learning*, a fim de verificar qual traria melhores resultados num contexto de recomendação de *assets*.

6.1 Configuração Experimental Comum aos Três Estudos

As técnicas escolhidas, Algoritmo Genético e `K-Medoids`, foram implementadas e testadas utilizando a linguagem `Python`, versão 3.8.5 e o editor de código `Visual Studio Code`.

6.1.1 Ameaças à Validade

- O tamanho do corpo de *assets* que possuíamos foi identificado como uma ameaça ao valor estatístico do experimento. Para isso, pretendia-se aumentar a quantidade de *assets* disponíveis para o experimento. No entanto, não foram encontradas base de *assets* adicionais disponíveis publicamente além do `ReMoDD`. Portanto, esta limitação de bases pode ter um impacto negativo no experimento.

- Os algoritmos foram desenvolvidos pelo autor do estudo e, portanto, mesmo que tenham sido testados, eles ainda são passíveis de *bugs* não identificados, que por sua vez podem afetar os resultados obtidos.

- O experimento foi conduzido por um pesquisador apenas, o que gera a possibilidade do viés experimental da visão do pesquisador. Tal viés, portanto, pode afetar os resultados.

6.1.2 *Assets* de Referência

A fim de calcular os valores de Precisão e *Recall*, é necessário primeiramente que o pesquisador identifique os elementos que serão usados como referência para o estudo. Foram selecionados, portanto, 17 *assets* que possuíam uma clara similaridade em relação ao seu conteúdo, sendo ela a palavra-chave `bCMS` (CAPOZUCCA BETTY H.C. CHENG, 2012).

A similaridade entre *assets* foi calculada somando a quantidade de atributos semelhantes que os *assets* tem em comum. Portanto, dado que tenhamos um conjunto de atributos A pertencente a um *asset* e o conjunto de atributos B pertencente a um segundo *asset*, a similaridade entre eles será o tamanho do conjunto $A \cap B$. Com isso, acredita-se

que *assets* relacionados terão uma quantidade maior de atributos semelhantes e consequentemente cada *cluster* conterá *assets* que possuam alguma relação de proximidade semântica entre si.

O *asset* detalhado na Tabela 12 foi escolhido de maneira arbitrária para servir de referência na escolha do *cluster* de onde serão obtidas as métricas de Precisão e *Recall*. Como resultado de uma busca por recomendações que complementem o *asset* de referência, espera-se que os resultados agrupados através do mesmo *cluster* sejam as recomendações retornadas ao consumidor do *asset*, pois seriam os dados semanticamente mais próximos.

Para maiores informações, o Apêndice A apresenta o restante dos 16 *assets* relacionados.

Modeling Specification for bCMS Product Line using Feature Model, Component Family Model and UML
Required Tool: Pure::Variants Rational Software Architect (RSA)
Programming/Modeling Languages: Feature model Component Family Model UML
Artifact Development Contexts: Academia Workshop/Focus Group
Artifact Types: Case study Methodology/Technique/Process Model
System/Software Domains: Software Product Line Crisis Management Systems
Lifecycle Phases: Validation/Verification/Analysis
Modeling Languages/Notations: UML Feature Model Component Family Model
Keywords: CMA@MODELS2013 bCMS product line feature model component family model UML

Tabela 12 – Dados do *asset* utilizado como referência no experimento.

6.1.3 Configuração dos Algoritmos

Uma vez que se busca identificar a técnica que traz melhores resultados, e que tais resultados são determinados pela qualidade dos *clusters* de *assets* em relação ao *asset* de referência, a comparação das técnicas pela execução dos algoritmos de recomendação selecionados ocorreu pela utilização dos seguintes parâmetros:

- A quantidade de *clusters*: variar a quantidade é importante porque influencia diretamente no resultado. Um conjunto de elementos apresenta um numero x de relações passíveis de formar subgrupos coerentes e o mais homogêneos possível. Tentar separar um conjunto de maçãs e laranjas por exemplo, em 3 grupos diferentes, resultaria em um grupo vazio, ou heterogêneo, e que portanto não apresenta nenhum padrão a ser observado. Portanto, optou-se por variar a quantidade de *clusters* de 5 à 15 a fim de buscar o número mais próximo do ideal para a amostra utilizada no experimento.
- Execução: Durante a execução, houve um impasse ocasionado pela demora da execução de alguns algoritmos devido ao funcionamento do algoritmo genético. O experimento foi executado simultaneamente em duas máquinas, um hardware local executando os algoritmos com 5 à 13 clusters, e uma máquina *Virtual Machine* do serviço AWS da Amazon para executar de maneira síncrona os algoritmos com 15 clusters, tendo levado alguns dias para finalizar a execução com 15 *clusters*.

6.2 Primeiro Experimento

O primeiro experimento buscou avaliar a aplicação da técnica *K-Medoids* no conjunto de *assets* representados pela DSL RAS++. Os dados utilizados aqui são referentes aos resultados obtidos utilizando 5 *clusters*. O algoritmo da técnica *K-Medoids* é apresentado no Pseudo-código 1.

Algorithm 1: Algoritmo K-Medoids

Input: Assets, $K \leftarrow NumerodeClusters$
Output: Clusters
$clusters \leftarrow alocarAssetsAleatoriamente(K)$
$centroidsMudaram \leftarrow True$
while *centroidsMudaram* **do**
    $clusters \leftarrow realocarAssets(Assets)$
    $centroidsMudaram \leftarrow recalcularCentroids(clusters)$
end
return *clusters*

6.2.1 Formulação das Hipóteses

1. H_1: Para o conjunto de *assets* estudado, o algoritmo de *K-Medoids* permite obter dados de Precisão maior que 50%.

2. H_2: Para o conjunto de *assets* estudado, o algoritmo de *K-Medoids* permite obter dados de *Recall* maior que 50%.

As hipóteses descritas podem ser apresentadas como segue:

1. Hipótese nula 1, H_1: O algoritmo testado não apresenta Precisão média maior que 50%.
   $H_0 : \mu_p < 50\%$.
   Hipótese alternativa, H_1: $\mu_p \geq 50\%$
   Medidas necessárias: Precisão da seleção de *assets* (5.1).

2. Hipótese nula 2, H_2: O algoritmo testado não apresenta *Recall* médio maior que 50%.
   $H_0 : \mu_p < 50\%$.
   Hipótese alternativa, H_1: $\mu_p \geq 50\%$
   Medidas necessárias: *recall* da seleção de *assets* (5.1). Medidas necessárias: *recall* na seleção de *assets* (5.2).

6.2.2 Questões de Pesquisa

Q.1: O algoritmo *K-Medoids* é capaz de recomendar com eficiência artefatos no contexto de reúso através de estruturas descritivas de *assets*?

6.2.3 Análise de Resultados

A Figura 25 mostra a variação da precisão do algoritmo *K-Medoids* com 5 *clusters* observados durante o experimento. Pode-se perceber que os valores tendem a variar quase que de maneira aleatória, um comportamento que não era esperado. Tal comportamento também foi observado quando o algoritmo foi executado com um número diferente de *clusters*. Acreditamos que essa anomalia nos resultados é resultado da maneira em que a similaridade entre os *assets* foi calculada.

Como explicado no Capitulo 5 com a Figura 23, calculamos a diferença entre *assets* contando quantos atributos semelhantes eles possuem entre si. Com isso, *assets* que não possuem nenhuma ou pouca semelhança entre si, mas que contém atributos semelhantes à um terceiro *asset*, que no momento é computado pelo atributo *centroid* do *cluster*, acabam sendo incluídos no mesmo conjunto.

Outra característica observada dessa abordagem foi o conjunto limitado de valores resultantes da função de semelhança. Como resultado observado, percebe-se que *assets* não mudam de *clusters*. Isso ocorre por motivos como: 1) A similaridade com mais de um *centroid* é igual; 2) consequentemente o algoritmo apresenta poucas iterações antes de encerrar 3) o algoritmo é fortemente impactado pela distribuição inicial de *assets* nos *clusters*, e 4) que, por sua vez, é percebida pelos pesquisadores como um comportamento

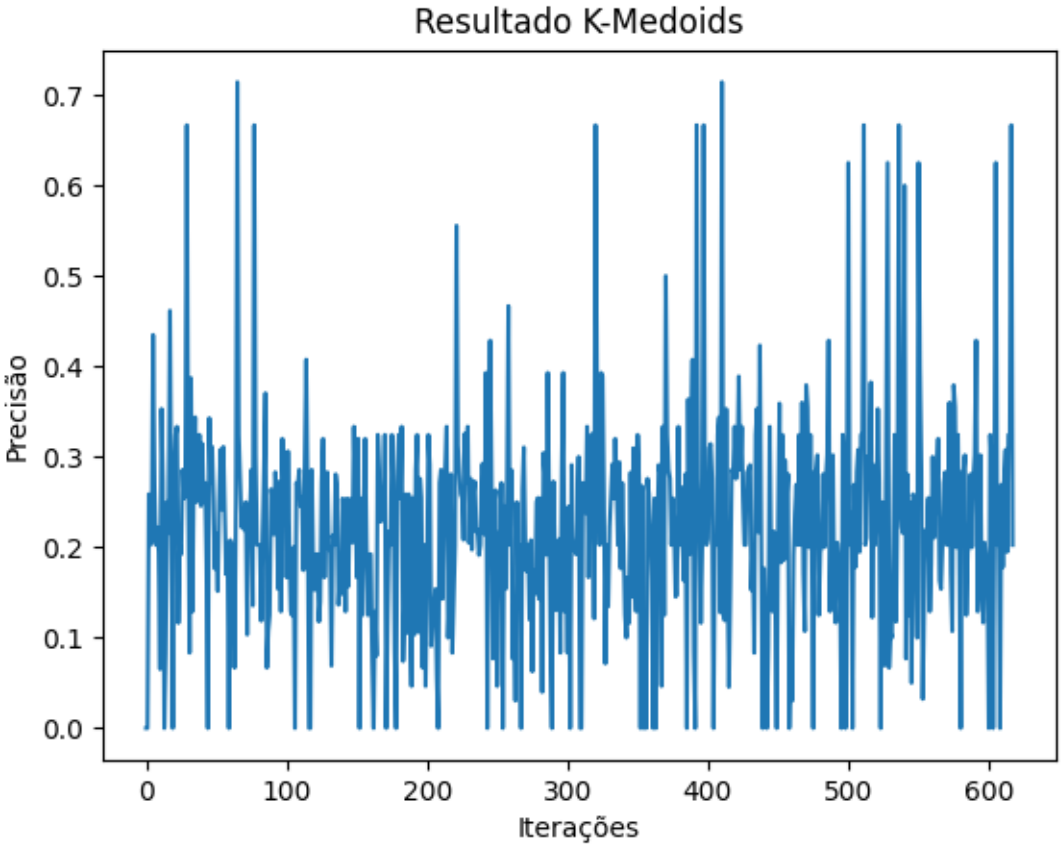

Figura 25 – Precisão do algoritmo *K-Medoids* com 5 *clusters*

de recomendação de maneira aleatória, mesmo que o *K-Medoids* seja uma das técnicas que tende à reduzir a aleatoriedade.

