08, 09, 10 e 11 de novembro de 2022
ISSN 2177-3866

APLICAÇÃO DO MÉTODO FUZZY TOPSIS PARA PRIORIZAÇÃO DOS REQUISITOS DE UM NOVO SOFTWARE DE DECISÃO MULTICRITÉRIO

TAINARA SILVA NOVAES
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ (UTFPR)

VIVIANE RUOTOLO
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ (UTFPR)

DAHAN POIEL LIMA SCHUSTER
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ (UTFPR)

LUCYANO CAMPOS MARTINS
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO TOCANTINS (IFTO)

FRANCISCO RODRIGUES LIMA JUNIOR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ (UTFPR)

Agradecimento à orgão de fomento:
O presente trabalho foi financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPQ (processo 409529/2021-4).

APLICAÇÃO DO MÉTODO *FUZZY* TOPSIS PARA PRIORIZAÇÃO DOS REQUISITOS DE UM NOVO SOFTWARE DE DECISÃO MULTICRITÉRIO

1. INTRODUÇÃO

O processo de criação de novos produtos de software é algo complexo. Em função disso, esse processo costuma ser dividido nas etapas de diagnóstico, concepção, levantamento e análise de requisitos, desenvolvimento e manutenção (PRESSMAN; MAXIM, 2006). Uma vez que essas etapas são interdependentes, é importante dedicar esforços às etapas iniciais que antecedem a etapa desenvolvimento do software, a fim de fazer melhor uso dos recursos e garantir que o produto resultante alcance o nível de qualidade almejado pelo cliente final (PACHECO; GARCÍA; REYES, 2018).

Uma etapa de suma importância consiste no levantamento e análise de requisitos do 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 et al., 2015). A noção de prioridade e a capacidade de escolha são habilidades essenciais na engenharia de *software.* Frequentemente, é necessário identificar uma larga escala de requisitos do sistema, embora muitas vezes a implementação de todos eles não seja possível, dado que dependem de fatores como o tempo e os recursos financeiros da equipe desenvolvedora (AFZAL; SADIM, 2018). Dessa forma, observa-se a importância da priorização adequada a fim de obter um sistema com escala menor de requisitos, boa qualidade, 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; BARBOSA et al., 2019). Enquanto algumas delas focam na elicitação dos requisitos, outras são voltadas para a priorização (ou escolha) destes. Essa priorização é especialmente útil quando se deseja estabelecer uma listagem de requisitos essenciais para a criação de um produto mínimo viável, ou seja, para o lançamento de uma versão inicial do software com as funcionalidades essenciais, que poderá ser atualizada e melhorada no decorrer do tempo (ALBUGA; ODEH, 2018).

A priorização de requisitos de software é frequentemente abordada na literatura como um problema de tomada de decisão multicritério. Ainda que existam vários estudos que proponham modelos multicritério para auxiliar na priorização de requisitos de software (SADIQ et al., 2021; NAZIM et al., 2022), por meio do levantamento bibliográfico realizado pelo presente estudo, constatou-se a ausência de trabalhos voltados para priorização de requisitos de software de apoio à tomada de decisão multicritério.

1.1 Contexto Investigado e Diagnóstico da situação-problema

Softwares de apoio à decisão multicritério são ferramentas importantes para apoiar gestores em processos decisórios inerentes a diversas áreas da gestão empresarial. Esses softwares são aplicáveis em problemas que envolvam a avaliação de duas ou mais alternativas com base em dois ou mais critérios de decisão (LIMA JUNIOR et al., 2012; MONTES et al., 2015). Embora existam diversos softwares baseados em técnicas multicritério tradicionais, como M-MACBETH, *Expert Choice* e *Super Decision* (LIMA JUNIOR et al., 2012), a partir do levantamento realizado neste estudo em bases de periódicos e de patentes, constatou-se que há poucos softwares (MONTES et al., 2015; HAMDAN; CHEAITOU, 2017; DYMOVA et al., 2021; TAHRI et al., 2022) de apoio à tomada multicritério baseadas na Teoria dos Conjuntos *Fuzzy* e nas extensões mais recentes desta teoria.

Diante do crescente aumento de uso das técnicas *fuzzy* em artigos acadêmicos e na área empresarial (KAHRAMAN; OZTAYSI; ÖNAR, 2016), um grupo de pesquisadores brasileiros, financiados pelo Edital Universal do CNPQ (Conselho Nacional de Desenvolvimento Científico e Tecnológico), iniciou o desenvolvimento de um software de tomada de decisão

multicritério, que permitirá automatizar a aplicação do método *Hesitant Fuzzy Linguistic* TOPSIS. A equipe de desenvolvimento do software é composta por três alunos de Iniciação Científica de um curso de Engenharia de Computação, um profissional da área de desenvolvimento de software, que também é professor de um Curso de Ciência de Computação, e um professor da área de Engenharia de Produção. Todos os membros participam de um grupo de pesquisa voltado à tomada de decisão multicritério.

