



 **Priorização de requisitos para um novo software de decisão multicritério: uma aplicação baseada no método *Fuzzy* TOPSIS**

 Viviane Ruotolo¹  Dahan Poiel Lima Schuster²  Tainara Silva Novaes³  Lucyano Campos
Martins⁴ e  Francisco Rodrigues Lima Junior⁵

Versão do autor aceita publicada online: 23 janeiro 2024

Publicado online: 11 março 2024

^{1,2,3} Engenharia de Computação

⁴ Desenvolvedor e Mestre em Modelagem Computacional de Sistemas

⁵ Universidade Tecnológica Federal do Paraná

Nota dos autores

Autores declaram que não há conflitos de interesses.

Agradecimentos: Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq (código 409529/2021-4)

Como citar esse artigo - American Psychological Association (APA):

Ruotolo, V., Schuter, D. P. L., Novaes, T. S., Martins, L. C. & Lima Junior, F. R. (2024, artigo aceito online). Priorização de requisitos para um novo software de decisão multicritério: uma aplicação baseada no método *Fuzzy* TOPSIS. *Exacta*, artigo aceito online.

<https://doi.org/10.5585/2024.23483>

Resumo

A priorização de requisitos de software é uma atividade de suma importância para a Engenharia de Software. É abordada na literatura como um problema de decisão multicritério, baseado em julgamentos de especialistas. Para lidar com a subjetividade desses julgamentos, este estudo propõe o uso do método *Fuzzy* TOPSIS e realiza uma aplicação para priorização

de requisitos para um novo software de apoio a tomada de decisões. Os requisitos foram avaliados por três pesquisadores envolvidos no projeto de desenvolvimento. Os resultados apontaram como prioritários os requisitos relacionados ao uso de critérios qualitativos e qualitativos, e à possibilidade de atribuição de pesos aos critérios. Diferentemente das abordagens prévias, o método aplicado torna possível avaliar os pesos dos decisores utilizando termos linguísticos. Os procedimentos metodológicos adotados podem ser replicados em aplicações futuras voltadas à priorização de requisitos.

PALAVRAS-CHAVE: priorização de requisitos de software. tomada de decisão multicritério. *Fuzzy TOPSIS*.

Requirements prioritization for a new multicriteria decision software: an application based on the Fuzzy TOPSIS method

Prioritizing software requirements is an extremely important activity for Software Engineering. It is approached in the literature as a multi-criteria decision problem, based on expert judgments. To deal with the subjectivity of these judgments, this study proposes the use of the Fuzzy TOPSIS method and carries out an application to prioritize requirements for new decision-making support software. The requirements were evaluated by three researchers involved in the development project. The results highlighted the requirements related to the use of qualitative and qualitative criteria, and the possibility of assigning weights to the criteria, as priorities. Unlike previous approaches, the applied method makes it possible to evaluate the weights of decision makers using linguistic terms. The methodological procedures adopted can be replicated in future applications aimed at prioritizing requirements.

KEYWORDS: prioritization of software requirements. multicriteria decision making. Fuzzy TOPSIS.

1 Introdução

A definição de requisitos é uma etapa de suma importância para a Engenharia de Software. É nela que o cliente especifica quais são as características desejáveis ao produto, para que este seja projetado de modo a atender às necessidades de uso (Mczara, Sarkani, Holzer & Eveleigh, 2015). Ao desenvolver novos softwares, frequentemente é necessário identificar uma larga escala de requisitos para o sistema, embora muitas vezes a implementação de todos

eles não seja possível, por dependerem de fatores como o tempo e os recursos financeiros (Afzal & Sadim, 2018). Dessa forma, observa-se a importância da priorização de requisitos, a qual propicia a obtenção de um sistema com escala menor de requisitos, redução de custos e aumento da satisfação do cliente (Pacheco, García & Reyes, 2018).

Na literatura são sugeridas diferentes metodologias para apoiar a definição dos requisitos de software (Pacheco, García & Reyes, 2018; Bukhsh et al., 2020). Enquanto algumas delas focam na elicitación dos requisitos, outras são voltadas para a priorização (ou escolha) destes. A priorização de requisitos é especialmente útil quando se deseja estabelecer uma listagem das funcionalidades essenciais para o lançamento de uma versão inicial do software, que poderá ser atualizada e melhorada no decorrer do tempo (Barbosa, Pinheiro, Silveira & Filho, 2019).

A priorização de requisitos de software é frequentemente abordada na literatura como um problema de tomada de decisão multicritério (*Multicriteria Decision Making*, MCDM). Na última década, o uso de métodos MCDM baseados na Teoria dos Conjuntos *Fuzzy* emergiu e tem se popularizado devido à capacidade destes métodos de lidarem com julgamentos humanos subjetivos, os quais são frequentemente utilizados como entrada nos modelos decisórios para a priorização de requisitos de software.

Embora haja vários estudos que aplicaram métodos MCDM para priorização de requisitos de software (Sadiq, Parveen & Jain, 2021; Nazim, Mohammad & Sadiq, 2022), por meio do levantamento bibliográfico realizado pelo presente estudo e da consulta a revisões sistemáticas sobre esta temática (Achimugu et al., 2014; Bukhsh et al., 2020), verificou-se a escassez de estudos brasileiros com esse enfoque. Além disso, constatou-se a ausência de trabalhos voltados para priorização de requisitos de software de apoio à MCDM.

Softwares de MCDM são ferramentas importantes para apoiar gestores em processos decisórios inerentes a diversas áreas da gestão empresarial. São aplicáveis em qualquer situação que envolva a avaliação de duas ou mais alternativas com base em múltiplos critérios de decisão, sendo úteis para apoiar processos decisórios do nível operacional ao estratégico (Montez, Sánchez, Villar & Herrera, 2015).

Diante do exposto, o presente estudo objetiva aplicar o método *Fuzzy TOPSIS* (*Technique for Order Preference by Similarity to Ideal Solution*) no apoio à priorização de requisitos para um novo software de MCDM. Esse estudo parte da hipótese de que o *Fuzzy TOPSIS* seja uma abordagem adequada para apoiar a priorização de requisitos de software de MCDM. A aplicação realizada contou com a participação de três decisores e foi realizada em um projeto de pesquisa que visa desenvolver um software voltada para decisões em grupo sob incerteza.

A aplicação foi apoiada pela norma ISO/IEC 25000 (Requisitos e Avaliação de Qualidade de Sistemas e Software) e buscou auxiliar a equipe na definição das características que irão compor a primeira versão desse sistema.

O restante deste estudo está organizado como segue. A Seção 2 detalha os procedimentos adotados para a realização do presente estudo. A Seção 3 apresenta um referencial teórico sobre elicitación e priorização de requisitos de software, aplicações de métodos MCDM, norma ISO/IEC 25000 e *Fuzzy* TOPSIS. A Seção 4 apresenta os resultados da aplicação do *Fuzzy* TOPSIS em um caso real e discute as contribuições deste estudo. Por fim, a Seção 5 apresenta a conclusão e oportunidades para estudos futuros.

2 Método

A abordagem metodológica adotada neste estudo é baseada em modelagem e simulação computacional. Segundo a concepção de Banks (1998), modelagem é uma reprodução de um sistema real, que requer certa preocupação em se definir os limites do modelo que representa o sistema. A metodologia de modelagem e simulação se baseia na construção de modelos compostos por variáveis de entrada e variáveis de saída que possuem relacionamentos quantificáveis entre si (Lima & Carpinetti, 2015). No presente estudo, enquanto as variáveis de entrada se referem aos pesos dos requisitos definidos por cada decisor, bem como ao peso atribuído à opinião de cada decisor, a variável de saída consiste no grau de prioridade global de cada requisito. As etapas adotadas para o desenvolvimento deste trabalho foram levantamento bibliográfico, coleta de dados, e modelagem e aplicação do *Fuzzy* TOPSIS, conforme descrito a seguir.

2.1 Levantamento bibliográfico

O levantamento bibliográfico foi realizado em dois estágios. O primeiro deles permitiu mapear estudos prévios que apresentam aplicações de métodos de MCDM para priorização de requisitos de software, como discutido na Seção 3.2. O segundo estágio do levantamento focou em investigar elementos fundamentais para a construção do modelo de tomada de decisão e aplicação do *Fuzzy* TOPSIS, tendo como principais temas o processo de elicitación de requisitos (Seção 3.1), ISO/IEC 25000 (Seção 3.3) e *Fuzzy* TOPSIS (Seção 3.4).