Para responder a questão de pesquisa Q.1, testamos as hipóteses apresentadas anteriormente utilizando *t-test*. Este é um teste parametrizado em que se computa informações sobre *t-value* e *p-value*, o que permite gerar um gráfico de distribuição para os resultados.

Como resultado para a primeira hipótese, que avalia quantos itens dentro de um conjunto são de fato relevantes, obtive-se o seguinte resultado: **t = -71.339** e **p-value = 1**. Tais resultados foram concebidos com um intervalo de confiança de 95%. Portando, não se obteve dados suficientes para descartar a hipótese nula. Além disso, tendo o valor p, também como um indicativo de que os valores podem ter sido obtidos por acaso, observa-se aqui mais um indício da razão do comportamento indesejado descrito anteriormente.

Para a segunda hipótese, obtive-se um resultado de: **t = 10.621** e ***p-value* < 2.2e-16**, usando-se um intervalo de confiança de 95%. Portanto, pode-se descartar a hipótese nula. Aqui observa-se indícios de que o *recall* do algoritmo de *K-Medoids* tende a ser maior do que 50%. Portanto, indica um comportamento não aleatório para retornar itens relevantes pela recomendação implementada pelo *K-Medoids*.

6.3 Segundo Experimento

O segundo experimento avaliou a aplicação do algoritmo genético, mostrado no Pseudo-código 2, no conjunto de *assets* representados pela DSL RAS++. Os dados utilizados nesta seção foram obtidos com o experimento configurado para 5 *clusters*.

Algorithm 2: Algoritmo Genético

```
Input: Assets, K ← NumerodeClusters,
       tamanhoPopulacao ← TamanhoDaPopulação,
       nIteracoes ← NumerodeIterações
Output: Clusters
populacao ← criarPopulacao(tamanhoPopulacao, K, Assets)
matrizSimilaridade ←
 criarMatrizSimilaridade(Assets, funcaoDeSimilaridade())
for iter = 1 to nIteracoes do
    proximaGeracao ← Empty
    fitnessIndividuos ← avaliarIndividuos(populacao, matrizSimilaridade)
    for x = 1 to populacao do
        pai ← torneio(populacao, fitnessIndividuos)
        mae ← torneio(populacao, fitnessIndividuos)
        novoIndividuo ← crossover(pai, mae)
        proximaGeracao[x] ← novoIndividuo
    end
    populacao ← proximaGeracao
end
return populacao
```

6.3.1 Formulação das Hipóteses

1. H_1: Para o conjunto de *assets* estudado, o algoritmo de genético permite obter dados de *Precision* maior que 50%.

2. H_2: Para o conjunto de *assets* estudado, o algoritmo genético permite obter dados de *Recall* maior que 50%.

As hipóteses descritas podem ser apresentadas como segue:

1. Hipótese nula 1, H_1: O algoritmo testado não apresenta Precisão média maior que 50%.
   $H_0 : \mu_p < 50\%$.
   Hipótese alternativa, H_1: $\mu_p \geq 50\%$
   Medidas necessárias: Precisão da seleção de *assets* (5.1).

2. Hipótese nula 2, H_2: O algoritmo testado não apresenta *Recall* médio maior que 50%.

$H_0 : \mu_p < 50\%$.
Hipótese alternativa, H_1: $\mu_p \geq 50\%$
Medidas necessárias: *recall* da seleção de *assets* (5.1).

6.3.2 Questões de Pesquisa

Q.2: O algoritmo genético é capaz de recomendar com eficiência artefatos no contexto de reúso através de estruturas descritivas de *assets*?

6.3.3 Análise de Resultados

Para a hipótese 1: o algoritmo testado não apresenta *Precision* média maior que 50%, observou-se os seguintes resultados:: $\mathbf{t = -61.661}$ e $\boldsymbol{p\text{-}value = 1}$. O estudo considerou um intervalo de confiança de 95%. Portanto, não se obteve dados suficientes para descartar a hipótese nula, o que significa um resultado ruim para o quesito relevância dos resultados dentro do cluster.

Para a hipótese 2: o algoritmo genético permite obter dados de *Recall* maior que 50%, observou-se os seguintes resultados:: $\mathbf{t = -3.7925}$ e $\boldsymbol{p\text{-}value = 0.9999}$. O estudo também considera um intervalo de confiança de 95%. Portanto, também não se pode descartar a hipótese nula, o que significa que existe aleatoriedade quanto ao quesito itens relevantes retornados na recomendação.

6.4 Terceiro Experimento

O objetivo do terceiro experimento foi de comparar os resultados dos dois primeiros estudos para identificar aquele que é mais recomendável ao cenário motivado. Assim como os dois experimentos anteriores, este também foi realizado utilizando os dados relativos à 5 *clusters*.

6.4.1 Formulação das Hipóteses

A hipótese formulada, descrita mais abaixo, objetiva verificar e comparar o desempenho dos métodos para recomendação de *assets*. A recomendação ocorre após a realização de uma busca, que retorna uma seleção preliminar de *assets*. Assim, parte-se da suposição de que a técnicas escolhidas, *K-Medoids* e *Algoritmo Genético*, possuem desempenho distinto no contexto de recomendação de *assets* de *software* armazenados no repositório ReMoDD, sendo assim uma técnica mais adequada do que a outra.

1. H_1: Os algoritmos testados apresentam diferença na medida de *Precision* média (μ_p).

2. H_2: Os algoritmos testados apresentam diferença na medida de *Recall* média (μ_r).

Formalmente, a hipótese descrita pode ser apresentada como segue:

1. Hipótese nula 1, H_1: Os algoritmos testados não apresentam diferença na medida de Precisão média (μ_p).
   $H_0 : \mu_{p_1} = \mu_{p_2}$.
   Hipótese alternativa, H_1: $\mu_{p_1} \neq \mu_{p_2}$
   Medidas necessárias: Precisão da seleção de *assets* (5.1).

2. Hipótese nula 2, H_2: Os algoritmos testados não apresentam diferença na medida de *Recall* média (μ_r).
   $H_0 : \mu_{r_1} = \mu_{r_2}$.
   Hipótese alternativa, H_2: $\mu_{r_1} \neq \mu_{r_2}$
   Medidas necessárias: *Recall* resultante na seleção de *assets* (5.2).

6.4.2 Questões de Pesquisa

Com base nestas hipóteses, busca responder as seguintes questões de pesquisa:

- Questão de Pesquisa 3 (Q3): Qual é a técnica mais adequada para ser adotada em repositórios de *assets* construídos na estrutura da informação descritiva em formato RAS++?

6.4.3 Achados do Estudo

Para a Hipótese 1: Os algoritmos testados não apresentam diferença na medida de precisão média. Com um intervalo de confiança de 95%, observou-se os seguintes resultados: $\mathbf{t = -5.0644}$ e $\boldsymbol{p\text{-}value = 4.731e\text{-}07}$. Com isto, pode-se descartar a hipótese nula de que os algoritmos apresentam a mesma eficácia dentro do contexto do experimento, e portanto pode-se concluir sobre qual é a melhor alternativa quanto ao quesito de retorno de resultados relevantes dentro dos *clusters*: o *Algoritmo Genético*.

Para a Hipótese 2: Os algoritmos testados apresentam diferença na medida de *Recall* média. Também usando um intervalo de confiança de 95%, observou-se os seguintes resultados: $\mathbf{t = 11.02}$ e $\boldsymbol{p\text{-}value < 2.2e\text{-}16}$. Consequentemente, também pode-se descartar a hipótese de que os algoritmos apresentam medidas de *Recall* similares, e portanto pode-se concluir sobre qual é a melhor alternativa quanto ao quesito de itens relevantes dentro dos *clusters*: o *K-Medoids*.

A aplicação dos algoritmos para o *asset* de referência mostrado na Tabela 12 resultou em gráficos de *box-plot* apresentados nas Figuras 26 e 27. Os gráficos apresentam na base dados sobre a similaridade média das medidas de intra-*cluster*, precisão, e *recall*. Estas foram determinadas conforme o número de *clusters* gerados. A Figura 26 apresenta uma análise dos resultados obtidos após a execução do algoritmo em *cluster K-Medoids*, enquanto que a Figura 27 demonstra os resultados para o *cluster* Algoritmo Genético.