Inicialmente a equipe do projeto realizou um levantamento na base de patentes do INPI (Instituto Nacional da Propriedade Industrial) e nas principais bases de periódicos internacionais, a fim de identificar e analisar os softwares já existentes para tomada de decisão multicritério. Os resultados das buscas nas bases de dados retornaram apenas quatro estudos (MONTES et al., 2015; HAMDAN; CHEAITOU, 2017; DYMOVA et al., 2021; TAHRI et al., 2022) baseados na Teoria dos Conjuntos *Fuzzy* e/ou nas extensões dessa teoria. Já as buscas na base do INPI retornaram apenas três patentes de softwares para decisão multicritério (GOMES et al., 2020a; GOMES et al., 2020b; SANTOS et al., 2020), sendo que nenhum deles é baseado em técnicas *Fuzzy*.

A equipe identificou apenas dois softwares baseados em técnicas *Hesitant Fuzzy*, embora nenhum deles utilize o método *Hesitant Fuzzy Linguistic* TOPSIS. Diante disso, percebeu-se a escassez de softwares de apoio à tomada de decisão multicritério capazes de apoiar os decisores em situações de hesitação, ou seja, em cenários em que a incerteza e a ausência de informações disponíveis sobre o problema contribuem para que os decisores hesitem ao exprimir seus julgamentos. Nesses casos, os decisores podem preferir expressar seus julgamentos de modo mais flexível, utilizando expressões linguísticas (como “entre médio e alto” ou “no máximo alto”), o que é possível por meio do uso do método *Hesitant Fuzzy Linguistic* TOPSIS (RODRÍGUEZ; MARTÍNEZ; HERRERA, 2013).

Diante da oportunidade de criar um novo software para suprir essa lacuna, surgiu a necessidade de estabelecer as funcionalidades e outras características técnicas desejáveis a este sistema. Nesse contexto, o presente estudo apresenta uma aplicação do método *Fuzzy* TOPSIS para realizar a priorização dos requisitos desse novo software de apoio à tomada de decisão multicritério. 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 do sistema.

Além da contribuição prática para o projeto do software em questão, a partir das buscas realizadas nas bases de periódicos supracitadas, verificou-se também a ausência de estudos voltados para priorização de requisitos de software de tomada de decisão multicritério, o que evidenciou uma oportunidade de pesquisa. A literatura apresenta vários modelos baseados em técnicas de decisão com o intuito de apoiar o processo de priorização de requisitos de softwares. Para explicitar as técnicas usadas, o Quadro 1 apresenta diversos estudos que realizaram aplicações dessas técnicas em problemas de seleção ou priorização de requisitos de *software.*

Quadro 1 – Estudos que aplicam métodos multicritério para priorização ou seleção de requisitos de software

Autor	**Técnica(s) utilizada(s)**	**Aplicação**
Afzal e Sadim (2018)	AHP	Software de examinação do desempenho acadêmico
Ahmad et al. (2017)	*Fuzzy* MoSCoW	Software de gerenciamento de biblioteca
Albuga e Odeh (2018)	KANO model	*Softwares* comerciais de startups
Barbosa et al. (2019)	Análise de Decisão Verbal	Definição da próxima versão de um *software*
Kumari e Srinivas (2016)	Algoritmos Evolutivos de Inspiração Quântica	Avaliação do funcionamento dos algoritmos evolutivos na seleção de requisitos de *software*
McZara et al. (2015)	*Linguistic Tools and Constraint Solvers*	Priorização de grandes conjuntos de requisitos
Mougouei e Powers (2021)	*Fuzzy Graph e Integer Programming*	Identificação das dependências entre os requisitos de *software* e selecioná-los com maior qualidade
Nazim et al. (2022)	*Fuzzy* AHP e *Fuzzy* TOPSIS	Sistema de examinação do desempenho acadêmico

Autor	Técnica(s) utilizada(s)	Aplicação
Pitangueira et al. (2015)	*Search-Based Software Engineering Approaches* (SBSE)	Investigar quais aspectos da seleção de requisitos são abordados pelo método SBSE
Sadiq e Afrin (2017)	*Fuzzy* AHP e *Fuzzy* TOPSIS	Sistema de examinação do desempenho acadêmico
Sadiq e Jain (2014)	*Fuzzy* AHP	Sistema de examinação do desempenho acadêmico
Sadiq et al. (2020)	*Fuzzy* TOPSIS	Sistema de examinação do desempenho acadêmico
Sadiq et al. (2021)	*Fuzzy Preference Relation*	Sistema de examinação do desempenho acadêmico

Fonte: Elaborado pelos autores.

As técnicas anteriormente aplicadas incluem AHP, *Fuzzy* AHP, *Fuzzy* TOPSIS, *Fuzzy Graph*, Kano model, *Fuzzy* MoSCoW (acrônimo para as expressões: *must-have*, *should-have*, *could-have*, *and won't-hav*e), entre outras. É notável o vasto uso dos métodos baseados na Teoria dos Conjuntos *Fuzzy*. Dentre eles, o *Fuzzy* TOPSIS se destaca pela sua simplicidade de implementação e por não possuir limitações na quantidade de alternativas e critérios que podem ser considerados no problema de decisão. Nesse método, a escala para avaliação das alternativas e critérios é mais simples de usar do que as escalas comparativas utilizadas no AHP e *Fuzzy* AHP, além de não requerer a realização de testes de consistência dos julgamentos (LIMA JUNIOR; CARPINETTI, 2015). Em função disso, o método *Fuzzy* TOPSIS foi escolhido para ser aplicado neste estudo. Embora o *Fuzzy* TOPSIS tenha sido aplicado de forma bem sucedida para priorização de requisitos de sistemas de examinação do desempenho acadêmico (SADIQ; AFRIN, 2017; SADIQ et al., 2020; NAZIM et al., 2022), não foi encontrada nenhuma aplicação voltada ao desenvolvimento de software de apoio à tomada de decisão multicritério.