No primeiro estágio, as buscas foram realizadas com base em Barud et al. (2021), utilizando a base *Scopus*, por ser uma das principais bases das áreas de Engenharia de Produção e Engenharia de Software. Os seguintes procedimentos foram aplicados:

- 1) definição das palavras-chave e inserção da *string* da seguinte busca na base de dados - ("*software requirement*" OR "*system requirement*") AND (*selection* OR *prioritization* OR *choice* OR *sorting* OR *evaluation*) AND (*multicriteria* OR *multi-criteria*). Essa *string* foi aplicada ao conteúdo do título, resumo e *keywords* de cada artigo. Identificou-se 52 estudos, quase todos escritos em idioma inglês, e apenas um em francês. Os resultados dessa etapa resultaram na Figura 1 da Seção 3.2, que mostra o ano de publicação e os países de origem;
- 2) Aplicação de critérios de elegibilidade – nesse passo foram incluídos apenas estudos publicados em periódicos ou anais de eventos, excluindo capítulos de livro. Por meio da leitura do *abstract* e do conteúdo dos artigos, foram selecionados apenas estudos que apresentam aplicações de métodos de MCDM para priorização de requisitos de software. Todos os artigos selecionados foram escritos em língua inglesa;
- 3) Classificação dos estudos selecionados – baseou-se na leitura completa do conteúdo dos artigos e levou em conta as técnicas de MCDM utilizadas e o objetivo do caso de aplicação. Os resultados desse passo resultaram na classificação de 19 estudos, como mostra o Quadro 1 da Seção 3.2;
- 4) Buscas complementares - também foram realizadas buscas adicionais na tentativa de identificar possíveis estudos com aplicações prévias de técnicas multicritério para priorização de requisitos de software de MCDM. Essas buscas foram realizadas no *Science Direct* e no *Google Scholar*, utilizando a *string* ("*software requirement*" OR "*system requirement*") AND (*multicriteria* OR *multi-criteria*) AND ("*decision software*" OR "*decision-making software*" OR "*decision support system*"). Ao analisar todos os resultados mostrados em cada base (308 no *Google* e 451 no *Science Direct*), não foram encontrados estudos voltados para priorização de requisitos de software de MCDM.

Na segunda etapa do levantamento bibliográfico, o enfoque foi identificar estudos sobre conceitos relativos à elicitación de requisitos de software, norma ISO/IEC 25000 e método *Fuzzy TOPSIS*. Foram consultadas as bases de dados *Scopus*, *Web of Science* e *Google Scholar*, utilizando palavras-chave como "*requirement elicitation*", "*ISO/IEC 25000*", "*ISO 25000*" e "*Fuzzy TOPSIS*". Foram selecionados trabalhos publicados em inglês e português, sendo que a maioria deles é proveniente de periódicos estrangeiros. Os resultados obtidos auxiliaram na definição e priorização dos requisitos no caso de aplicação;

2.2 Coleta de dados, modelagem e aplicação

A pesquisa de campo se iniciou pela etapa de coleta de dados. Nessa etapa, os requisitos foram definidos por três pesquisadores de um projeto de desenvolvimento de software, que

atuaram como decisores na aplicação em questão. A definição dos requisitos foi feita por meio de um *brainstorming* para levantar as funcionalidades requeridas do software. Também se embasou em patentes (Gomes, Santos & Vanni, 2020; Gomes, Teixeira & Santos, 2020) e estudos (Montes et al., 2015; Hamdan & Cheaitou, 2017; Dymova et al., 2021; Tahri et al., 2022) que propõem softwares com interface gráfica para MCDM. Utilizando a ferramenta *Google Forms*, um formulário foi criado para que os decisores fornecessem seus julgamentos a respeito do nível de importância (pesos) dos requisitos listados. Esses dados foram tabulados e inseridos no modelo *Fuzzy TOPSIS*.

Na etapa de modelagem e aplicação, por meio do uso do software *MS Excel*, construiu-se um modelo computacional para automatizar os cálculos do *Fuzzy TOPSIS*. Essa ferramenta foi escolhida devido à facilidade de implementação das equações e de manipulação do modelo por usuários não especialistas em programação. O modelo computacional foi implementado conforme as equações propostas por Chen (2000) e seguindo a sequência lógica utilizada em Tominaga et al. (2021) (detalhada na Seção 3.4).

Os dados coletados dos decisores foram inseridos no modelo para obter um ranqueamento segundo o grau de prioridade global de cada requisito. O primeiro passo foi converter os julgamentos linguísticos dos decisores, associando-os a números *fuzzy* triangulares. Na sequência, calculou-se a matriz de decisão normalizada e ponderada. A partir dos valores das distâncias entre as pontuações dos requisitos e as soluções ideais, calculou-se o grau de prioridade global de cada requisito. Os resultados da aplicação foram analisados pelos decisores, que então definiram quais requisitos serão considerados na implementação da primeira versão do software.

3 Referencial Teórico

3.1 Elicitação de requisitos de software

Uma etapa de suma importância da Engenharia de Software consiste na elicitação de requisitos do software, na qual o cliente e/ou futuros usuários explicitam as características requeridas do produto (Mczara, Sarkani, Holzer & Eveleigh, 2015). Para que seja possível melhorar a qualidade de um sistema, primeiramente é importante aprimorar a qualidade dos requisitos elicitados e determinar uma boa técnica de obtenção destes requisitos (Pacheco, García & Reyes, 2018).

Diversas técnicas podem ser usadas para apoiar a elicitação de requisitos. As abordagens tradicionais envolvem entrevistas, questionários e pesquisas com clientes e usuários (Tiwari, Rathore e Gupta, 2012). Há também a técnica denominada *brainstorming*, que busca juntar

ideias de diferentes decisores e analisá-las. Pacheco, García & Reyes (2018) desenvolveram um estudo de revisão sistemática sobre técnicas de elicitação de requisitos, o qual analisou 140 estudos publicados entre 1993 a 2015. Esses autores concluíram que as técnicas de elicitação mais utilizadas foram entrevistas, prototipagem, análise de cenários, grupo focal e *brainstorming*.

Especialmente quando se deseja desenvolver um produto mínimo viável e/ou prevenir mudanças bruscas nos requisitos do sistema, é importante possuir uma técnica bem definida para priorizar, dentre os requisitos listados, aqueles que mais refletem a real necessidade do cliente (Bukhsh et al., 2020). Priorizar um conjunto de requisitos de software é um trabalho complexo, que pode envolver a avaliação das prioridades de cada cliente e dos riscos associados a cada requisito. Tal priorização não deve ser feita de maneira informal, mas sim tratada como um problema de tomada de decisão (Pacheco, García & Reyes, 2018). A seção a seguir foca em estudos que apresentam aplicações de métodos de MCDM para priorização de requisitos de software.

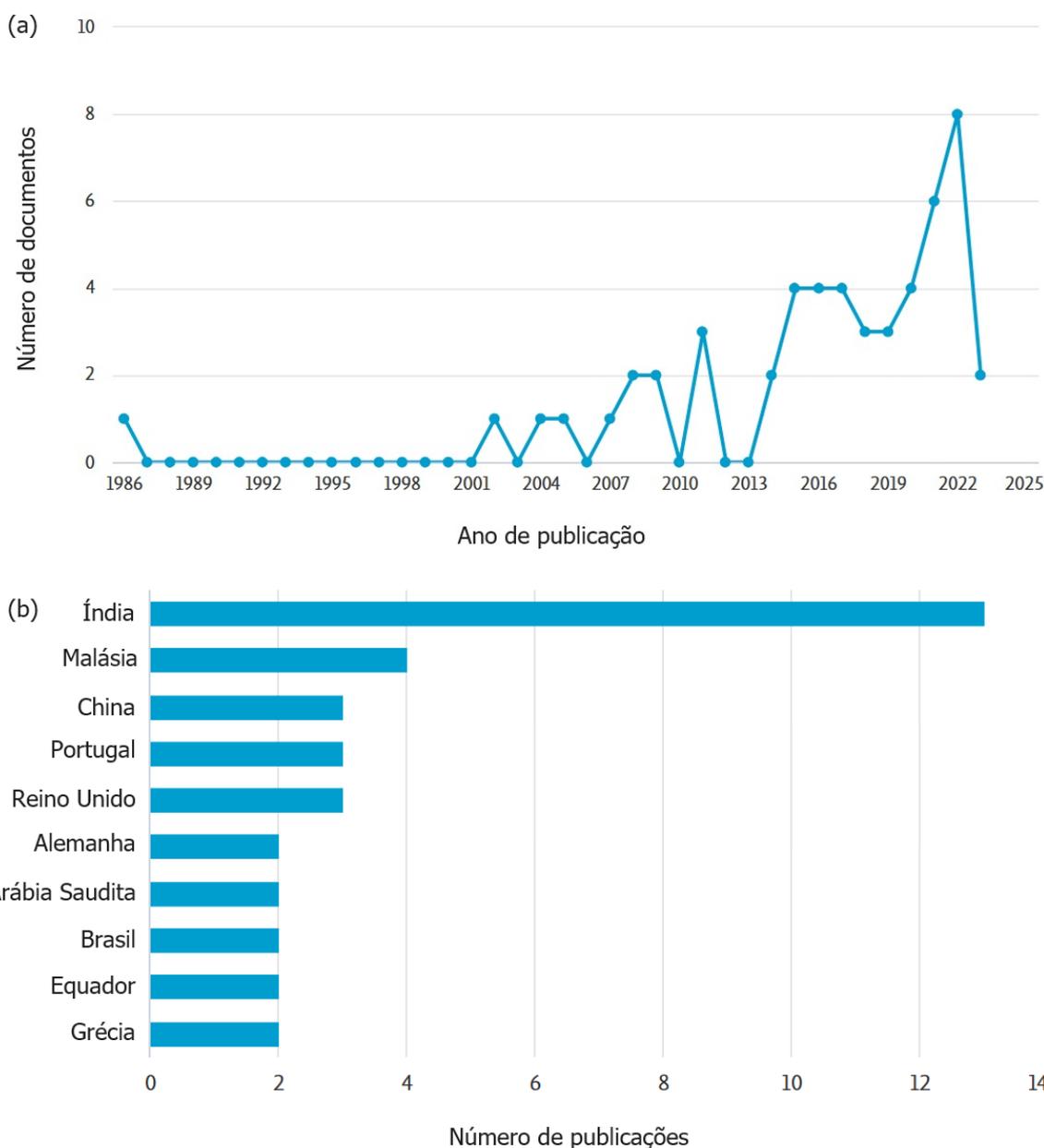
3.2 Aplicações de métodos multicritério para priorização de requisitos de software