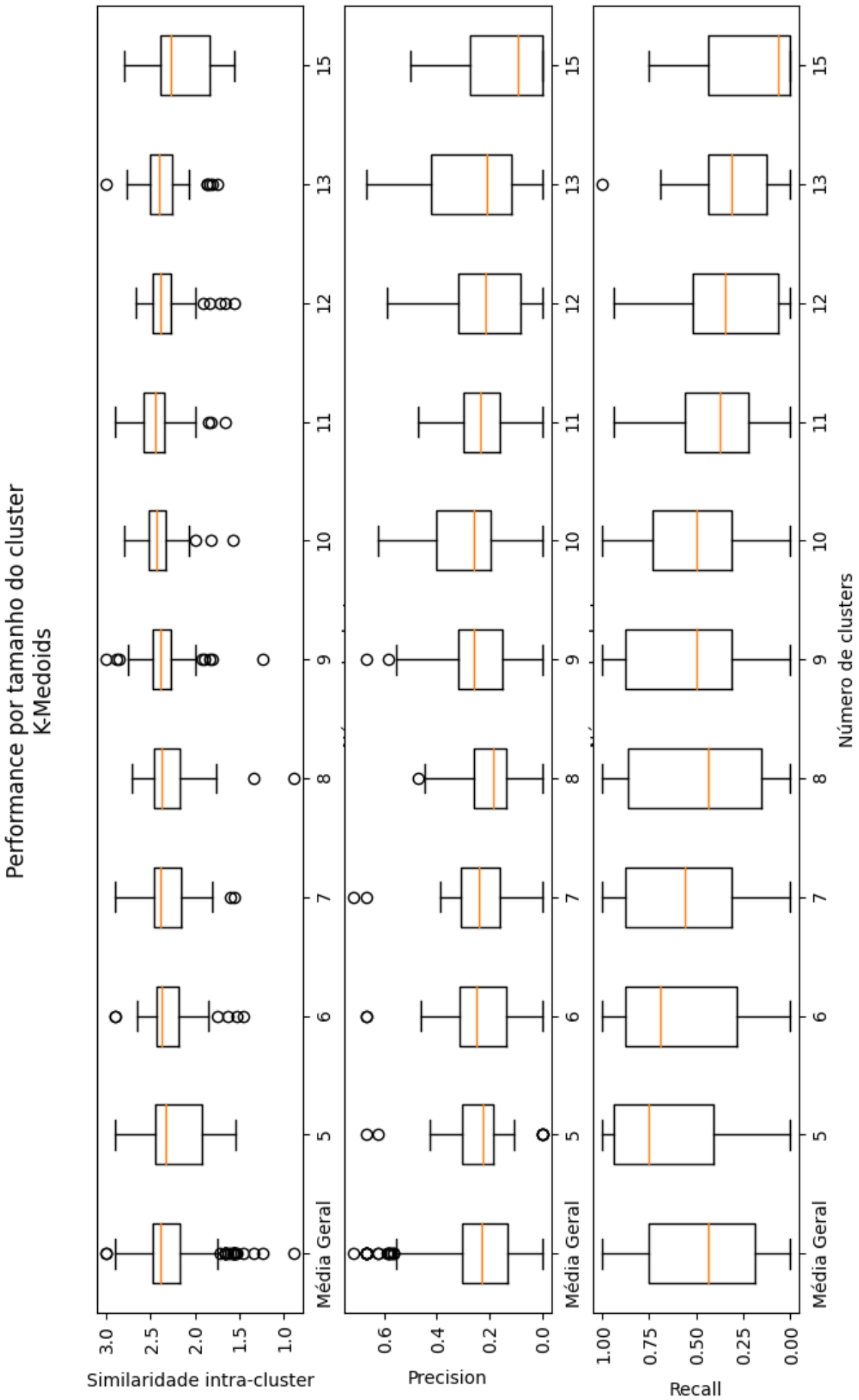

Figura 26 – Análise de resultados por *cluster* K-Medoids

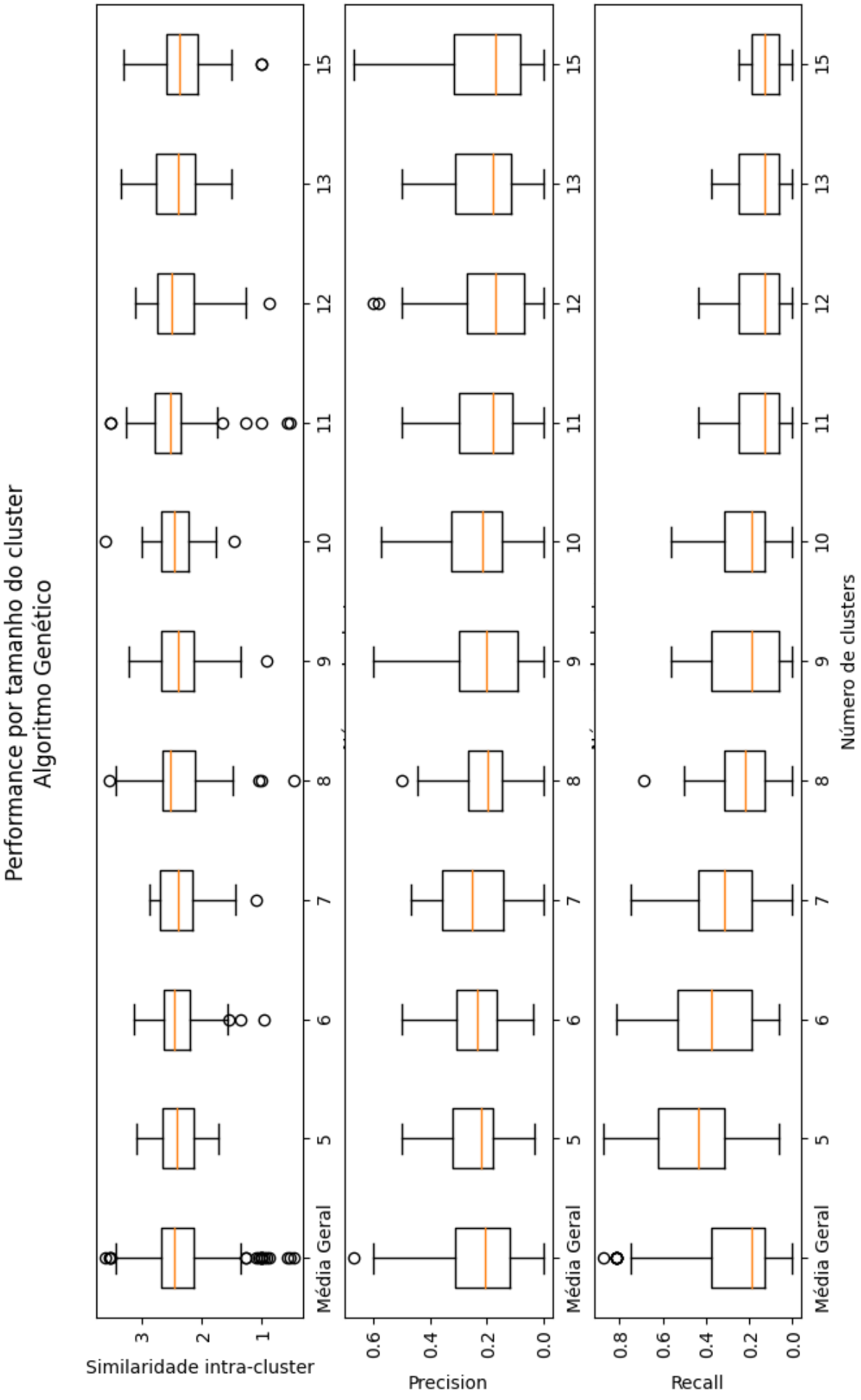

Figura 27 – Análise de resultados por *cluster* Algoritmo Genético

6.4.4 Análise e Comparação dos Resultados

Para o experimento, foram feitas 618 iterações com 5 *clusters*, 416 com 6 *clusters* e 352 com 7 *clusters*. Foram extraídos do experimento as informações listadas na Tabela 13.

Clusters	Quantidade de *Clusters*
Tamanho do *Cluster*	Quantidade de itens dentro do *cluster*
Similaridade *Intra-Cluster*	Similaridade total entre os itens dentro do *cluster*
Similaridade Média *Intra-Cluster*	Similaridade média entre os itens do *cluster*
Precisão	Precisão do algoritmo ao agrupar os *assets* relacionados à bCMS
Recall	*Recall* do algoritmo ao agrupar os *assets* relacionados à bCMS

Tabela 13 – Dados extraídos do experimento.

As figuras descritas a seguir apresentam os resultados comparando-os entre as duas técnicas citadas. Na Figura 28, é apresentada a comparação entre a similaridade intra-*cluster*. Como pode-se observar, a mediana das 3 situações diferentes é maior com o algoritmo genético, indicando maior eficiência em agrupar os *assets* similares.

Já a Figura 29 compara a precisão dos dois algoritmos, ou seja, quantos dos elementos dentro do *cluster* são relevantes. Assim como a similaridade, a precisão é maior com o algoritmo genético, porém, nesse caso o algoritmo de *K-Medoids* apresenta *outliers* acima do limite superior, indicando que ainda pode apresentar resultados melhores que o algoritmo genético em algumas poucas iterações.

Finalmente, a Figura 30 apresenta a comparação entre o *Recall* resultante, ou seja, a razão entre quantos *assets* relevantes foram selecionados e quantos *assets* relevantes existiam no total. Diferente dos gráficos de similaridade média e precisão, o *recall* resultante do algoritmo de `K-Medoids` teve resultados acima dos do algoritmo genético.

A partir da quantidade 9 de *clusters*, `K-Medoids` começa a apresentar valores de precisão superiores ao algoritmo genético, como mostra a Figura 31. A partir daí, quanto maior a quantidade de *clusters*, mais o algoritmo `K-Medoids` se destacava em relação ao algoritmo genético.

6.5 Q3: Qual é a técnica mais adequada para ser adotada em repositórios de *assets* construídos no formato RAS++?

A partir da análise dos dados e dos resultados dos testes de hipótese, percebe-se que os resultados do algoritmo *K-Medoids* não refletem a real capacidade do algoritmo de associar *assets* corretamente dentro de grupos coerentes. Isso ocorre, provavelmente e levando em consideração que o algoritmo está implementado corretamente, devido ao método de comparação utilizado não ter a precisão necessária para a recomendação.