O restante deste estudo está organizado como segue. A Seção 2 apresenta os procedimentos adotados para a intervenção proposta pelo presente estudo. A Seção 3 apresenta um referencial teórico sobre o processo de definição de requisitos de software e sobre a ISO/IEC 25000. A Seção 4 apresenta e discute os resultados da aplicação do método *Fuzzy* TOPSIS para priorização dos requisitos de um software de apoio à decisão multicritério. Por fim, a Seção 5 discute as contribuições tecnológicas-sociais e propõe sugestões para estudos futuros.

2. INTERVENÇÃO PROPOSTA

A intervenção proposta por este 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. Para esse autor, a simulação é a representação do modo de operação de um sistema, em que se observa tal cenário a fim de obter resultados aplicáveis ao sistema real representado. A metodologia de modelagem e simulação se baseia na construção de modelos compostos por variáveis de entrada e de saída que possuem relacionamentos quantificáveis entre si (MAGALHÃES; LIMA JUNIOR, 2021). 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 valor de prioridade global de cada requisito.

As etapas adotadas para o desenvolvimento do presente estudo são descritas a seguir:

i) Pesquisa bibliográfica: realizou-se um levantamento bibliográfico sobre estudos que abordam a definição e priorização de requisitos de software, a norma ISO/IEC 25000 e o método *Fuzzy* TOPSIS. Foram consultadas as bases *Science Direct*, *Springer*, *Taylor & Francis* e *IEEE-Xplore*, bem como a ferramenta de busca Google Scholar. Os resultados da busca incluíram trabalhos publicados em periódicos, eventos científicos e dissertações de mestrado. As buscas foram realizadas utilizando combinações entre as palavras-chave: *Fuzzy*, *Hesitant Fuzzy*, *Fuzzy* TOPSIS, *Graphical User Interface*, *software requirements*, *software requirements engineering*, *software requirements elicitation*, *Fuzzy software*, MCDM, *Multicriteria*, Multicritério, *software quality*,

dentre outras. Também foi feita uma busca na base de patentes do INPI e nas bases citadas anteriormente, por meio do uso das palavras-chave *Fuzzy*, *Hesitant Fuzzy*, *Decisão*, Multicritério e MCDM, a fim de identificar softwares com interface gráfica baseados em técnicas multicritério *Fuzzy*. O levantamento bibliográfico realizado permitiu mapear as aplicações de técnicas multicritério para priorização de requisitos de software, como propiciou o entendimento das características e subcaracterísticas da qualidade de software, o que subsidiou a definição dos requisitos pelos decisores e a condução da aplicação realizada posteriormente;

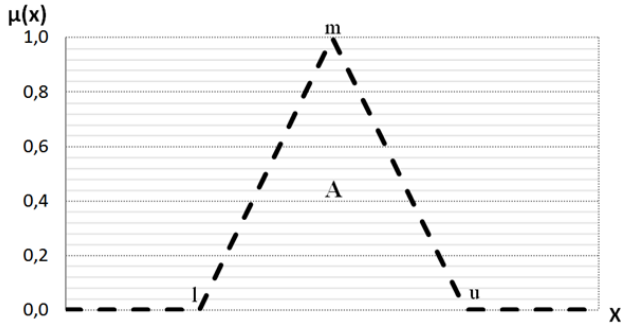
ii) Coleta de dados: inicialmente os requisitos foram definidos por três pesquisadores do projeto de desenvolvimento de software, que atuam como decisores na aplicação apresentada neste estudo. A definição dos requisitos foi feita por meio de um *brainstorming* para levantar as funcionalidades requeridas do software. Também se embasou nas patentes (GOMES et al., 2020a; GOMES et al., 2020b; SANTOS et al., 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 decisão multicritério, sejam eles baseados na Teoria dos Conjuntos *Fuzzy* ou nas extensões desta teoria. 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. Após a coleta dos julgamentos dos decisores, esses dados foram tabulados e inseridos no modelo *Fuzzy* TOPSIS;

iii) Modelagem e aplicação: nessa etapa, 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 2.1). Os dados coletados dos decisores foram inseridos no modelo para obter o ranqueamento dos requisitos segundo o nível 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 os valores do coeficiente de proximidade, que indica a prioridade global de cada requisito. Os resultados da aplicação foram analisados pelos decisores, que então definiram quais requisitos serão considerados na implementação computacional da primeira versão do software de apoio à decisão multicritério.