A literatura apresenta vários estudos que aplicam métodos de MCDM com o intuito de apoiar a priorização de requisitos de software (Achimugu et al., 2014; Bukhsh et al., 2020). Esses métodos são ferramentas capazes de gerar uma classificação global de um conjunto de alternativas a partir das pontuações alcançadas por elas individualmente em múltiplos critérios de decisão (Lima & Carpinetti, 2015). Alguns desses métodos são capazes de levar em consideração a opinião de vários especialistas ou usuários envolvidos no problema em questão, também chamados de decisores. Achimugu et al. (2014) realizaram um estudo de revisão sistemática da literatura a fim de mapearem os métodos utilizados para priorização de requisitos de software. Dentre os 73 estudos selecionados por esses autores, os métodos mais aplicados foram AHP e QFD (*Quality Function Deployment*). Outro estudo de revisão sistemática sobre essa temática foi proposto por Bukhsh et al. (2020). Nesse estudo foram analisados 102 artigos, publicados entre 2007 e 2019, que propuseram ou avaliaram abordagens de priorização de requisitos. Esses autores concluíram que o método AHP e as técnicas *fuzzy* estão entre as abordagens mais utilizadas.

Para mapear os estudos prévios que aplicaram métodos MCDM em problemas de priorização de requisitos de *software*, no presente estudo foram realizadas buscas conforme os procedimentos descritos na Seção 2.1. A Figura 1(a) apresenta a quantidade de estudos encontrados no Scopus relacionados essa temática, totalizando 52 estudos. Esse total inclui

estudos práticos e teóricos, como estudos de revisão sistemática. Nota-se que a quantidade de estudos cresceu nas últimas duas décadas, com a maioria deles tendo sido publicada nos últimos oito anos. Já a Figura 1(b) classifica esses estudos de acordo com o país de origem do primeiro autor. Essa figura se restringe a exibir os países que possuem no mínimo duas publicações. Foram identificados estudos de 25 países distintos, sendo que os estudos da Índia e Malásia predominam. Apenas dois estudos brasileiros foram encontrados, o que demonstra a escassez de trabalhos nacionais sobre a presente temática.

Figura 1 – Quantidade de estudos por ano (a) e por país de origem (b)



Fonte: Elaborado pelos autores.

O Quadro 1 apresenta 19 estudos que realizaram aplicações desses métodos em problemas de priorização de requisitos de *software*, os quais foram coletados conforme descrito na seção 2. As técnicas anteriormente aplicadas incluem AHP, *Fuzzy AHP*, *Fuzzy TOPSIS*, *Fuzzy Graph*, *Fuzzy MoSCoW* (acrônimo para as expressões: *must-have*, *should-have*, *could-have*, and *won't-have*), entre outras. É notável o vasto uso dos métodos baseados na Teoria dos Conjuntos *Fuzzy*, empregados em nove estudos.

Quadro 1 – Aplicações de métodos de MCDM para priorização de requisitos de software

Autor	Técnica(s) utilizada(s)	Aplicação
Afrin e Sadiq (2017)	<i>Fuzzy AHP</i> e <i>Fuzzy TOPSIS</i>	Sistema de avaliação do desempenho acadêmico
Afzal e Sadim (2018)	AHP	Software de avaliação do desempenho acadêmico
Ahmad et al. (2017)	<i>Fuzzy MoSCoW</i>	Software de gerenciamento de biblioteca
Arévalo, Carrasco, Espinoza e Córdova (2020)	<i>Neutrosophic AHP</i>	Sistema de reconhecimento facial
Barbosa et al. (2019)	Análise de Decisão Verbal	Definição da próxima versão de um <i>software</i>
Cañizares, Becerra, Jalón e Acosta (2021)	<i>Neutrosophic TOPSIS</i>	Sistema de informação
Farhana, Shabana e Mohd (2023)	<i>Rough attributed goal-oriented software requirements analysis (RAGOSRA)</i>	Sistema de avaliação institucional
Fernandes, Rodrigues e Costa (2015)	AHP e ELECTRE I	Software de gestão de projetos
Gerogiannis e Tzikas (2017)	<i>Fuzzy Linguistic 2-tuple</i>	Projeto de software distribuído
McZara et al. (2015)	<i>Linguistic Tools and Constraint Solvers</i>	Priorização de grandes conjuntos de requisitos
Mohd, Chaudhary e Mohd (2022)	<i>Fuzzy AHP</i> e <i>Fuzzy TOPSIS</i>	Sistema de avaliação de um instituto educacional
Mougouei e Powers (2021)	<i>Fuzzy Graph e Integer Programming</i>	Análise das dependências entre os requisitos de <i>software</i> e seleção dos principais
Mohd e Susheela (2022)	<i>Rough sets</i>	Sistema de avaliação de um instituto educacional
Muhammad et al. (2018)	AHP	Definição de requisitos não funcionais

Autor	Técnica(s) utilizada(s)	Aplicação
Nazim, Mohammad e Sadiq (2022)	<i>Fuzzy AHP e Fuzzy TOPSIS</i>	Sistema de avaliação do desempenho acadêmico
Onar, Oztaysi e Cengiz (2018)	<i>Hesitant Fuzzy Linguistic Term Sets</i>	Desenvolvimento de plataforma de busca de empregos
Sadiq e Jain (2014)	<i>Fuzzy Preference Relations</i>	Sistema de avaliação do desempenho acadêmico
Sadiq, Khan e Mohammad (2020)	<i>Fuzzy TOPSIS</i>	Sistema de avaliação do desempenho acadêmico
Sadiq, Parveen e Jain (2021)	<i>Incomplete Linguistic Preference Relations</i>	Sistema de avaliação do desempenho acadêmico

Fonte: Elaborado pelos autores.

Dentre os métodos mostrados no Quadro 1, o *Fuzzy TOPSIS* se destaca pela sua simplicidade de implementação e por certos benefícios de uso. Apesar de ter sido aplicado de forma bem sucedida em quatro casos de priorização de requisitos de sistemas de avaliação do desempenho acadêmico (Afrin & Sadiq, 2017; Sadiq, Khan & Mohammad, 2020; Nazim, Mohammad & Sadiq, 2022), não foram encontradas aplicações deste método voltadas à priorização de requisitos de outros tipos de software.

Ainda que o TOPSIS possua diversas versões aprimoradas que se mostram bastante efetivas, como o TOPSIS-2NE (Silva, Gomes & Souza, 2020), no presente estudo foi escolhido o *Fuzzy TOPSIS* proposto por Chen (2000) para dar suporte à priorização de requisitos. Essa escolha se justifica pelo fato desse método ser voltado para decisões sob incerteza, baseadas nos julgamentos subjetivos dos decisores. Além disso, a escala para avaliação das alternativas e critérios é mais simples do que as escalas comparativas utilizadas no AHP e *Fuzzy AHP* (Lima & Carpinetti, 2015).