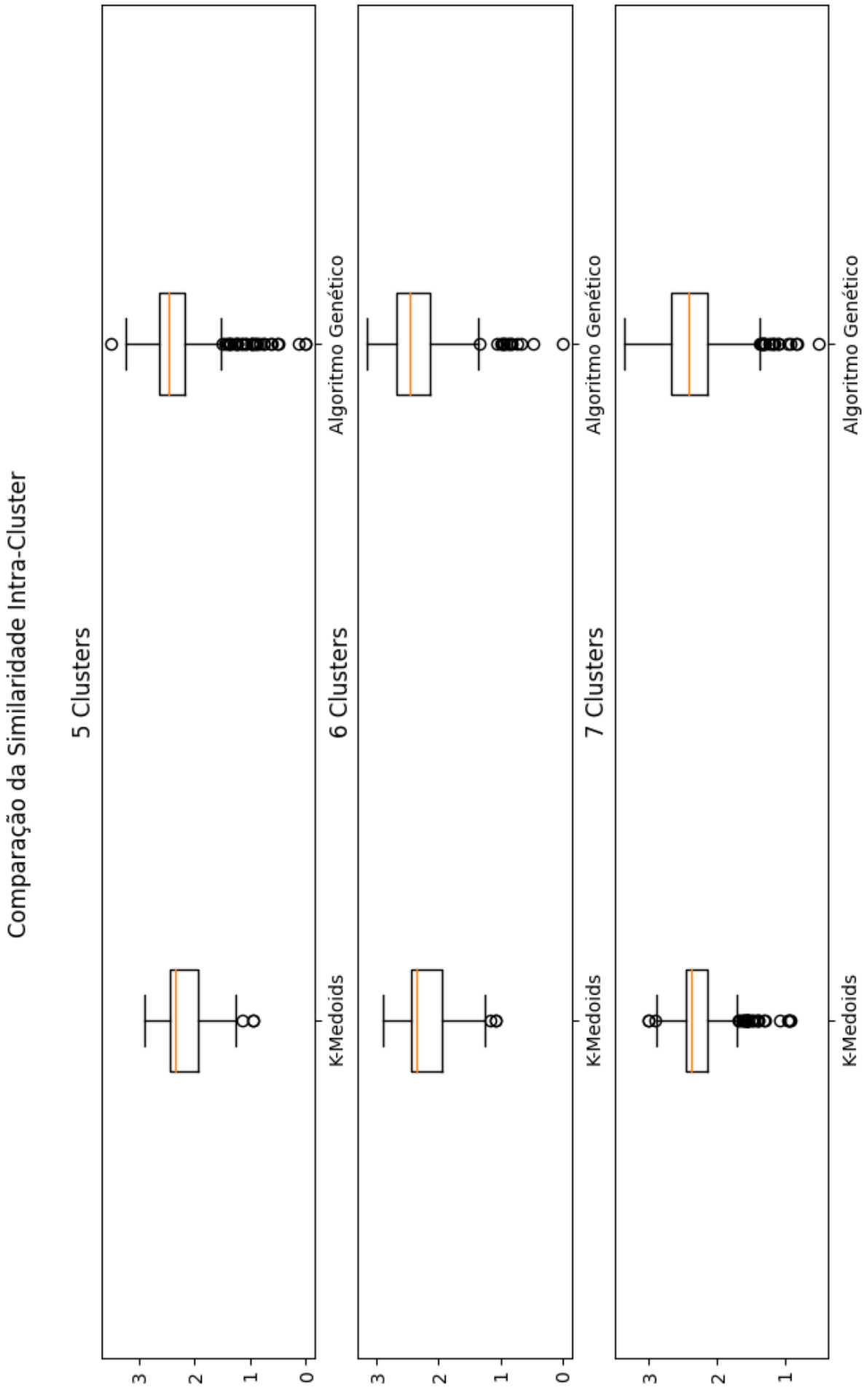

Figura 28 – Comparação da Similaridade Intra-*Cluster*

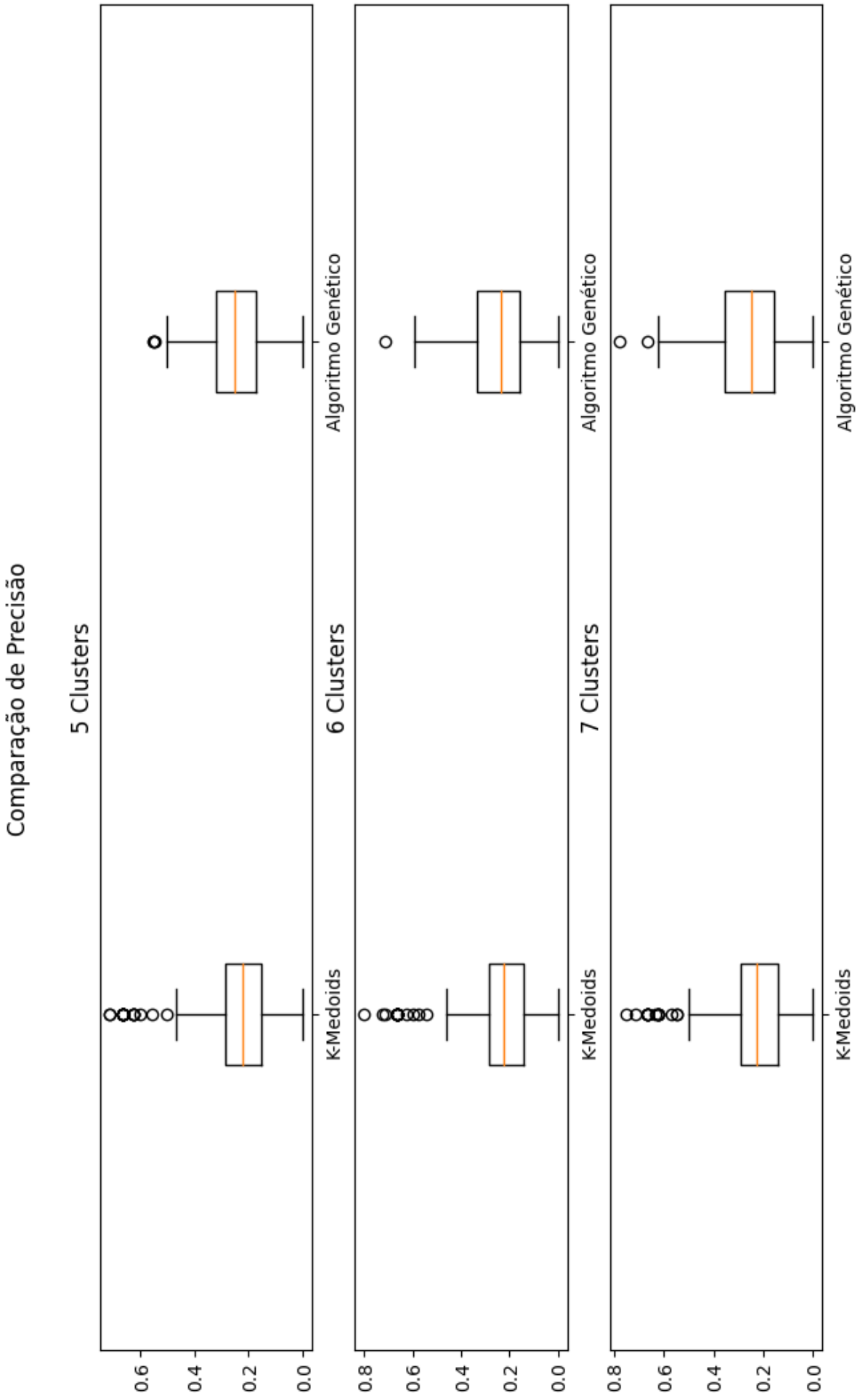

Figura 29 – Comparação de Precisão

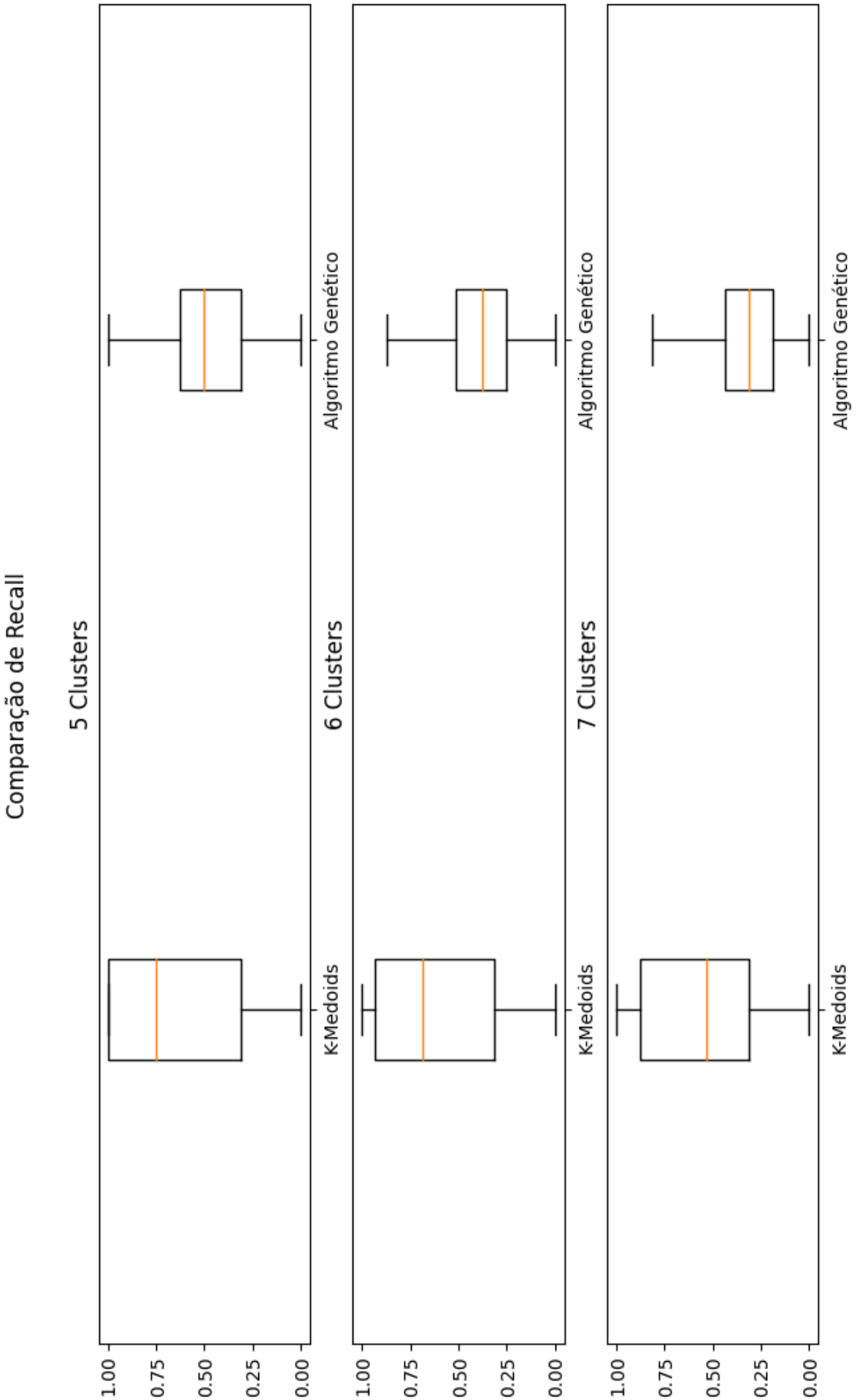

Figura 30 – Comparação de *Recall*

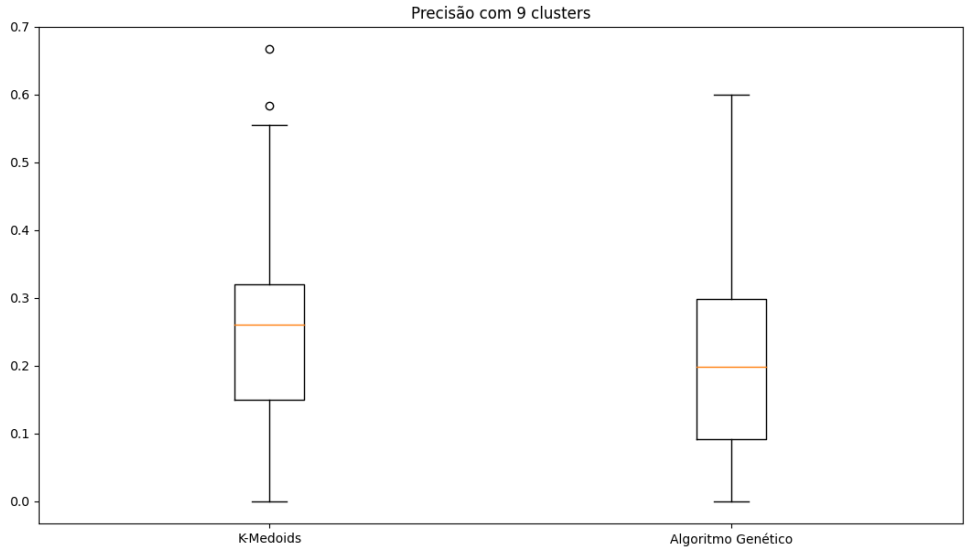

Figura 31 – `K-Medoids` passa a apresentar maior precisão à partir dos 9 *clusters*.

Portanto, fica inconclusiva a eficácia do algoritmo *K-Medoids*. Diferentemente, o algoritmo genético apresentou resultados coerentes nos testes feitos pós experimento. Mesmo que o método de comparação utilizado não tenha a eficácia desejada, isso indica que a sua eficácia também pode ser maior com a otimização do método utilizado em trabalhos futuros. Porém, ainda temos como inconclusiva qual abordagem seria superior.

6.6 Comparação com os Trabalhos Relacionados

Nesta seção comparamos nossa abordagem com trabalhos relacionados.

Primeiramente, por termos uma camada de representação (RASS++) que permite aplicar as técnicas escolhidas a qualquer tipo de *asset*, entendemos que nosso trabalho é mais generalizável do que trabalhos que tem um tipo específico de *asset* como objeto de estudo.

Com isso, vale ressaltar que os trabalhos (DIAMANTOPOULOS; KARAGIANNOPOULOS; SYMEONIDIS, 2018) e (KUMAR; BHATIA; KUMAR, 2011), que aplicam *K-Means*, no contexto de código-fonte e casos de uso no formato MDL. Estes provavelmente possuem implementações da técnica citada que diferem da nossa, assim como os trabalhos de (KöGEL, 2017), que aplica uma técnica híbrida se aproveitando dos conceitos de algoritmo genético que utilizamos.

Três focos distintos de estudos podem ser observados em relação aos trabalhos relacionados apresentados na Tabela 14: 1) utilizam técnicas de agrupamento e organização de repositórios de ativos; e 2) usam técnicas de filtragem e mineração de dados para recomendar ativos que podem ser úteis dependendo do contexto de desenvolvimento, e; 3) alguns trabalhos também buscam recursos em ambientes web.

A Tabela 15 mostra a intenção das plataformas de ativos, investigadas no estudo

Estudo	Título	Ano
S01	Codecatch: extracting source code snippets from online sources	2018
S02	An Old Problem with a New Therapy: Coupling Topic Modeling and Mining Sequential Patterns in Recommending Source Code	2018
S03	Recommender System for Model Driven Software Development	2017
S04	K-Means Clustering of Use-Cases Using MDL	2017
S05	Classification and clustering for efficient storage and retrieval of component repositories	2017
S06	Automated clustering of metamodel repositories	2016
S07	An UML class recommender system for software design	2016
S08	A clustering technique based on the specifications of software components	2015
S09	An evolutionary multiobjective approach for the dynamic multilevel component selection problem	2015
S10	Facilitating reuse in model-based development with context-dependent model element recommendations	2012
S11	Software Feature Model Recommendations Using Data Mining	2012
S12	A Component Clustering Index Tree Based on Semantic	2011
S13	On-demand Feature Recommendations Derived from Mining Public Product Descriptions	2011
S14	Sourcerer: An Internet-scale Software Repository	2009
S15	Suggesting Software Components for Reuse in Search Engines Using Discovered Knowledge Techniques	2009
S16	Construction of Ontology-Based Software Repositories by Text Mining	2007