2.1 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. Dentre as técnicas multicritério baseadas em FST, o método *Fuzzy* TOPSIS é um dos mais aplicados devido a sua versatilidade e facilidade de implementação e uso (MAGALHÃES; LIMA JR., 2021; TOMINAGA et al., 2021). Esse método foi desenvolvido por Chen (2000) e permite aos decisores emitir seus julgamentos em formato linguístico (como “alto” e “baixo”) para avaliar as pontuações das alternativas do problema. De acordo com Chen (2000), os julgamentos dos decisores são quantificados por números *fuzzy* triangulares, conforme exemplifica a Figura 1. 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* (MAGALHÃES; LIMA JUNIOR., 2021).

Figura 1 - Número *fuzzy* triangular



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

Fonte: Magalhães e Lima Junior (2021)

$$\tilde{A} + \tilde{B} = [l_A, m_A, u_A] + [l_B, m_B, u_B] = [l_A + l_B,\ m_A + m_B, u_A + u_B] \quad (1)$$
$$\tilde{A} - \tilde{B} = [l_A, m_A, u_A] - [l_B, m_B, u_B] = [l_A - u_B, m_A - m_B, u_A - l_B] \quad (2)$$
$$\tilde{A} * \tilde{B} = [l_A, m_A, u_A] * [l_B, m_B, u_B] = [l_A * l_B, m_A * m_B, u_A * u_B] \quad (3)$$
$$\tilde{A} / \tilde{B} = [l_A, m_A, u_A]/[l_B, m_B, u_B] = [l_A/u_B, m_A/m_B, u_A/l_B] \quad (4)$$

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 tomada de decisão), de acordo com a opinião de múltiplos decisores. Dessa forma, passa a ser possível considerar o peso da opinião dos decisores, de acordo com seus níveis de experiência e/ou responsabilidade a respeito do processo decisório em questão. Considerando isso, os seguintes passos são descritos a seguir (TOMINAGA et al., 2021):

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

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

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

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;

$$\tilde{R} = [\tilde{r}_{ij}]_{m\ x\ n} \quad (7)$$
$$\tilde{r}_{ij} = \left(\frac{l_{ij}}{u_j^+}, \frac{m_{ij}}{u_j^+}, \frac{u_{ij}}{u_j^+}\right), \text{sendo } u_j^+ = \max_i u_{ij} \quad (8)$$

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 $\tilde{r}_{ij}$;

$$\tilde{V} = [\tilde{v}_{ij}]_{m\ x\ n} \quad (9)$$
$$\tilde{v}_{ij} = \tilde{r}_{ij} * \tilde{w}_j \quad (10)$$

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 $\tilde{v}_j^- = (0,0,0)$;

$$A^+ = \{\tilde{v}_1^+, \tilde{v}_j^+, \ldots, \tilde{v}_m^+\} \quad (11)$$
$$A^- = \{\tilde{v}_1^-, \tilde{v}_j^-, \ldots, \tilde{v}_m^-\} \quad (12)$$

v) Calcular as distâncias entre as pontuações dos requisitos e os valores de FPIS (D_i^+), bem em relação aos valores de FNIS (D_i^-). Nesse passo, usam-se as 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 *CCi*, melhor é a prioridade global do requisito.

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

3. REFERENCIAL TEÓRICO

3.1 Processo de Definição de Requisitos de Software

A Engenharia de Software tem como base uma parte crucial do planejamento, chamada Engenharia de Requisitos. Segundo Thayer (1997), a Engenharia de Requisitos estuda a elicitação, análise, verificação, gerenciamento e documentação dos requisitos de software. Cada uma dessas preocupações se torna uma atividade importante do processo de definição dos requisitos. Como explica Thayer (1997), na elicitação o cliente é ouvido e pode dizer “o quê” o sistema deve fazer. Essa etapa não é concluída uma única vez de forma estática, sendo necessária uma interação constante com o cliente para que novas ideias e perspectivas sejam acolhidas. Todas as demais etapas da Engenharia de Requisitos podem conter elicitação (BELGAMO; MARTINS, 2008).

Para que seja possível melhorar a qualidade do sistema, é importante apriomorar anteriormente a qualidade dos requisitos elicitados, de determinar uma boa técnica de obtenção desses requisitos dos clientes (PACHECO; GARCÍA; REYES, 2018). Nesse contexto, é essencial que, ao buscar ao buscar otimizar o processo de elicitação dos requisitos, certas questões venham à tona para que a escolha seja direcionada de forma coerente. De acordo com Tiwari, Rathore e Gupta (2012), algumas dessas questões são: "Quais os requisitos funcionais e não funcionais? Quais as limitações do sistema no contexto de *hardware* e *software*? Quais tipos de usuários estão envolvidos no sistema?”.

Segundo Tiwari, Rathore e Gupta (2012), diversas técnicas podem ser usadas para apoiar o processo de elicitação de requisitos. As abordagens tradicionais envolvem entrevistas, questionários e pesquisas com clientes e usuários. Há também a técnica denominada

brainstorming, que busca juntar ideias de diferentes decisores e analisá-las. Ainda segundo os autores, a escolha de uma única abordagem limita o processo de elicitação e não considera a natureza mutável dos requisitos.