Vale destacar também que, por meio das buscas realizadas e da análise de estudos de revisão sistemática sobre esse tema (Bukhsh et al., 2020; Achimugu et al., 2014), não foi encontrado um estudo que aplique um método MCDM para priorização de requisitos de softwares de apoio à MCDM. A fim de preencher essa lacuna, as seções a seguir apresentam a fundamentação teórica utilizada como apoio para definição dos requisitos de software (Seção 3.3) e para aplicação do *Fuzzy TOPSIS* (Seção 3.4) em um caso real.

3.3 ISO/IEC 25000

A série de normas ISO/IEC 25000, também conhecida como SQuaRE (*System and Software Quality Requirements and Evaluation*), pode ser considerada a norma mais relevante para

avaliar a qualidade de softwares (ISO/IEC25000, 2014). As características da qualidade de software definidas pela ISO/IEC 25000 incluem adequação funcional, eficiência de desempenho, usabilidade, compatibilidade, confiança, segurança, manutenibilidade e portabilidade. O Quadro 2 descreve sucintamente o conjunto de subcaracterísticas associadas a essas características (ISO/IEC25000, 2014; Morais & Lima, 2017). Quadro 2 –

Características e subcaracterísticas da qualidade de software

Característica	Subcaracterísticas
Adequação funcional	Integridade funcional: capacidade de atender às tarefas e aos objetivos do usuário.
	Correção funcional: fornecer resultados corretos e com precisão.
	Adequação funcional: realizar tarefas e certos objetivos de maneira fácil.
Eficiência de desempenho	Comportamento em relação ao tempo: fornecer tempos de resposta e processamento apropriados quando o software executa suas funções.
	Utilização de recursos: grau com que os recursos usados atendem aos requisitos.
	Capacidade: grau com que a capacidade de parâmetros do software atende aos requisitos.
Usabilidade	Reconhecibilidade: grau com que o usuário é capaz de reconhecer se o produto é adequado às suas necessidades.
	Apreensibilidade: capacidade de ser usado para alcançar objetivos específicos de aprendizagem, de forma eficiente, eficaz e sem riscos
	Operacionalidade: permitir ao usuário operá-lo e controlá-lo de forma fácil.
	Proteção de erro: proteger os usuários de cometer falhas.
	Estética da Interface: possuir uma interface que seja satisfatória ao usuário.
	Acessibilidade: capacidade de ser usado por usuários com diferentes características e habilidades para alcançar objetivos especificados.
Compatibilidade	Coexistência: compartilhar recursos com outro software impactar outro produto.
	Interoperabilidade: interagir com um ou mais sistemas especificados.
Confiança	Maturidade: atender às necessidades de confiabilidade quando operado em condições normais.
	Disponibilidade: ser operacional e acessível quando requerido para uso.
	Tolerância a falhas: garantir um nível de desempenho especificado em caso de falhas.
	Capacidade de Recuperação: restabelecer seu nível de desempenho especificado e recuperar os dados diretamente afetados no caso de uma falha.
Segurança	Confidencialidade: permitir acesso de dados somente a usuários autorizados.
	Integridade: bloquear acesso e modificações de usuários não autorizados.
	Não repúdio: comprovar ações, eventos, alterações e envio de informações para que não possam ser repudiados futuramente.
	Responsabilidade: rastrear as ações de entidades específicas.
	Autenticidade: comprovar a identidade de um sujeito ou recurso caso seja requerido.
	Modularidade: alterar elementos do software com impacto mínimo.

Característica	Subcaracterísticas
Manutenibilidade	Reutilização: capacidade que os componentes do software possuem de serem utilizados por outros sistemas existentes ou em construção.
	Analisabilidade: capacidade de avaliar o impacto de uma mudança em uma ou mais partes de um sistema, diagnosticar partes do sistema em que podem haver falhas e identificar componentes a serem modificados.
	Modificabilidade: permitir a implementação de modificações sem causar defeitos no software
	Testabilidade: capacidade de estabelecer critérios de teste para um sistema que foi modificado e de determinar se esses critérios foram cumpridos
Portabilidade	Adaptabilidade: ser adaptado para ambientes de operação especificados, sem a necessidade de aplicação de outras ações ou meios além daqueles já fornecidos
	Instalabilidade: ser corretamente instalado e / ou desinstalado em um ambiente especificado.
	Substituibilidade: substituir outro software no mesmo ambiente e para o mesmo fim.

Fonte: Morais e Lima (2017).

Ainda que a ISO/IEC 25000 descreva mais de 30 subcaracterísticas da qualidade de software, a adoção de todas elas pode se tornar inviável devido ao tempo consumido e a possibilidade de não serem todas relevantes para o sistema. Nesse sentido, essas subcaracterísticas servem como referência para a criação de modelos que sejam adequados ao contexto de uso (ISO/IEC25000, 2014).

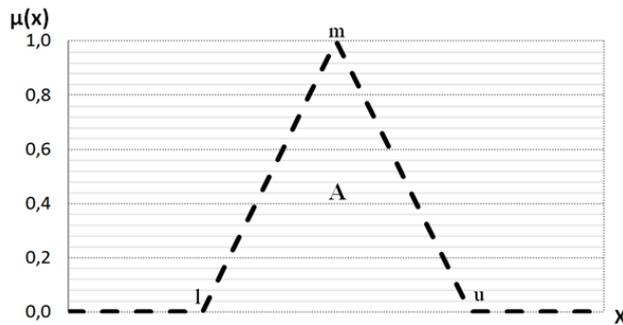
Morais e Lima (2017) destacam que a maneira como algumas das subcaracterísticas são definidas pela ISO/IEC 25000 não permite sua medição direta. Por isso, é necessário desdobrar as subcaracterísticas em requisitos objetivos, que possam ser entendidos e avaliados por diferentes usuários (ISO/IEC25000, 2014). Seguindo essa recomendação, no presente estudo foi estabelecido um conjunto de requisitos mensuráveis e objetivos, alinhados às características da ISO/IEC 25000. Para estimar quantitativamente o grau de prioridade desses requisitos, o método *Fuzzy* TOPSIS foi aplicado.

3.4 O método *Fuzzy* TOPSIS

A Teoria dos Conjuntos *Fuzzy* (*Fuzzy Set Theory* - FST) possibilita a modelagem de problemas com variáveis cujos valores possíveis são incertos, imprecisos e/ou subjetivos. Há diversos métodos híbridos que combinam a FST com métodos de MCDM, tais como *Fuzzy* AHP (Afrin; Sadiq, 2017), *Fuzzy* THOR (Elacoste et al., 2022) (*Multicriteria Decision Support Hybrid Algorithm for Decision Making Processes with Discrete Alternatives*, acronym THOR) e *Fuzzy* TOPSIS.

Dentre as técnicas multicritério baseadas em FST, o *Fuzzy TOPSIS* é um dos mais aplicados devido a sua versatilidade e facilidade de implementação e uso (Tominaga et al., 2021). Esse método permite aos decisores a expressão de seus julgamentos em formato linguístico (como “alto” e “baixo”) para avaliar as pontuações das alternativas do problema. Os julgamentos dos decisores são quantificados por números *fuzzy* triangulares (Chen, 2000), conforme exemplifica a Figura 2. Cada número *fuzzy* triangular é descrito por meio de seus vértices (l, m, u), em que m indica um valor *crisp* central, l é o limite inferior e u é o limite superior. As operações de soma (Equação 1), subtração (Equação 2), multiplicação (Equação 3) e divisão (Equação 4) envolvendo números triangulares são realizadas com base nos valores de l, m e u (Lima & Carpinetti, 2015).