S17	Rascal: A Recommender Agent for Agile Reuse	2005
S18	An RPCL-based indexing approach for software component classification	2004
S19	Attribute ranking: An entropy-based approach to accelerating browsing-based component retrieval	2004
S20	Improved SOM Clustering for Software Component Catalogue	2004
S21	A visualised software library: nested self-organising maps for retrieving and browsing reusable software assets	2000

Tabela 14 – Estudos selecionados como propostas para plataformas de ativos.

de mapeamento sistemático, em termos do tipo de ativo disponível para a mineração.

6.7 Trabalhos Futuros

Nesta seção, discutimos as inovações potenciais que podem ajudar a progredir no estado da arte das plataformas de ativos. Conforme mostrado na Tabela 16, as soluções atuais usam uma das seguintes fases para aquisição de ativos: pesquisa, organização e recomendação.

Uma vez que se observou a dificuldade em encontrar *assets* em bases abertas, trabalhos futuros podem explorar: 1) maneiras de fomentar tais repositórios com dados reais; 2) formas de recuperar os dados compartilhados e integrá-los em abordagens de design oportunistas orientadas à processos de *data mining*; 3) utilizar os *assets* representados pelas técnicas de extração e transformação com o objetivo de automatizar os processos de desenvolvimento de software; 4) adaptar as técnicas de mineração de dados para explorar

Estudos Mapeados	**Tipos de *Assets***	**Cobertura do Nosso Estudo**
S05 S08 S09 S12 S15 S17 S18 S19 S20 S21	Component	Contempla
S16	Document	Contempla
S13	Hybrid	Contempla
S06	Metamodel	Parcialmente - Necessita de aplicação
S03 S04 S07	Model	Parcialmente - Necessita de aplicação
S01 S02 S11 S14	Source Code	Parcialmente - Necessita de aplicação
S10	Tool	Parcialmente - Necessita de aplicação

Tabela 15 – Estudos por tipo de ativo que mostram a heterogeneidade do cenário motivado, considerando as necessidades de reúso oportunista por meio de *data mining*.

Estudo	**Busca**	**Organização**	**Recomendação**
S21		X	
S20		X	
S19			X
S18		X	
S17			X
S16		X	
S15			X
S14	X	X	
S13	X		X
S12		X	
S11			X
S10			X
S09			X
S08		X	
S07		X	X
S06		X	
S05		X	
S04		X	
S03			X
S02			X
S01	X	X	
Cobertura do nosso estudo	X	X	X

Tabela 16 – Cobertura dos estudos para fases identificadas para recuperação de ativos, ordenadas de acordo com o mais antigo para o mais novo.

as fases de Busca, Organização e Recomendação com dados compartilhados por portfólios de produtos de software de *web pages* de fábricas de software, meio de estilos de integração *Lookup* (*spiders*, *crawlers*, etc); 5) conectar todas as informações heterogêneas entre eles através da federação de ativos e artefatos, em uma nuvem de nuvens ou sistemas de sistemas de informação; e 6) para fins de obter indícios de viabilidade por meio de comparação, desenvolver novos mecanismos de recomendação que considerem blocos de cooperação para ativos distribuídos usando uma variedade de algoritmos ilustrados na

Estudos	**Técnica de Recomendação**	**Cobertura do Nosso Estudo**
S02	Sequential Pattern	Não Contempla
S03 S09	Evolutionary Algorithm	Contempla
S07	CACB	Não Contempla
S10 S13 S15	Association Rule	Não Contempla
S10 S17	Collaborative Filtering	Não Contempla
S11	Range Ranking	Não Contempla
S13	k-Nearest-Neighbor	Contempla
S19	Ranking Tree	Não Contempla

Tabela 17 – Técnicas de recomendação adotadas por estudos selecionados.