Pensando na prevenção de mudanças bruscas nos requisitos do sistema, faz-se necessária uma abordagem científica e adaptável a mudanças para extrair do usuário final os requisitos mais importantes do sistema. Asfora (2009) afirma que, dando liberdade ao cliente para escolher quais requisitos o sistema deve atender, o projeto torna-se passível de erros que impactam a qualidade global do sistema e sua aceitação pelos usuários finais. É importante, portanto, possuir uma técnica bem definida para priorizar, dentre os requisitos listados, aqueles que mais refletem a real necessidade do cliente.

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 da maneira informal, mas sim tratada como um problema de tomada de decisão (MARIA et al, 2011). A seção a seguir discute alguns modelos que foram propostos na literatura para priorização de requisitos de softwares.

3.2 ISO/IEC 25000

A qualidade de software é uma vertente da engenharia de software que consiste, segundo Pressman e Maxim (2016), na conformidade dos requisitos funcionais e do desempenho de um software a padrões de desenvolvimento documentados e características implícitas previstas em um bom produto de software. Existem modelos de qualidade que permitem a avaliação de softwares de acordo com critérios predefinidos. Esses modelos podem ser úteis no desenvolvimento de um novo software ou na avaliação da qualidade de produtos existentes (MORAIS; LIMA JUNIOR, 2017).

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 (DUARTE FILHO, 2011). Ela é a evolução das normas ISO/IEC 9126 e da ISO/IEC 14598, dada a difícil compreensão e certa confusão dos usuários em relação a essas normas precedentes. A SQuaRE, portanto, tem por característica intrínseca, ser uma norma mais organizada e unificada com duas dimensões: especificação de requisitos e avaliação de qualidade de um produto de software (DUARTE FILHO, 2011).

A série de normas possui uma arquitetura composta por 5 divisões, com suas respectivas especificidades. A primeira delas é a ISO/IEC 2500n, que diz respeito ao gerenciamento de qualidade e define todos os termos, documentos e modelos presentes em todos os outros segmentos da série SQuaRE. A ISO/IEC 2501n corresponde à norma precedente ISO/IEC 9126, e oferece modelos de qualidade detalhados para avaliação de produtos de software. Atualmente, essa divisão é composta por outras duas subdivisões: a ISO/IEC 25010 e a ISO/IEC 25012. A ISO/IEC 25010 descreve o modelo de software em detalhes, e define características que permitam avaliar a qualidade do software e a qualidade em uso. Já a ISO/IEC 25012 consiste no modelo de qualidade de dados, que define um formato estruturado de dados que são retidos num software (ISO/IEC25000, 2014).

A terceira divisão se refere à ISO/IEC 2502n, cujo enfoque é a medição de qualidade e definição de métricas matemáticas para qualidade de software. A divisão ISO/IEC 2503n orienta durante o levantamento e especificação de requisitos. Já a ISO/IEC 2504n fornece documentos, metodologias, diretrizes e recomendações direcionados diretamente para avaliação de qualidade de um sistema de computador (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 JUNIOR, 2017).

Quadro 2 – Características e subcaracterísticas da qualidade de software

Características	**Subcaracterísticas**	**Definição da subcaracterística**
Adequação funcional	Integridade funcional	Capacidade de atender às tarefas e aos objetivos específicos do usuário a que foi destinado
	Correção funcional	Capacidade de fornecer resultados corretos e com precisão.
	Adequação funcional	Capacidade de realizar tarefas e certos objetivos de maneira fácil.
Eficiência de desempenho	Comportamento em relação ao tempo	Capacidade de fornecer tempos de resposta e processamento apropriados quando o software executa suas funções.
	Utilização de recursos	Grau com que os tipos e as quantidades de recursos usados atende aos requisitos.
	Capacidade	Grau com que a capacidade máxima 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 que o software possui de ser usado para alcançar objetivos específicos de aprendizagem, de forma eficiente, eficaz e sem riscos, garantindo que o usuário se sinta satisfeito no contexto em questão.
	Operacionalidade	Capacidade do software de permitir ao usuário operá-lo e controlá-lo de forma fácil.
	Proteção de erro	Capacidade de proteger os usuários de cometer falhas.
	Estética da Interface	Capacidade de possuir uma interface que seja satisfatória ao usuário.
	Acessibilidade	Capacidade de ser usado por usuários com diferentes características e habilidades para alcançar objetivos especificados.
Compatibilida-de	Coexistência	Capacidade de compartilhar recursos com outro software sem causar impactos sobre qualquer outro produto.
	Interoperabilidade	Capacidade de interagir com um ou mais sistemas especificados.
Confiança	Maturidade	Capacidade de atender às necessidades de confiabilidade quando operado em condições normais.
	Disponibilidade	Capacidade de ser operacional e acessível quando requerido para uso.
	Tolerância a falhas	Capacidade de garantir um nível de desempenho especificado em caso de ocorrência de falhas de software ou hardware.
	Capacidade de Recuperação	Capacidade de restabelecer seu nível de desempenho especificado e recuperar os dados diretamente afetados no caso de uma falha.
Segurança	Confidencialidade	Capacidade de permitir acesso de dados somente a usuários autorizados.
	Integridade	Capacidade de bloquear acesso e modificações de usuários não autorizados.
	Não repúdio	Capacidade de comprovar ações, eventos, alterações e envio de informações para que não possam ser repudiados futuramente.
	Responsabilidade	Capacidade de rastrear as ações de entidades específicas.
	Autenticidade	Capacidade de comprovar a identidade de um sujeito ou recurso caso seja requerido.
Manutenibilida-de	Modularidade	Capacidade de alterar elementos do software com impacto mínimo.
	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	Capacidade de permitir que uma modificação seja implementada sem causar defeitos no produto existente.
	Testabilidade	Capacidade de estabelecer critérios de teste para um sistema que foi modificado e de determinar se esses critérios foram cumpridos de forma eficaz e eficiente.
Portabilidade	Adaptabilidade	Capacidade de ser adaptado para ambientes de operação especificados, sem a necessidade de aplicação de outras ações ou meios além daqueles fornecidos para essa finalidade pelo software considerado.