Figura 2 - Número *fuzzy* triangular



$$\text{Sendo } \mu_A(x) = \begin{cases} 0, & \text{se } x \leq l \\ \frac{x-l}{m-l}, & \text{se } x \in [l, m] \\ \frac{u-x}{u-m}, & \text{se } x \in [m, u] \\ 0, & \text{se } x \geq u \end{cases}$$

Fonte: Lima e Carpinetti (2015).

$$\tilde{A} + \tilde{B} = [l_A, m_A, u_A] + [l_B, m_B, u_B] = [l_A + l_B, m_A + m_B, u_A + u_B] \quad (1)$$

$$\tilde{A} - \tilde{B} = [l_A, m_A, u_A] - [l_B, m_B, u_B] = [l_A - l_B, m_A - m_B, u_A - u_B] \quad (2)$$

$$\tilde{A} * \tilde{B} = [l_A, m_A, u_A] * [l_B, m_B, u_B] = [l_A * l_B, m_A * m_B, u_A * u_B] \quad (3)$$

$$\tilde{A} / \tilde{B} = [l_A, m_A, u_A] / [l_B, m_B, u_B] = [l_A/u_B, m_A/m_B, u_A/l_B] \quad (4)$$

Os passos para realizar a aplicação do método *Fuzzy TOPSIS* são descritos a seguir de acordo com Tominaga et al. (2021). O algoritmo aplicado por Tominaga et al. (2021) é baseado na versão original proposta por Chen (2000). Todavia, enquanto na versão de Chen (2000) as alternativas são dispostas nas linhas da matriz de decisão, e os critérios nas colunas, no presente estudo as alternativas são dispostas nas linhas e os decisores nas colunas. Essa adaptação do método é feita quando se pretende priorizar as alternativas com base em uma única variável (que neste caso é o peso dos requisitos de um novo software), de acordo com a opinião de múltiplos decisores. Dessa forma, passa a ser possível considerar os pesos das opiniões dos decisores, conforme seus níveis de experiência e/ou responsabilidade a respeito

do processo decisório em questão. Considerando isso, os seguintes passos são descritos a seguir (Chen, 2000; Tominaga et al., 2021):

- i) Sendo \tilde{x}_{ij} o peso de cada requisito A_i ($i = 1, \dots, n$), fornecido pelo decisor D_j ($j = 1, \dots, m$), deve-se montar a matriz de decisão \tilde{D} (Equação 5). Os pesos atribuídos aos decisores são indicados pelo vetor \tilde{W} (Equação 6);

$$\tilde{D} = \begin{matrix} & D_1 & D_2 & \dots & D_j & \dots & D_m \\ \begin{matrix} A_1 \\ \vdots \\ A_i \\ \vdots \\ A_n \end{matrix} & \begin{bmatrix} \tilde{x}_{11} & \tilde{x}_{12} & \dots & \tilde{x}_{1j} & \dots & \tilde{x}_{1m} \\ \vdots & \vdots & & \vdots & & \vdots \\ \tilde{x}_{i1} & \tilde{x}_{i2} & \dots & \tilde{x}_{ij} & \dots & \tilde{x}_{im} \\ \vdots & \vdots & & \vdots & & \vdots \\ \tilde{x}_{n1} & \tilde{x}_{n2} & \dots & \tilde{x}_{nj} & \dots & \tilde{x}_{nm} \end{bmatrix} \end{matrix} \quad (5)$$

$$\tilde{W} = [\tilde{w}_1, \tilde{w}_2, \dots, \tilde{w}_m] \quad (6)$$

- ii) Normalizar a matriz de decisão \tilde{D} . A matriz normalizada \tilde{R} é gerada pela Equação 7, sendo que os valores de \tilde{r}_{ij} devem ser obtidos por meio da Equação 8;

$$\tilde{R} = [\tilde{r}_{ij}]_{m \times n} \quad (7)$$

$$\tilde{r}_{ij} = \left(\frac{l_{ij}}{u_j^+}, \frac{m_{ij}}{u_j^+}, \frac{u_{ij}}{u_j^+} \right), \text{ sendo } u_j^+ = \max_i u_{ij} \quad (8)$$

- iii) Calcular a matriz ponderada \tilde{V} , indicada pela Equação 9. Os valores de cada célula dessa matriz são produzidos pela aplicação da Equação 10, que calcula o produto dos pesos dos decisores \tilde{w}_j pelos valores de \tilde{r}_{ij} ;

$$\tilde{V} = [\tilde{v}_{ij}]_{m \times n} \quad (9)$$

$$\tilde{v}_{ij} = \tilde{r}_{ij} * \tilde{w}_j \quad (10)$$

- iv) De acordo com as Equações 11 e 12, deve-se obter a solução ideal positiva fuzzy (*Fuzzy Positive Ideal Solution*, A^+) e a solução ideal negativa fuzzy (*Fuzzy Negative Ideal Solution*, A^-). Com base em Chen (2000), os valores das soluções ideais para cada coluna são definidos como $\tilde{v}_j^+ = (1, 1, 1)$ e $\tilde{v}_j^- = (0, 0, 0)$;

$$A^+ = \{\tilde{v}_1^+, \tilde{v}_j^+, \dots, \tilde{v}_m^+\} \quad (11)$$

$$A^- = \{\tilde{v}_1^-, \tilde{v}_j^-, \dots, \tilde{v}_m^-\} \quad (12)$$

- v) Calcular as distâncias entre as pontuações dos requisitos e os valores de FPIS (D_i^+) e de FNIS (D_i^-). Nesse passo, usam-se as respectivas Equações 13 e 14, nas quais $d(. , .)$ representa a distância entre dois números *fuzzy*. Quando são adotados números *fuzzy* triangulares, aplica-se a Equação 15 para calcular os valores de $d(. , .)$;

$$D_i^+ = \sum_{j=1}^n d_v(\tilde{v}_{ij}, \tilde{v}_j^+) \quad (13)$$

$$D_i^- = \sum_{j=1}^n d_v(\tilde{v}_{ij}, \tilde{v}_j^-) \quad (14)$$

$$d(\tilde{x}, \tilde{z}) = \sqrt{\frac{1}{3} [(l_x - l_z)^2 + (m_x - m_z)^2 + (u_x - u_z)^2]} \quad (15)$$

- vi) Considerando os valores de D_i^+ e D_i^- , aplica-se a Equação 16 para calcular o coeficiente de aproximação CC_i , que indica a prioridade global de cada requisito. Por fim, gera-se um ranqueamento dos requisitos por meio da ordenação decrescente dos valores de CC_i . Quanto maior o valor de CC_i , melhor é a prioridade global do requisito.

$$CC_i = \frac{D_i^-}{(D_i^+ + D_i^-)} \quad (16)$$

4 Resultados e Discussão

4.1 Contexto de aplicação

Diante do crescente aumento de uso das técnicas *fuzzy* em artigos acadêmicos, um grupo de pesquisadores brasileiros iniciou o desenvolvimento de um software de MCDM, o qual permitirá automatizar a aplicação do método *Hesitant Fuzzy Linguistic TOPSIS* (Besleki & Lima, 2015). Esse software se diferencia das soluções existentes por ser adequado para situações de hesitação, ou seja, cenários em que a incerteza e a ausência de informações sobre o problema contribuem para que os decisores hesitem ao exprimir seus julgamentos sobre os pesos dos critérios e/ou pontuações das alternativas. Nesses casos, os decisores podem preferir expressar seus julgamentos de modo mais flexível, utilizando expressões linguísticas comparativas (como “entre médio e alto”, “no máximo alto” ou “pelo menos alto”), o que é possível por meio do uso de *Hesitant Fuzzy Linguistic Term Sets* (HFLTS) (Rodríguez, Martínez & Herrera, 2013).

No contexto apresentado, o objetivo da aplicação do *Fuzzy TOPSIS* é determinar os requisitos prioritários para a implementação da primeira versão do referido software. A priorização dos

requisitos auxiliará a equipe a alcançar um produto mínimo viável, que contenha as funcionalidades essenciais do software e garanta a satisfação dos usuários.

4.2 Avaliação da importância dos requisitos

A aplicação se iniciou pela composição de uma equipe de três decisores que são membros do projeto. O Decisor 1 (D_1) é o coordenador do projeto. O Decisor 2 (D_2) e o Decisor 3 (D_3) são pesquisadores que também possuem experiência em técnicas HFLTS.

Inicialmente esses decisores realizaram a definição dos requisitos, baseando-se nas necessidades do projeto, na análise de estudos sobre técnicas HFLTS (Rodríguez, Martínez & Herrera, 2013) e no levantamento sobre softwares para tomada de decisão baseados na Teoria dos Conjuntos Fuzzy (Montes et al., 2015; Hamdan & Cheaitou, 2017; Dymová et al., 2021; Tahri et al., 2022). Por meio de um *brainstorming*, esses requisitos foram definidos pelos decisores de forma alinhada às subcaracterísticas da qualidade sugeridas pela ISO 25000. Foram contempladas subcaracterísticas associadas às características de funcionalidade, usabilidade, segurança, portabilidade, confiança e manutenibilidade. Posteriormente, cada decisor estimou os pesos dos requisitos de acordo com sua experiência e com as necessidades do projeto em questão.

O Quadro 3 apresenta os requisitos e os julgamentos fornecidos a respeito dos pesos de cada um dos requisitos listados. Também apresenta os pesos atribuídos aos decisores, o que foi feito considerando a experiência de cada um na área de HFLTS e o nível de responsabilidade sobre o projeto.