Tabela 17.

Também é de suma importância que para futuros experimentos a otimização do método de comparação, buscando alternativas como modelos matemáticos que representem com mais precisão a dissimilaridade entre *assets* e/ou explorar técnicas de enriquecimento semântico de *assets*.

7 CONSIDERAÇÕES FINAIS

O principal objetivo deste trabalho é apresentar um estudo exploratório sobre como a área de *Data Mining* auxilia o reúso oportunista de *assets*. Ou seja, tem como objetivo oferecer uma forma de apoio ao processo decisório para a aquisição de tecnologias de MDE. Nele, apresentamos um mapeamento sistemático de literatura que resultou em indícios da necessidade de estudos quantitativos e comparativos em relação às técnicas aplicadas.

A partir desse mapeamento, foi possível também identificar técnicas que foram posteriormente testadas no contexto de recomendação de *assets* utilizando o suporte da DSL RAS++. As técnicas consideradas inicialmente para comparação foram *K-Medoids*, *Algoritmo Genético* e *Association Rule*. Esta, porém, requer um histórico de relação entre os objetos de estudo, o que teria que ser inventado pelos próprios autores do estudo uma vez que não se dispõe desses dados. Além de não ser algo positivo para o experimento ter os objetos de estudo manipulados pelos autores, a estratégia de gerar informações foge completamente do escopo do trabalho, e portanto, optamos por abandonar a técnica *Association Rule*.

O experimento demonstrou resultados insatisfatórios em relação as abordagens utilizadas para agrupamento de *assets*. Isso se deve principalmente devido ao método de comparação discutido na seção que relata os achados do estudo. Portanto, para estudos futuros temos a forma de avaliação da similaridade entre os *assets* como principal foco. A maneira que foi procedido durante o experimento se mostrou ineficiente, porém foi um primeiro passo no processo de aprimorar e evoluir os procedimentos de recomendação. Com isso em mente, há a possibilidade de experimentar outras maneiras que as técnicas possam ser aplicadas, bem como evoluir a DSL RAS++, caso esta se mostre uma alternativa válida. Por fim, também espera-se que outras técnicas sejam experimentadas a fim de integra-las à um ambiente de desenvolvimento.

As dificuldades que foram identificadas durante o desenvolvimento mostram também a falta de repositórios públicos/abertos de *assets*, o que dificulta a execução de estudos exploratórios como este. Buscou-se explorar outros quatro repositórios além do `ReMoDD`, porém eles ou estão inativos, ou possuem restrições de acesso. Ao mesmo tempo que isso pode incentivar a criação de um novo repositório por parte do grupo de pesquisa, também sugere que existem desafios a serem investigados para que um novo repositório não tenha o mesmo destino que os anteriores. Nossa conclusão neste sentido é que faltam dados de qualidade, que justamente por não terem qualidade apresentam ruídos na clusterização.

Tendo em vista que os próximos passos do grupo de pesquisa estão relacionados à criação de um repositório de *assets* com dados que possuam maior valor do que os oferecidos pelo ReMoDD, assim como uma possível fonte de uma abordagem para data *warehouse*, um próximo passo será tratar sobre o enriquecimento semântico dos *assets* do

ReMoDD, como o feito manualmente em (BASSO, 2017). Com *assets* de melhor qualidade semântica, espera-se re-executar os estudos experimentais para identificar se a clusterização nos algoritmos *K-Medoids* e Algoritmo Genético apresentam algum comportamento diferente do observado.

Por fim, este TCC buscou explorar um tópico relevante para a pesquisa em reúso oportunista, uma área importante da Engenharia de Software que vem tendo barreiras de entrada para a adoção. Cabe salientar que até o momento tal barreira não é completamente entendida na literatura da área. Portanto, estudos exploratórios, como os apresentados neste TCC, são fundamentais para compreender tais barreiras e se delinear melhorias em processos e práticas de reutilização.

REFERÊNCIAS

AGRAWAL, R.; IMIELIńSKI, T.; SWAMI, A. Mining association rules between sets of items in large databases. **SIGMOD Rec.**, ACM, New York, NY, USA, v. 22, n. 2, p. 207–216, jun. 1993. ISSN 0163-5808. Disponível em: <http://doi.acm.org/10.1145/170036.170072>. Citado na página 36.

ANDERSSON, P.; HST, M. Uml and systemc - a comparison and mapping rules for automatic code generation. In: **Embedded Systems Specification and Design Languages**. [S.l.: s.n.], 2008. v. 10, p. 199–209. Citado na página 19.

ASSET Management Specification. Av. at <http://open-services.net/wiki/asset-management/OSLC-Asset-Management-2.0-Specification/>. 2014. Citado na página 23.

BAJRACHARYA, S.; OSSHER, J.; LOPES, C. Sourcerer: An internet-scale software repository. In: **Proceedings of the 2009 ICSE Workshop on Search-Driven Development-Users, Infrastructure, Tools and Evaluation**. Washington, DC, USA: IEEE Computer Society, 2009. (SUITE '09), p. 1–4. ISBN 978-1-4244-3740-5. Disponível em: <https://doi.org/10.1109/SUITE.2009.5070010>. Citado 4 vezes nas páginas 46, 48, 49 e 51.

BASCIANI, F. et al. Automated clustering of metamodel repositories. In: SPRINGER. **International Conference on Advanced Information Systems Engineering**. [S.l.], 2016. p. 342–358. Citado 3 vezes nas páginas 46, 48 e 51.

BASSO, F. P. **RAS++: Representing Hybrid Reuse Assets for MDE as a Service. Av at <www.cos.ufrj.br/uploadfile/publicacao/2811.pdf>**. Tese (Doutorado), September 2017. Citado 13 vezes nas páginas 20, 21, 23, 25, 27, 29, 30, 54, 55, 56, 59, 80 e 97.

BASSO, F. P. et al. Building the foundations for 'mde as service'. **IET Software**, v. 11, p. 195–206(11), August 2017. Citado 2 vezes nas páginas 25 e 97.

BASSO, F. P. et al. Supporting large scale model transformation reuse. In: **12th International Conference on Generative Programming: Concepts & Experiences.** [S.l.: s.n.], 2013. (GPCE'13), p. 169–178. Citado 2 vezes nas páginas 19 e 20.

BASSO, F. P.; WERNER, C. M. L.; OLIVEIRA, T. C. de. Automated approach for asset integration in eclipse IDE. In: **2017 IEEE/ACM Joint 5th International Workshop on Software Engineering for Systems-of-Systems and 11th Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems, JSOSICSE, Buenos Aires, Argentina, May 23, 2017**. [S.l.: s.n.], 2017. p. 34–40. Citado 5 vezes nas páginas 20, 24, 28, 55 e 97.

BASSO, F. P.; WERNER, C. M. L.; OLIVEIRA, T. C. de. Revisiting criteria for description of MDE artifacts. In: **2017 IEEE/ACM Joint 5th International Workshop on Software Engineering for Systems-of-Systems and 11th Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems, JSOSICSE, Buenos Aires, Argentina, May 23, 2017**. [S.l.: s.n.], 2017. p. 27–33. Citado 3 vezes nas páginas 23, 25 e 26.

BATORY, D.; LATIMER, E.; AZANZA, M. Teaching model driven engineering from a relational database perspective. In: **16th International Conference on Model Driven Engineering Languages and Systems**. [S.l.: s.n.], 2013. (MODELS'13), p. 121–137. Citado 2 vezes nas páginas 19 e 20.

BAWA, R. K.; KAUR, I. Classification and clustering for efficient storage and retrieval of component repositories. In: IEEE. **2017 International Conference on Computer Communication and Informatics (ICCCI)**. [S.l.], 2017. p. 1–6. Citado 4 vezes nas páginas 46, 48, 49 e 51.

BECKER, L.; HOLTZ, R.; PEREIRA, C. On mapping rt-uml specifications to rt-java api: bridging the gap. In: **Object-Oriented Real-Time Distributed Computing, 2002. (ISORC 2002).** [S.l.: s.n.], 2002. p. 348–355. Citado na página 19.

B'EZIVIN, J.; JOUAULT, F.; VALDURIEZ, P. On the need for megamodels. 10 2004. Citado na página 23.

BOEHM, B. A view of 20th and 21st century software engineering. In: **28th International Conference on Software Engineering**. [S.l.: s.n.], 2006. (ICSE '06), p. 12–29. Citado na página 20.

CALDIERA, V. R. B. G.; ROMBACH, H. D. The goal question metric approach. **Encyclopedia of software engineering**, p. 528–532, 1994. Citado na página 39.

CAPOZUCCA BETTY H.C. CHENG, G. G. N. G. P. I. G. M. A. **REQUIREMENTS DEFINITION DOCUMENT FOR A SOFTWARE PRODUCT LINE OF CAR CRASH MANAGEMENT SYSTEMS**. 2012. Disponível em: <http://cserg0.site.uottawa.ca/cma2012/CaseStudy.pdf>. Citado na página 61.

CHEN, L.; BABAR, M. A.; ZHANG, H. Towards an evidence-based understanding of electronic data sources. 2010. Citado na página 40.

DIAMANTOPOULOS, T.; KARAGIANNOPOULOS, G.; SYMEONIDIS, A. Codecatch: extracting source code snippets from online sources. In: IEEE. **2018 IEEE/ACM 6th International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering (RAISE)**. [S.l.], 2018. p. 21–27. Citado 4 vezes nas páginas 46, 48, 51 e 75.

DUMITRU, H. et al. On-demand feature recommendations derived from mining public product descriptions. In: **Proceedings of the 33rd International Conference on Software Engineering**. New York, NY, USA: ACM, 2011. (ICSE '11), p. 181–190. ISBN 978-1-4503-0445-0. Disponível em: <http://doi.acm.org/10.1145/1985793.1985819>. Citado 4 vezes nas páginas 46, 48, 49 e 51.