Características	Subcaracterísticas	Definição da subcaracterística
	Instalabilidade	Capacidade de ser corretamente instalado e / ou desinstalado em um ambiente especificado.
	Substituibilidade	Capacidade de substituir outro software no mesmo ambiente e para o mesmo fim.

Fonte: Morais e Lima Junior (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 em questão. Nesse sentido, essas subcaracterísticas servem como referência para a criação de modelos que sejam adequados ao contexto de uso (ISO/IEC25000, 2014; MORAIS; LIMA JUNIOR, 2017).

Morais e Lima Junior et al. (2017) também destacam que a maneira como algumas das subcaracterísticas são definidas pela ISO/IEC 25000 não permite sua medição direta, dada a subjetividade de interpretação e de medição destas subcaracterísticas. 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, a Seção 4 apresenta os resultados da priorização de requisitos de um novo software de tomada de decisão, os quais foram definidos de forma alinhada às características da ISO/IEC 25000.

4. RESULTADOS OBTIDOS

A aplicação se iniciou pela composição de uma equipe de três decisores, formada por membros de um projeto financiado pelo CNPQ que visa o desenvolvimento de um novo software de apoio à tomada de decisão. 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 na modelagem e aplicação de técnicas HFLTS. O objetivo principal desse projeto é criar um software de apoio à tomada de decisão multicritério, com base no método *Hesitant Fuzzy Linguistic* TOPSIS, a fim de permitir aos decisores a utilização de expressões linguísticas em processos decisórios. Dessa forma, o software será capaz de apoiar a tomada de decisão em situações de incerteza e hesitação, ou seja, quando há ausência de informações suficientes para realizar a avaliação de alternativas.

Inicialmente esses decisores realizaram a definição dos requisitos, baseando-se nas necessidades do projeto, na análise de estudos sobre técnicas HFLTS (RODRÍGUEZ; MARTINEZ; HERRERA, 2013) e no levantamento sobre softwares para tomada de decisão baseados na Teoria dos Conjuntos *Fuzzy* (MONTES et al., 2015; HAMDAN; CHEAITOU, 2017; DYMOVA et al., 2021; TAHRI et al., 2022). Esses requisitos foram definidos pelos decisores de forma alinhada às subcaracterísticas da qualidade sugeridas pela ISO 25000, de modo a contemplar subcaracterísticas relacionadas à funcionalidade, usabilidade, segurança, portabilidade, confiança, manutenibilidade, usabilidade e segurança. Posteriormente, os decisores realizaram a avaliação dos pesos dos requisitos de acordo com sua experiência e com as necessidades do projeto em questão.

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

Quadro 3 – Listagem de requisitos e avaliação linguística quanto aos pesos dos requisitos

Subcaracterística	**Requisitos (alternativas)**	**D_1**	**D_2**
Completude 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
	R_2. O software deve permitir o uso de critérios qualitativos para avaliação de alternativas	Muito alto	Muito a
	R_3. O software deve permitir o uso de critérios quantitativos para avaliação de alternativas	Muito alto	Muito a
	R_4. O desempenho das alternativas em critérios quantitativos deve ser avaliado pelos decisores utilizando valores numéricos, numa escala definida pelo decisor	Alto	Baixo
	R_5. Os pesos dos critérios devem ser avaliados pelos decisores utilizando julgamentos linguísticos (como baixo, médio ou alto)	Alto	Alto
Operacionalidade	R_6. Os decisores poderão escolher uma escala com 5 ou 7 termos linguísticos para pontuar as alternativas (MONTES et al., 2015)	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 a
	R_8. O software deve fornecer suporte ao cadastro de usuários	Alto	Médio
Confidencialidade	R_9. O administrador deve ter acesso ao cadastro, exclusão e alteração de contas dos usuários	Muito alto	Alto
Instalabilidade	R_{10}. O software deve funcionar em plataforma WEB (MONTES et al., 2015)	Muito alto	Muito a
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
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 a
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 a
	R_{14}. Na seção de "ajuda ao usuário", 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
	R_{15}. O software deve apresentar os principais elementos do problema em uma mesma página, de modo que o usuário consiga visualizá-los e manipulá-los facilmente (HAMDAN; CHEAITOU, 2017; TAHRI et al., 2022)	Médio	Muito a
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
Pesos atribuídos aos decisores:		Muito Alto	Alto

Fonte: Elaborado pelos autores.