Quadro 3 – Listagem de requisitos e avaliações quanto aos pesos dos requisitos

Subcaracterística	Requisitos	D_1	D_2	D_3
Completeness funcional	R ₁ . O software deve permitir aos decisores a atribuição de pesos (níveis de importância) aos critérios (Tahri et al., 2022)	Muito alto	Alto	Muito Alto
	R ₂ . O software deve permitir o uso de critérios qualitativos para avaliação de alternativas	Muito alto	Muito alto	Muito Alto
	R ₃ . O software deve permitir o uso de critérios quantitativos para avaliação de alternativas	Muito alto	Muito alto	Muito Alto
	R ₄ . O desempenho das alternativas em critérios quantitativos deve ser avaliado pelos decisores utilizando valores numéricos	Alto	Baixo	Médio
	R ₅ . Os pesos dos critérios devem ser avaliados pelos decisores utilizando julgamentos linguísticos (como baixo, médio ou alto)	Alto	Alto	Alto

Subcaracterística	Requisitos	D ₁	D ₂	D ₃
Operacionalidade	R ₆ . Os decisores poderão escolher uma escala com 5 ou 7 termos linguísticos (Montes et al., 2015)	Alto	Alto	Alto
Acessibilidade	R ₇ . O software precisa ser autoexplicativo em relação ao seu uso, orientando os decisores em relação a cada etapa de aplicação (Hamdan & Cheaitou, 2017)	Alto	Muito alto	Muito Alto
	R ₈ . O software deve fornecer suporte ao cadastro de usuários	Alto	Médio	Médio
Confidencialidade	R ₉ . O administrador deve ter acesso ao cadastro, exclusão e alteração de contas dos usuários	Muito alto	Alto	Alto
Instabilidade	R ₁₀ . O software deve funcionar em plataforma WEB (Montes et al., 2015)	Muito alto	Muito alto	Baixo
Capacidade de recuperação	R ₁₁ . Quando houver interrupção do servidor da página WEB, o software deve salvar temporariamente os dados inseridos pelos decisores na página até o retorno	Alto	Alto	Baixo
Modificabilidade	R ₁₂ . O código de programação do software deve conter comentários que expliquem as etapas implementadas, de modo a facilitar possíveis atualizações futuras	Alto	Muito alto	Alto
Apreensibilidade	R ₁₃ . Na seção de “ajuda ao usuário”, deve haver uma figura que forneça uma visão geral das etapas de uso, de modo ilustrativo e fácil de entender (Montes et al., 2015)	Alto	Muito alto	Alto
	R ₁₄ . Deve haver uma figura que represente a relação das teorias usadas como embasamento matemático para a realização dos cálculos do software (Dymova et al., 2021)	Alto	Médio	Médio
	R ₁₅ . Deve apresentar os principais elementos do problema em uma mesma página, de modo que o usuário consiga visualizá-los e manipulá-los facilmente (Tahri et al., 2022)	Médio	Muito alto	Alto
Autenticidade	R ₁₆ . O decisor poderá salvar os dados de entrada (pesos dos critérios e pontuações das alternativas) e os resultados calculados pelo software, o que permitirá a consulta a aplicações realizadas anteriormente pelo mesmo decisor	Médio	Alto	Alto
Pesos atribuídos aos decisores:		Muito Alto	Alto	Alto

Fonte: Elaborado pelos autores.

4.3 Resultados da aplicação Fuzzy TOPSIS

Os valores mostrados na Tabela 1 foram normalizados por meio do uso da Equação 8 e ponderados pela Equação 10, o que resultou na matriz mostrada na Tabela 2. A solução ideal positiva foi definida com base na Equação 11 como $A^+ = [(1; 1; 1), (1; 1; 1), (1; 1; 1)]$,

enquanto a solução ideal negativa foi definida de acordo com a Equação 12 e resultou em $A^- = [(0; 0; 0), (0; 0; 0), (0; 0; 0)]$.

Tabela 2 - Matriz de decisão normalizada e ponderada

Requisitos	D ₁			D ₂			D ₃		
	l	m	u	l	m	u	l	m	u
R ₁	0,56	1,00	1,00	0,25	0,56	1,00	0,38	0,75	1,00
R ₂	0,56	1,00	1,00	0,38	0,75	1,00	0,38	0,75	1,00
R ₃	0,56	1,00	1,00	0,38	0,75	1,00	0,38	0,75	1,00
R ₄	0,38	0,75	1,00	0,00	0,19	0,50	0,13	0,38	0,75
R ₅	0,38	0,75	1,00	0,25	0,56	1,00	0,25	0,56	1,00
R ₆	0,38	0,75	1,00	0,25	0,56	1,00	0,25	0,56	1,00
R ₇	0,38	0,75	1,00	0,38	0,75	1,00	0,38	0,75	1,00
R ₈	0,38	0,75	1,00	0,13	0,38	0,75	0,13	0,38	0,75
R ₉	0,56	1,00	1,00	0,25	0,56	1,00	0,25	0,56	1,00
R ₁₀	0,56	1,00	1,00	0,38	0,75	1,00	0,00	0,19	0,50
R ₁₁	0,38	0,75	1,00	0,25	0,56	1,00	0,00	0,19	0,50
R ₁₂	0,38	0,75	1,00	0,38	0,75	1,00	0,25	0,56	1,00
R ₁₃	0,38	0,75	1,00	0,38	0,75	1,00	0,25	0,56	1,00
R ₁₄	0,38	0,75	1,00	0,13	0,38	0,75	0,13	0,38	0,75
R ₁₅	0,19	0,50	0,75	0,38	0,75	1,00	0,25	0,56	1,00
R ₁₆	0,19	0,50	0,75	0,25	0,56	1,00	0,25	0,56	1,00

Fonte: Elaborado pelos autores.

A partir dos valores da Tabela 2, as distâncias entre as pontuações dos requisitos e as soluções ideais foram calculadas. A Tabela 3 apresenta os resultados do cálculo das distâncias (D^+) entre as pontuações dos requisitos e os valores de A^+ , os quais foram gerados pela aplicação das Equações 13 e 15. Essa tabela também apresenta os resultados da aplicação das Equações 14 e 15 no cálculo das distâncias (D^-) entre as pontuações dos requisitos e os valores de A^- .

Tabela 3 – Resultados dos cálculos das distâncias das alternativas em relação às soluções ideais

Requisitos	$d(\tilde{v}_{ij}, \tilde{v}_j^+)$			D^+	$d_v(\tilde{v}_{ij}, \tilde{v}_j^-)$			D^-
	D ₁	D ₂	D ₃		D ₁	D ₂	D ₃	
R ₁	0,253	0,501	0,389	1,143	0,879	0,723	0,753	2,355
R ₂	0,253	0,389	0,389	1,030	0,879	0,804	0,753	2,436
R ₃	0,253	0,389	0,389	1,030	0,879	0,804	0,753	2,436
R ₄	0,389	0,798	0,637	1,824	0,753	0,308	0,489	1,551
R ₅	0,389	0,501	0,501	1,391	0,753	0,723	0,678	2,154

Requisitos	$d(\tilde{v}_{ij}, \tilde{v}_j^+)$			D^+	$d_v(\tilde{v}_{ij}, \tilde{v}_j^-)$			D^-
	D_1	D_2	D_3		D_1	D_2	D_3	
R ₆	0,389	0,501	0,501	1,391	0,753	0,723	0,678	2,154
R ₇	0,389	0,389	0,389	1,166	0,753	0,804	0,753	2,311
R ₈	0,389	0,637	0,637	1,663	0,753	0,525	0,489	1,768
R ₉	0,253	0,501	0,501	1,255	0,879	0,723	0,678	2,279
R ₁₀	0,253	0,389	0,798	1,439	0,879	0,804	0,308	1,991
R ₁₁	0,389	0,501	0,798	1,688	0,753	0,723	0,308	1,784
R ₁₂	0,389	0,389	0,501	1,279	0,753	0,804	0,678	2,235
R ₁₃	0,389	0,389	0,501	1,279	0,753	0,804	0,678	2,235
R ₁₄	0,389	0,637	0,637	1,663	0,753	0,525	0,489	1,768
R ₁₅	0,569	0,389	0,501	1,459	0,532	0,804	0,678	2,013
R ₁₆	0,569	0,501	0,501	1,572	0,532	0,723	0,678	1,932

Fonte: Elaborado pelos autores.