ELKAMEL, A.; GZARA, M.; BEN-ABDALLAH, H. An uml class recommender system for software design. In: IEEE. **2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)**. [S.l.], 2016. p. 1–8. Citado 4 vezes nas páginas 46, 48, 49 e 51.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. From data mining to knowledge discovery in databases. **AI magazine**, v. 17, n. 3, p. 37–37, 1996. Citado 2 vezes nas páginas 31 e 33.

FERREIRA, B. M. S. **Investigando a Integração de Ferramentas com OSLC no Contexto do Desenvolvimento de Software.** 2020. Monografia (Bacharel em Engenharia de Software), Unipampa (Universidade Federal do Pampa), Alegrete, Brazil. Citado na página 25.

FOWLER, M. **Domain Specific Languages**. 1st. ed. [S.l.]: Addison-Wesley Professional, 2010. ISBN 0321712943, 9780321712943. Citado na página 36.

FUGGETTA, A.; NITTO, E. D. Software process. In: **36th International Conference on Software Engineering**. [S.l.: s.n.], 2014. (ICSE '14), p. 1–12. Citado na página 20.

HEBIG, R.; BENDRAOU, R. On the need to study the impact of model driven engineering on software processes. In: **2014 International Conference on Software and System Process**. [S.l.: s.n.], 2014. (ICSSP 2014), p. 164–168. Citado 2 vezes nas páginas 19 e 20.

HEINEMANN, L. Facilitating reuse in model-based development with context-dependent model element recommendations. In: IEEE PRESS. **Proceedings of the Third International Workshop on Recommendation Systems for Software Engineering**. [S.l.], 2012. p. 16–20. Citado 4 vezes nas páginas 46, 48, 49 e 51.

KITCHENHAM, B.; LINKMAN, S.; LAW, D. Desmet: a methodology for evaluating software engineering methods and tools. **Computing Control Engineering Journal**, v. 8, n. 3, p. 120–126, June 1997. Citado na página 46.

KöGEL, S. Recommender system for model driven software development. In: **Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering**. New York, NY, USA: ACM, 2017. (ESEC/FSE 2017), p. 1026–1029. ISBN 978-1-4503-5105-8. Disponível em: <http://doi.acm.org/10.1145/3106237.3119874>. Citado 5 vezes nas páginas 46, 48, 49, 51 e 75.

KUMAR, S.; BHATIA, R. K.; KUMAR, R. K-means clustering of use-cases using mdl. In: SPRINGER. **International Conference on Computing and Communication Systems**. [S.l.], 2011. p. 57–67. Citado 5 vezes nas páginas 46, 48, 49, 51 e 75.

LAFI, L.; HAMMOUDI, S.; FEKI, J. Metamodel matching techniques in mda: Challenge, issues and comparison. In: **Model and Data Engineering**. [S.l.]: Springer Berlin Heidelberg, 2011, (Lecture Notes in Computer Science, v. 6918). p. 278–286. Citado na página 20.

LE, D.-T. et al. An old problem with a new therapy: Coupling topic modeling and mining sequential patterns in recommending source code. In: IEEE. **2018 International Conference on Advanced Computing and Applications (ACOMP)**. [S.l.], 2018. p. 117–124. Citado 4 vezes nas páginas 46, 48, 49 e 51.

LI, G. et al. Attribute ranking: An entropy-based approach to accelerating browsing-based component retrieval. In: SPRINGER. **International Conference on Software Reuse**. [S.l.], 2004. p. 232–241. Citado 4 vezes nas páginas 46, 48, 49 e 51.

LIEBEL, G. et al. Assessing the state-of-practice of model-based engineering in the embedded systems domain. In: **Model-Driven Engineering Languages and Systems**. [S.l.: s.n.], 2014. (MODELS'14), p. 166–182. Citado 2 vezes nas páginas 19 e 20.

Martins, A. C. et al. Suggesting software components for reuse in search engines using discovered knowledge techniques. In: **2009 35th Euromicro Conference on Software Engineering and Advanced Applications**. [S.l.: s.n.], 2009. p. 412–419. ISSN 1089-6503. Citado 4 vezes nas páginas 46, 48, 49 e 51.

MCCAREY, F.; CINNÉIDE, M. Ó.; KUSHMERICK, N. Rascal: A recommender agent for agile reuse. **Artificial Intelligence Review**, v. 24, n. 3, p. 253–276, Nov 2005. ISSN 1573-7462. Disponível em: <https://doi.org/10.1007/s10462-005-9012-8>. Citado 4 vezes nas páginas 46, 48, 49 e 51.

MOHAGHEGHI, P. et al. Where does model-driven engineering help? experiences from three industrial cases. **Software & Systems Modeling**, v. 12, n. 3, p. 619–639, july 2013. ISSN 1619-1374. Citado na página 19.

NAKKRASAE, S.; SOPHATSATHIT, P. An rpcl-based indexing approach for software component classification. **International Journal of Software Engineering and Knowledge Engineering**, World Scientific, v. 14, n. 05, p. 497–518, 2004. Citado 3 vezes nas páginas 46, 48 e 51.

NETO, V. V. G. et al. Model-driven engineering ecosystems. In: **Proceedings of the 7th International Workshop on Software Engineering for Systems-of-Systems and 13th Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems**. [S.l.]: IEEE Press, 2019. (SESoS-WDES '19). Citado na página 25.

OMG. **RAS Reusable Asset Specification Version 2.2 November 2005. At September 2017.** [S.l.], 2005. Disponível em: <http://www.omg.org/spec/RAS/>. Citado na página 23.

OSLC Asset Management 2.0 Specification. 2012. Disponível em: <https://archive.open-services.net/wiki/asset-management/OSLC-Asset-Management-2.0-Specification/index.html#Asset>. Citado na página 34.

PEFFERS, K. et al. A design science research methodology for information systems research. **J. Manage. Inf. Syst.**, v. 24, n. 3, p. 45–77, dez. 2007. ISSN 0742-1222. Citado 2 vezes nas páginas 23 e 97.

PREGUNTA, L. A estratégia pico para a construção da pergunta de pesquisa e busca de evidências. **Rev Latino-am Enfermagem**, SciELO Brasil, v. 15, n. 3, 2007. Citado na página 40.

ReMoDD Av. at <www.cs.colostate.edu/remodd/v1/>. At Sept. 2014. 2014. Disponível em: <http://www.cs.colostate.edu/remodd/v1/>. Citado na página 23.

ROCCO, J. D. et al. Using ATL transformation services in the mdeforge collaborative modeling platform. In: **Theory and Practice of Model Transformations - 9th International Conference, ICMT 2016, Held as Part of STAF 2016, Vienna, Austria, July 4-5, 2016**. [S.l.: s.n.], 2016. p. 70–78. Citado 3 vezes nas páginas 20, 24 e 25.

SAMETINGER, J. **Software engineering with reusable components**. [S.l.]: Springer Science & Business Media, 1997. Citado na página 19.

SAYYAD, A. S.; AMMAR, H.; MENZIES, T. Software feature model recommendations using data mining. In: **Proceedings of the Third International Workshop on Recommendation Systems for Software Engineering**. Piscataway, NJ, USA: IEEE Press, 2012. (RSSE '12), p. 47–51. ISBN 978-1-4673-1759-7. Disponível em: <http://dl.acm.org/citation.cfm?id=2666719.2666730>. Citado 4 vezes nas páginas 46, 48, 49 e 51.

SOMMERVILLE, I. Software engineering 9th edition. **ISBN-10**, v. 137035152, 2011. Citado na página 19.

SOUZA, V. E. S.; FALBO, R. D. A.; GUIZZARDI, G. A uml profile for modeling framework-based web information systems. In: **12th International Workshop on Exploring Modelling Methods in Systems Analysis and Design EMMSAD2007**. [S.l.: s.n.], 2007. p. 153–162. Citado na página 19.

STARY, C. Contextual prototyping of user interfaces. In: **3rd conference on Designing interactive systems: processes, practices, methods, and techniques**. [S.l.: s.n.], 2000. p. 388–395. Citado na página 19.

STEINBERG, D. et al. **EMF: Eclipse Modeling Framework (2nd Edition)**. [S.l.]: Addison-Wesley Professional, 2008. Citado na página 58.

STOCQ, J.; VANDERDONCKT, J. A domain model-driven approach for producing user interfaces to multi-platform information systems. In: **working conference on Advanced visual interfaces**. [S.l.: s.n.], 2004. p. 395–398. Citado na página 19.

VARA, J. et al. Dealing with traceability in the mdd of model transformations. **Transactions on Software Engineering**, v. 40, n. 6, p. 555–583, 2014. Citado 2 vezes nas páginas 19 e 20.

VESCAN, A. An evolutionary multiobjective approach for the dynamic multilevel component selection problem. In: SPRINGER. **International Conference on Service-Oriented Computing**. [S.l.], 2015. p. 193–204. Citado 4 vezes nas páginas 46, 48, 49 e 51.

VODITHALA, S.; PABBOJU, S. A clustering technique based on the specifications of software components. In: IEEE. **2015 International Conference on Advanced Computing and Communication Systems**. [S.l.], 2015. p. 1–6. Citado 4 vezes nas páginas 46, 48, 49 e 51.

WANG, C.; REN, Y. A component clustering index tree based on semantic. In: SPRINGER. **International Conference on Web Information Systems and Mining**. [S.l.], 2011. p. 356–362. Citado 3 vezes nas páginas 46, 48 e 51.

WANG, R.; DAGLI, C. Executable system architecting using systems modeling language in conjunction with colored petri nets in a model-driven systems development process. **Systems Engineering**, v. 14, n. 4, p. 383–409, 2011. Citado na página 48.