A escala utilizada pelos decisores para avaliação dos pesos dos requisitos foi definida com base em Chen (2000) e Lima Junior e Carpinetti (2015). Essa escala é composta por cinco termos linguísticos, os quais são associados aos seguintes números *fuzzy* triangulares: "Muito Baixo (0; 0; 2,5)", "Baixo (0; 2,5; 5,0)", "Médio (2,5; 5,0; 7,5)", "Alto (5,0; 7,5; 10)" e "Muito Alto (7,5; 10; 10)". Já a escala utilizada para ponderar a opinião de cada decisor foi definida com base nos mesmos autores e termos linguísticos, porém com números *fuzzy* que variam entre 0 e 1: "Muito Baixo (0; 0; 0,25)", "Baixo (0; 0,25; 0,50)", "Médio (0,25; 0,50; 0,75)", "Alto (0,50; 0,75; 1)" e "Muito Alto (0,75; 1; 1)". Os julgamentos mostrados no Quadro 1 foram convertidos em números *fuzzy* triangulares de acordo com as escalas linguísticas apresentadas anteriormente, formando a matriz de decisão $\widetilde{D}$ e o vetor $\widetilde{W}$. Os resultados da conversão são apresentados na Tabela 1.

Tabela 1 – Matriz de decisão e vetor de pesos convertidos em números *fuzzy*

		D₁			**D₂**			**D₃**		
		l	**m**	**u**	**l**	**m**	**u**	**l**	**m**	**u**
Matriz de decisão ($\widetilde{D}$) - Equação 5:	R_1	7,5	10,0	10,0	5,0	7,5	10,0	7,5	10,0	10,0
	R_2	7,5	10,0	10,0	7,5	10,0	10,0	7,5	10,0	10,0
	R_3	7,5	10,0	10,0	7,5	10,0	10,0	7,5	10,0	10,0
	R_4	5,0	7,5	10,0	0,0	2,5	5,0	2,5	5,0	7,5
	R_5	5,0	7,5	10,0	5,0	7,5	10,0	5,0	7,5	10,0
	R_6	5,0	7,5	10,0	5,0	7,5	10,0	5,0	7,5	10,0
	R_7	5,0	7,5	10,0	7,5	10,0	10,0	7,5	10,0	10,0
	R_8	5,0	7,5	10,0	2,5	5,0	7,5	2,5	5,0	7,5
	R_9	7,5	10,0	10,0	5,0	7,5	10,0	5,0	7,5	10,0
	R_{10}	7,5	10,0	10,0	7,5	10,0	10,0	0,0	2,5	5,0
	R_{11}	5,0	7,5	10,0	5,0	7,5	10,0	0,0	2,5	5,0
	R_{12}	5,0	7,5	10,0	7,5	10,0	10,0	5,0	7,5	10,0
	R_{13}	5,0	7,5	10,0	7,5	10,0	10,0	5,0	7,5	10,0
	R_{14}	5,0	7,5	10,0	2,5	5,0	7,5	2,5	5,0	7,5
	R_{15}	2,5	5,0	7,5	7,5	10,0	10,0	5,0	7,5	10,0
	R_{16}	2,5	5,0	7,5	5,0	7,5	10,0	5,0	7,5	10,0
Pesos dos decisores ($\widetilde{W}$) - Equação 6:		0,75	1,0	1,0	0,5	0,75	1,0	0,5	0,75	1,0

Fonte: Elaborado pelos autores.

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. Posteriormente, 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	**D₁**			**D₂**			**D₃**		
	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, foram calculadas as distâncias entre as pontuações dos requisitos e as soluções ideais positiva e negativa. A Tabela 3 apresenta os resultados do cálculo das distâncias entre as pontuações dos requisitos e os valores de A^+, os quais foram gerados pela aplicação das Equações 13 e 15. Já a Tabela 4 apresenta os resultados da aplicação das Equações 14 e 15 no cálculo das distâncias entre as pontuações dos requisitos e os valores de A^-.

Tabela 3 – Resultados do cálculo das distâncias das alternativas em relação a A^+

Requisitos	**D_1**	**D_2**	**D_3**	**D^+**
R_1	0,253	0,501	0,389	1,143
R_2	0,253	0,389	0,389	1,030
R_3	0,253	0,389	0,389	1,030
R_4	0,389	0,798	0,637	1,824
R_5	0,389	0,501	0,501	1,391
R_6	0,389	0,501	0,501	1,391
R_7	0,389	0,389	0,389	1,166
R_8	0,389	0,637	0,637	1,663
R_9	0,253	0,501	0,501	1,255
R_{10}	0,253	0,389	0,798	1,439
R_{11}	0,389	0,501	0,798	1,688
R_{12}	0,389	0,389	0,501	1,279
R_{13}	0,389	0,389	0,501	1,279
R_{14}	0,389	0,637	0,637	1,663
R_{15}	0,569	0,389	0,501	1,459
R_{16}	0,569	0,501	0,501	1,572

Fonte: Elaborado pelos autores.