Por fim, realizou-se o cálculo do grau de prioridade global dos requisitos (CC_i) usando a Equação 16. A Tabela 4 apresenta os resultados finais da aplicação. Nessa tabela, os valores de CC_i foram ordenados decrescentemente, de acordo com a prioridade global dos requisitos.

Tabela 4 – Ranqueamento dos requisitos de software de acordo com a prioridade global (CC_i)

Classificação	CC_i	Requisitos
1º	0,703	R ₂
1º	0,703	R ₃
2º	0,673	R ₁
3º	0,665	R ₇
4º	0,645	R ₉
5º	0,636	R ₁₂
5º	0,636	R ₁₃
6º	0,608	R ₅
6º	0,608	R ₆
7º	0,580	R ₁₀
7º	0,580	R ₁₅
8º	0,551	R ₁₆
9º	0,515	R ₈
9º	0,515	R ₁₄
10º	0,514	R ₁₁
11º	0,460	R ₄

Fonte: Elaborado pelos autores.

Houve empate em alguns requisitos no ranqueamento, já que alguns deles alcançaram valores idênticos de CC_i (por exemplo, R_2 e R_3 ; e R_{12} e R_{13}). Nota-se que os requisitos considerados como prioritários estão relacionados a funcionalidades essenciais do software, como permitir o uso de critérios qualitativos (R_2) e quantitativos (R_3) para avaliação das alternativas, e possibilitar a atribuição de pesos aos critérios (R_1). O requisito R_7 também se destacou, demonstrando a importância de o software ser autoexplicativo em relação ao seu uso e orientar os decisores em relação a cada etapa de aplicação. Por outro lado, os requisitos com menor prioridade foram R_4 e R_{11} . Esses requisitos se referem à tolerância a falhas caso haja interrupção do servidor WEB, e permitir a avaliação de critérios quantitativos pelos decisores utilizando valores numéricos. Embora sejam relevantes, os decisores consideraram que essas funcionalidades não são cruciais para a primeira versão do software.

Os resultados mostrados na Tabela 4 foram apresentados aos decisores, que concordaram com a ordem de prioridade estabelecida. Em acordo com a equipe desenvolvedora do software, os decisores optaram por incluir na primeira versão deste produto todos os requisitos listados entre a 1ª e a 8ª posição do ranqueamento. Por não serem primordiais nessa avaliação, os demais requisitos foram descartados, o que contribuiu para que a equipe conseguisse concluir a primeira versão do software dentro do prazo estabelecido.

4.4 Contribuições teóricas e práticas

Em comparação com os estudos prévios apresentados no Quadro 1, a metodologia proposta apresenta algumas vantagens. Por exemplo, ao contrário das abordagens baseadas em comparações pareadas, como AHP (Afzal & Sadim, 2018; Arévalo et al., 2020; Fernandes, Rodrigues e Costa, 2015; Muhammad et al., 2018) e *Fuzzy AHP* (Afrin & Sadiq, 2017; Mohd, Chaudhary & Mohd, 2022; Nazim, Mohammad & Sadiq, 2022), o *Fuzzy TOPSIS* não limita a quantidade de critérios e requisitos de software que podem ser considerados na avaliação.

Além disso, requer menor quantidade de julgamentos dos decisores do que esses métodos e não exige testes de consistência dos julgamentos, consumindo menos tempo no processo decisório.

Destaca-se também a capacidade da abordagem aplicada neste estudo para apoiar processos de MCDM em grupo. Em contraste aos estudos prévios que aplicaram *Fuzzy TOPSIS* para priorização de requisitos, no presente estudo é possível atribuir pesos aos decisores por meio de julgamentos linguísticos. Como esses pesos são estimados considerando fatores como o conhecimento, a experiência e/ou cargo de cada decisor, é difícil quantificá-los utilizando valores numéricos reais. Dessa forma, com a abordagem sugerida neste estudo para

priorização de requisitos, é possível lidar com a incerteza inerente à definição dos pesos de cada decisor.

Em relação às contribuições práticas, a principal delas está no apoio à priorização dos requisitos de um novo software de apoio a MCDM em grupo sob incerteza e hesitação. De acordo com o levantamento realizado pela equipe do projeto, esse é o primeiro software baseado no método *Hesitant Fuzzy Linguistic TOPSIS* (Beleski & Lima, 2023) com interface gráfica amigável para usuários não especialistas.

A definição dos requisitos prioritários foi de suma importância para o referido projeto, uma vez que subsidiou as etapas posteriores da construção desse software. Assim, os requisitos definidos e priorizados neste estudo foram utilizados na elaboração de *storyboards* que projetaram o processo de uso deste produto. Também auxiliaram nas etapas de modelagem e de implementação computacional do software. A priorização de requisitos contribuiu para que o processo de implementação computacional fosse mais efetivo, o que ajudou a garantir a finalização da primeira versão do software dentro do prazo estabelecido. Portanto, o *feedback* dos decisores em conjunto com as informações apresentadas nessa seção ajudaram a confirmar a hipótese de que o uso do *Fuzzy TOPSIS* é uma abordagem adequada para a priorização de requisitos de software de MCDM.

Outra contribuição deste estudo está no modelo computacional *Fuzzy TOPSIS*, implementado em *MS Excel* e disponibilizado na base *Mendeley Data*. Esse modelo pode ser adaptado e reutilizado por outros pesquisadores e gestores que necessitem priorizar ou selecionar alternativas, considerando múltiplos critérios, com base em julgamentos linguísticos dos decisores.

5 Conclusão

O presente estudo realizou uma aplicação real do método *Fuzzy TOPSIS* para priorização de requisitos de um novo software de apoio à MCDM. Os requisitos foram definidos pelos decisores de forma alinhada à ISO/IEC 25000, o que contribuiu para que eles abrangessem aspectos essenciais à qualidade de produtos de software. Os resultados da aplicação do método *Fuzzy TOPSIS* se mostraram adequados por refletirem a opinião dos decisores e por permitirem a eles a utilização de valores linguísticos, mais próximos da linguagem natural, para exprimir os pesos dos requisitos e os pesos dos decisores.

Com base no levantamento bibliográfico realizado, pode-se afirmar que o presente estudo é o primeiro a aplicar uma técnica multicritério para realizar a priorização de requisitos de

software de MCDM. Quando comparado às abordagens utilizadas nos estudos prévios, o *Fuzzy TOPSIS* apresentou diversas vantagens. Contudo, uma limitação do presente estudo é que os requisitos definidos pelos decisores não contemplaram todas as características da qualidade estabelecidas na norma ISO/IEC25000, pois eles optaram por focar nos elementos cruciais a fim de não onerar o processo de desenvolvimento.

Os procedimentos metodológicos adotados nesse estudo podem ser replicados em aplicações futuras que envolvam a priorização de requisitos visando ao desenvolvimento de novos produtos de software, ou ainda, a avaliação da qualidade de softwares existentes, a fim de selecionar os produtos mais adequados para aquisição. Para isso, recomenda-se que os decisores estabeleçam requisitos adequados ao contexto em questão, utilizando como base as métricas sugeridas pela ISO/IEC 25000. Essas aplicações poderão envolver a priorização de requisitos ou a avaliação da qualidade de diferentes tipos de sistemas, tais como softwares ERP (*Enterprise Resource Planning*), sistemas de gestão de projetos, sistemas de modelagem e gestão de processos, dentre outros.

Referências

Achimugu, P., Selamat, A., Ibrahim R., & Mahrin, M. N. (2014). A systematic literature review of software requirements prioritization research. *Information and Software Technology*, 56, 568-585. <https://doi.org/10.1016/j.infsof.2014.02.001>

Afrin, A., & Sadiq, Mohd. (2017). An integrated approach for the selection of software requirements using fuzzy AHP and fuzzy TOPSIS method. 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), 1094–1100. <https://doi.org/10.1109/ICICICT1.2017.8342722>

Afzal, N., & Sadim, M. (2018). Software Requirements Selection using AHP. *International Journal of Computer Science and Communication*, 9(2), 47–52.

Ahmad, K. S., Ahmad, N., Tahir, H., & Khan, S. (2017). Fuzzy_MoSCoW: A fuzzy based MoSCoW method for the prioritization of software requirements. 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), 433–437. <https://doi.org/10.1109/ICICICT1.2017.8342602>

Arévalo, N. P. B., Carrasco, M. F. C., Espinoza, J. L. T., & Córdova, M. V. (2020). Neutrosophic AHP for the prioritization of requirements for a computerized facial recognition system. *Neutrosophic sets and systems*, 34, 159-168. https://digitalrepository.unm.edu/nss_journal/vol34/iss1/21

Banks, J. (1998). *Handbook of simulation: principles, methodology, advances, applications, and practice*. Wiley.