WANG, Z.; LIU, D.; FENG, X. Improved som clustering for software component catalogue. In: YIN, F.-L.; WANG, J.; GUO, C. (Ed.). **Advances in Neural Networks – ISNN 2004**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004. p. 846–851. ISBN 978-3-540-28647-9. Citado 3 vezes nas páginas 46, 48 e 51.

WOHLIN, C. et al. **Experimentation in Software Engineering**. [S.l.]: Springer, 2012. Citado 2 vezes nas páginas 50 e 60.

WU, Y. et al. Construction of ontology-based software repositories by text mining. In: SHI, Y. et al. (Ed.). **Computational Science – ICCS 2007**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. p. 790–797. ISBN 978-3-540-72588-6. Citado 4 vezes nas páginas 46, 48, 49 e 51.

YE, H.; LO, B. W. A visualised software library: nested self-organising maps for retrieving and browsing reusable software assets. **Neural Computing & Applications**, Springer, v. 9, n. 4, p. 266–279, 2000. Citado 4 vezes nas páginas 44, 46, 48 e 51.

Apêndices

APÊNDICE A – ASSETS UTILIZADOS NO EXPERIMENTO

Coloured Petri Net Model of the bCMS system using CPN Tools
Required Tool: CPN Tools
Programming/Modeling Languages: Coloured petri nets and Standard ML.
Artifact Development Contexts: Workshop/Focus Group
Artifact Types: Model
System/Software Domains: Real-time embedded systems System Software
Lifecycle Phases: Requirements System
Modeling Languages/Notations: Petri nets (Coloured)
Keywords: Requirements CMA@RE2013 coloured petri nets executable verification

bCMS Case Study: FAMILIAR
Required Tool: Familiar
Programming/Modeling Languages: Feature Modeling with Familiar
Artifact Development Contexts: Academia Challenge Problem/Competition Workshop/Focus Group
Artifact Types: Model
System/Software Domains: Reactive System Software Product Line
Lifecycle Phases: Requirements Validation/Verification/Analysis Domain Analysis
Modeling Languages/Notations: Familiar
Keywords: CMA@RE2013

URML Model of bCMS (HTML Export)
Required Tool:
Programming/Modeling Languages: URML
Artifact Development Contexts: Academia Workshop/Focus Group
Artifact Types: Model
System/Software Domains: Business/Information Systems Software Product Line
Lifecycle Phases: Requirements
Modeling Languages/Notations: URML
Keywords: bCMS Early Requirements CMA@RE2013 URML

Applying BPMN on bCMS
Artifact Development Contexts: Academia Workshop/Focus Group
Artifact Types: Model
System/Software Domains:
Lifecycle Phases: Requirements Validation/Verification/Analysis Implementation/Deployment Domain Analysis
Modeling Languages/Notations: BPMN

Updated Activity Theory bCMS Model Description for CMA-2012
Artifact Development Contexts: Workshop/Focus Group
Artifact Types: Model
System/Software Domains: Cyberphysical Systems
Lifecycle Phases: Requirements
Keywords: User Requirements Notation bCMS Activity Theory Early Requirements

Umple submission for Comparing Modeling Artifacts workship at Models 2012
Artifact Development Contexts: Challenge Problem/Competition
Artifact Types: Model
Lifecycle Phases: Software Architecture
Modeling Languages/Notations: Umple
Keywords: bCMS Umple CMA workshop

Models for bCMS using AspectSM
Programming/Modeling Languages: UML, AspectSM UML Profile, MARTE UML Profile
Artifact Development Contexts: Academia
Artifact Types: Model
Lifecycle Phases: Validation/Verification/Analysis
Modeling Languages/Notations: statemachine diagram

bCMS in LEAP
Programming/Modeling Languages: LEAP
Artifact Development Contexts: Academia
Artifact Types: Model
System/Software Domains: Enterprise Systems
Lifecycle Phases: Requirements Runtime/Operational Software Architecture
Modeling Languages/Notations: LEAP
Keywords: CMA@MODELS2012

Comparison Criteria for bCMS Models of CMA Workshop
Artifact Development Contexts: Challenge Problem/Competition Workshop/Focus Group
Artifact Types: Lecture/Tutorial/Training or Course material
Keywords: bCMS Comparing Modeling Approaches CMA Workshop Criteria Document Models

bCMS - Requirements Definition
Artifact Development Contexts: Challenge Problem/Competition Workshop/Focus Group
Artifact Types: Requirements Document
System/Software Domains: Reactive System Software Product Line
Lifecycle Phases: Requirements
Keywords: Software product line Comparing Modeling Approaches CMA Workshop Requirements

Reusable Aspect Models for the bCMS Case Study
Programming/Modeling Languages: Reusable Aspect Models
Artifact Development Contexts: Academia Workshop/Focus Group
Artifact Types: Model
Lifecycle Phases: Design
Modeling Languages/Notations: RAM
Keywords: Reusable Aspect Models

Activity Theory Models for the bCMS Case Study - CMA@MODELS2011
Artifact Development Contexts: Academia Research Project
Artifact Types: Model
System/Software Domains: System Software
Lifecycle Phases: Requirements software development phases
Keywords: Activity Theory Early Requirements High-Level Design URN AOM

bCMS case study models for OO-SPL approach
Required Tool: MagicDraw
Programming/Modeling Languages: UML
Artifact Development Contexts: Workshop/Focus Group
Artifact Types: Model
System/Software Domains: Software Product Line
Lifecycle Phases: Domain Analysis
Modeling Languages/Notations: class diagram sequence diagram statemachine diagram
Keywords: Object oriented model Software product line

bCMS-SPL case study: A proposition based on the Cloud Component Approach.
Required Tool: EMF
Artifact Development Contexts: Workshop/Focus Group
Artifact Types: Methodology/Technique/Process
System/Software Domains: Model transformation adaptive systems System Software
Lifecycle Phases: Requirements Implementation/Deployment Software Architecture software development phases Implementation Software Development process
Modeling Languages/Notations: Ecore
Keywords: Cloud components

Model Driven Service Engineering applied to bCMS
Programming/Modeling Languages: UML2
Artifact Development Contexts: Workshop/Focus Group
Artifact Types: Model
Lifecycle Phases: Requirements Software Architecture Validation Implementation Software Development process
Modeling Languages/Notations: activity diagram collaborations
Keywords: Service choreography component design model-driven development

bCMS Case Study: AoURN
Programming/Modeling Languages: AoURN
Artifact Development Contexts: Academia Challenge Problem/Competition Workshop/Focus Group
Artifact Types: Model
System/Software Domains: Reactive System
Lifecycle Phases: Requirements
Modeling Languages/Notations: User Requirements Notation
Keywords: User Requirements Notation Aspects bCMS CMA 2011 Workshop

APÊNDICE B – ATIVIDADES PREVISTAS NO TCC 1

Assim, a metodologia de pesquisa associada ao trabalho de conclusão de curso é embasada em Piffers et al. Peffers et al. (2007). As etapas da metodologia são discutidas como segue:

B.1 Pesquisa para o Trabalho de Conclusão de Curso 1

Identificar o problema e definir os objetivos da solução: O TCC 1 foi dedicado inteiramente para esta etapa. Inicialmente, a identificação do problema deu-se em tese de doutorado (BASSO, 2017), como um trabalho futuro que foi atacado neste novo estudo. Trata-se de um estudo que segue o que foi investigado anteriormente sobre *assets* para MDE como candidatos para a composição de *tool chains* (BASSO; WERNER; OLIVEIRA, 2017a). O tema de pesquisa em *data mining* foi o foco escolhido nesta temática, motivado após a realização de um estágio em *big data* realizado no `DTIC`. A partir e 22 de dezembro de 2018, alinhou-se um interesse de pesquisa comum, para investigar as técnicas adotadas na recomendação de *assets* por meio de *data mining*. Portanto, o problema investigado trata de um interesse de pesquisa do grupo LESSE, caracterizado por MDE como um Serviço, e também de um interesse profissional, caracterizado por uma especialidade em banco de dados que o proponente busca em seu currículo.

Projeto e desenvolvimento da solução: O projeto e desenvolvimento da solução será realizado somente após a conclusão do mapeamento sistemático. Isto porque requer a compreensão de modo mais aprofundado das interfaces necessárias para a execução das tarefas de mineração de *assets* de MDE. Busca-se o desenvolvimento de suporte ferramental que permita a recomendação de *assets* de MDE entre duas ferramentas disponíveis em nosso grupo de pesquisa: A `FOMDA DSL`, centrada em reúso sistemático e operando em ambiente de desenvolvimento integrado (IDEs), e o `RAS++ DSL`, centrado em reúso oportunista e operando em ambiente distribuído de repositório de *assets*. Ou seja, o objetivo é recomendar *assets*, incluindo as informações de *tool chain* de um repositórios, para integrá-los automaticamente numa *tool chain*. Para tal, prevê-se a elaboração de um projeto arquitetural bem elaborado, para o estilo arquitetura distribuída, útil.

Implementação e Demonstração: Esta atividade prevê a demonstração de duas técnicas de *data mining* para *assets* de um repositório de artefatos de MDE. Também para três cenários na composição de *tool chain* em contextos inter-organizacionais discutidos em (BASSO et al., 2017) como: Sistemas Embarcados, Redes de Sensores sem Fio e Sistemas de informação Web.

Avaliação: Para avaliar os componentes desenvolvidos para recomendação, planejou-se: 1) Um estudo experimental quantitativo, para avaliar o número *assets* retornados como falsos positivos e falsos negativos, determinando assim que técnica traz melhores recomendações; e talvez 2) Um estudo experimental qualitativo, à ser realizado com alunos, onde o objetivo é avaliarmos à aplicabilidade do suporte ferramental proposto.

Comunicação: Planejava-se a publicação dos resultados preliminares do TCC 2 para o Simpósio Brasileiro de Componentes e Reúso de Software (SBCARS). Posteriormente, planeja-se o envio de artigos com os resultados finais experimentais do TCC 2 para uma revista da área, como o Journal of Systems and Software e Software Practice and Experience.

Anexos

ÍNDICE