Barbosa, P. A. M., Pinheiro, P. R., Silveira, F. R. V., & Filho, M. S. (2019). Selection and Prioritization of Software Requirements Applying Verbal Decision Analysis. *Complexity*, 2019, 1–20. <https://doi.org/10.1155/2019/2306213>

Barud, N. A., Oliveira, R. A. de, Gomes, C. F. S., Sanseverino, A. M., Barcelos, M. R. dos S., & Santos, M. dos. (2021). Lean in information technology departments or companies: identifying publications on the Scopus and Web of Science databases. *Scientometrics*, 126, 2437-2457. <https://doi.org/10.1007/s11192-020-03662-8>

Beleski, I. M.; Lima, F. R., Junior. (2023). Um estudo comparativo entre métodos Hesitant Fuzzy TOPSIS no contexto de seleção de fornecedores. *Revista Eletrônica E-xacta*. <https://doi.org/10.5585/2023.23218>

Bukhsh, F. A., Bukhsh, Z. A., & Daneva, M. (2020). A systematic literature review on requirement prioritization techniques and their empirical evaluation. *Computer Standards & Interfaces*, 69. <https://doi.org/10.1016/j.csi.2019.103389>

Chen, C.-T. (2000). Extensions of the TOPSIS for group decision-making under fuzzy environment. *Fuzzy Sets and Systems*, 114(1), 1–9. [https://doi.org/10.1016/s0165-0114\(97\)00377-1](https://doi.org/10.1016/s0165-0114(97)00377-1)

Dymova, L., Kaczmarek, K., Sevastjanov, P., & Kulawik, J. (2022). A Fuzzy Multiple Criteria Decision Making Approach with a Complete User Friendly Computer Implementation. *Entropy*, 23(2), 1–28. <https://doi.org/10.3390/e23020203>

Elacoste, T. S., Machado, C. M. dos S., Longaray, A. A., & Gomes, L. F. A. M. (2022). A fuzzy scale approach to the THOR algorithm. *Pesquisa Operacional*, 1-25. <https://doi.org/10.1590/0101-7438.2022.042.00261547>

Fernandes, J. M., Rodrigues, S. P., & Costa, L. A. (2015). Comparing AHP and ELECTRE I for prioritizing software requirements. 2015 IEEE/ACIS 16th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 1-8. <https://doi.org/10.1109/SNPD.2015.7176282>

Gerogiannis, V. C., & Tzikas, G. (2017). Using Fuzzy Linguistic 2-Tuples to Collectively Prioritize Software Requirements based on Stakeholders' Evaluations. *Proceedings of the 21st Pan-Hellenic Conference on Informatics*, 1-6. <https://doi.org/10.1145/3139367.3139430>

Gomes, C. F. S., Teixeira, L. F. H. D. S. D. B., & Santos, M. (2020). Sapevo Web / Auxílio à Decisão Multicritério / Pesquisa Operacional - Engenharia de Produção. BR 51 2020 000667-1.

Gomes, C.F.S., Santos, M., & Vanni, V.A.M. (2020). TOPSIS-2NE. *Aplicativos / Engenharia*. BR 51 2020 001956 0.

Hamdan, S., & Cheaitou, A. (2017). Supplier selection and order allocation with green criteria: An MCDM and multi-objective optimization approach. *Computers & Operations Research*, 81, 282–304. <https://doi.org/10.1016/j.cor.2016.11.005>

ISO/IEC 25000 (2014). Software engineering - System and software Quality Requirements and Evaluation (SQuaRE). <https://www.iso.org/standard/64764.html>

Lima, F. R., & Carpinetti, L. C. R. (2015). Uma comparação entre os métodos TOPSIS e Fuzzy-TOPSIS no apoio à tomada de decisão multicritério para seleção de fornecedores. *Gestão & Produção*, 22(1), 17–34. <https://doi.org/10.1590/0104-530x1190>

McZara, J., Sarkani, S., Holzer, T., & Eveleigh, T. (2014). Software requirements prioritization and selection using linguistic tools and constraint solvers - a controlled experiment. *Empirical Software Engineering*, 20(6), 1721–1761. <https://doi.org/10.1007/s10664-014-9334-8>

Montes, R., Sánchez, A. M., Villar, P., & Herrera, F. (2015). A web tool to support decision making in the housing market using hesitant fuzzy linguistic term sets. *Applied Soft Computing*, 35, 949-957. <https://doi.org/10.1016/j.asoc.2015.01.030>

Moraes, M., & Lima, F. R. (2017). Proposição e Aplicação de uma Metodologia baseada no AHP e na ISO/IEC 25000 para apoiar a Avaliação da Qualidade de Softwares de Gestão de Projetos. *Gestão Da Produção, Operações e Sistemas*, 12(2), 239-260. <https://doi.org/10.15675/gepros.v12i2.1653>

Mougouei, D., Ghose, A., Dam, H., Fahmideh, M., & Powers, D. (2021). A Fuzzy-Based Requirement Selection Method for Considering Value Dependencies in Software Release Planning. 2021 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). <https://doi.org/10.1109/fuzz45933.2021.9494422>

Muhammad, A., Siddique, A., Muhammad, M., Aldweesh, A., & Naveed, Q. N. (2017). Prioritizing non-Functional Requirements in Agile Process using Multi Criteria Decision Making Analysis. *IEEE Access*, 11, 24631-24654. <https://doi.org/10.1109/ACCESS.2023.3253771>

Nazim, M., Mohammad, C., & Sadiq, M. (2022). A comparison between fuzzy AHP and fuzzy TOPSIS methods to software requirements selection. *Alexandria Engineering Journal*, 61(12), 10851–10870. <https://doi.org/10.1016/j.aej.2022.04.005>

Onar, S. C., Oztaysi, B., & Kahraman, C. (2017). Hesitant Fuzzy Evaluation of System Requirements in Job Matching Platform Design. *Advances in Fuzzy Logic and Technology* 2017, 48-55. https://doi.org/10.1007/978-3-319-66827-7_5

Pacheco, C., García, I., & Reyes, M. (2018). Requirements elicitation techniques: a systematic literature review based on the maturity of the techniques. *IET Software*, 12(4), 365–378. <https://doi.org/10.1049/iet-sen.2017.0144>

Rodríguez, R. M., Martínez, L., & Herrera, F. (2013). A group decision making model dealing with comparative linguistic expressions based on hesitant fuzzy linguistic term sets. *Information Sciences*, 241, 28–42. <https://doi.org/10.1016/j.ins.2013.04.006>

Sadiq, M., & Jain, S. K. (2014). Applying fuzzy preference relation for requirements prioritization in goal-oriented requirements elicitation process. *International Journal of System Assurance Engineering and Management*, 5(4), 711–723. <https://doi.org/10.1007/s13198-014-0236-3>

Sadiq, M., Khan, S., & Mohammad, C. W. (2020). Selection of software requirements using TOPSIS under fuzzy environment. *International Journal of Computers and Applications*, 1–10. <https://doi.org/10.1080/1206212x.2020.1820689>

Sadiq, M., Parveen, A., & Jain, S. K. (2021). Software Requirements Selection with Incomplete Linguistic Preference Relations. *Business & Information Systems Engineering*, 63, 669-882. <https://doi.org/10.1007/s12599-021-00696-x>

Silva, M. do C., Gomes, C. F. S., & Souza, R. C. (2020). TOPSIS-2NEs Proposal. *International Journal of Fuzzy Systems*, 22, 1118-1122. <https://doi.org/10.1007/s40815-020-00871-4>

Tahri, M., Maanan, M., Tahri, H., Kašpar, J., Purwestri, R., Mohammadi, Z., & Marušák, R. (2022). New Fuzzy-AHP Matlab based graphical user interface (GUI) for a broad range of users: Sample applications in the environmental field. *Computers & Geosciences*, 158, 104951. <https://doi.org/10.1016/j.cageo.2021.104951>

Tiwari, S., Rathore, S. S., & Gupta, A. (2012). Selecting requirement elicitation techniques for software projects. 2012 CSI Sixth International Conference on Software Engineering (CONSEG), 15559481. <https://doi.org/10.1109/CONSEG.2012.6349486>

Tominaga, L. K. de G., Martins, V. W. B., Rampasso, I. S., Anholon, R., Silva, D., Pinto, J. S., Leal Filho, W., & Lima, F. R. (2020). Critical analysis of engineering education focused on sustainability in supply chain management: an overview of Brazilian higher education institutions. *International Journal of Sustainability in Higher Education*, 22(2), 380–403. <https://doi.org/10.1108/ijshe-01-2020-0002>