Tabela 4 – Resultados do cálculo das distâncias das alternativas em relação a A^-

Requisitos	**D_1**	**D_2**	**D_3**	**D^-**
R_1	0,879	0,723	0,753	2,355
R_2	0,879	0,804	0,753	2,436
R_3	0,879	0,804	0,753	2,436
R_4	0,753	0,308	0,489	1,551
R_5	0,753	0,723	0,678	2,154
R_6	0,753	0,723	0,678	2,154
R_7	0,753	0,804	0,753	2,311
R_8	0,753	0,525	0,489	1,768
R_9	0,879	0,723	0,678	2,279
R_{10}	0,879	0,804	0,308	1,991
R_{11}	0,753	0,723	0,308	1,784
R_{12}	0,753	0,804	0,678	2,235
R_{13}	0,753	0,804	0,678	2,235
R_{14}	0,753	0,525	0,489	1,768
R_{15}	0,532	0,804	0,678	2,013
R_{16}	0,532	0,723	0,678	1,932

Fonte: Elaborado pelos autores.

Por fim, realizou-se o cálculo da prioridade global dos requisitos (CC_i) com base na Equação 16. A Tabela 5 apresenta os resultados finais da aplicação. Nessa tabela, os valores de CC_i foram ordenados decrescentemente, de acordo com a prioridade global dos requisitos. Houve empate em alguns requisitos no ranqueamento, já que alguns deles alcançaram valores idênticos de CC_i, como, por exemplo, R_2 e R_3, e R_{12} e R_{13}. Nota-se que os requisitos prioritários estão relacionados a funcionalidades essenciais do software, como permitir o uso de critérios qualitativos (R_2) e qualitativos (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.

Tabela 5 – Ranqueamento dos requisitos de software de acordo com a prioridade global (CC_i)

Classificação	CCi	Requisitos	Descrição dos requisitos
1º	0,703	R_2	O software deve permitir o uso de critérios qualitativos para avaliação de alternativas
1º	0,703	R_3	O software deve permitir o uso de critérios quantitativos para avaliação de alternativas
2º	0,673	R_1	O software deve permitir aos decisores a atribuição de pesos (níveis de importância) aos critérios
3º	0,665	R_7	O software precisa ser autoexplicativo em relação ao seu uso, orientando os decisores em relação a cada etapa de aplicação
4º	0,645	R_9	O administrador deve ter acesso ao cadastro, exclusão e alteração de contas dos usuários
5º	0,636	R_{12}	O código de programação do software deve conter comentários que expliquem as etapas implementadas, de modo a facilitar possíveis atualizações futuras
5º	0,636	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
6º	0,608	R_5	Os pesos dos critérios devem ser avaliados pelos decisores utilizando julgamentos linguísticos (como baixo, médio ou alto)
6º	0,608	R_6	Os decisores poderão escolher uma escala com 5 ou 7 termos linguísticos para pontuar as alternativas
7º	0,580	R_{10}	O software deve funcionar em plataforma WEB
7º	0,580	R_{15}	O software deve apresentar os principais elementos do problema em uma mesma página, de modo que o usuário consiga visualizá-los e manipulá-los facilmente
8º	0,551	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
9º	0,515	R_8	O software deve fornecer suporte ao cadastro de usuários (empresas compradoras)
9º	0,515	R_{14}	Na seção de "ajuda ao usuário", deve haver uma figura que represente a relação das teorias usadas como embasamento matemático para a realização dos cálculos do software
10º	0,514	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
11º	0,460	R_4	O desempenho das alternativas em critérios quantitativos deve ser avaliado pelos decisores utilizando valores numéricos, numa escala definida pelo decisor

Fonte: Elaborado pelos autores.

Os resultados mostrados na Tabela 5 foram apresentados aos decisores, que concordaram com a ordem de prioridade estabelecida para os requisitos por meio *Fuzzy* TOPSIS. Em acordo com a equipe desenvolvera 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. Dessa forma, os requisitos R_4, R_{11}, R_{14} e R_8 não estarão contemplados na primeira versão, devido a baixa relevância destes e a limitação de tempo para conclusão do projeto.

5. CONTRIBUIÇÃO TECNOLÓGICA-SOCIAL

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 apoio à decisão multicritério. 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. É importante destacar a capacidade desse método de apoiar a tomada de decisão em grupo e de possibilitar a atribuição de pesos aos julgamentos individuais dos decisores.

A principal contribuição social deste estudo está no apoio à priorização dos requisitos de um novo software, que está sendo desenvolvido por pesquisadores de diferentes universidades brasileiras por meio de um projeto financiado pelo CNPQ. Os próximos passos para o desenvolvimento do software incluem a elaboração de *storyboards* que relatem o processo de uso deste produto, o que será feito pelo coordenador e os pesquisadores do projeto tomando como base os requisitos prioritários. Utilizando esses *storyboards*, a equipe desenvolvedora construirá um conjunto de diagramas da área de engenharia de software, a fim de subsidiar a etapa de implementação computacional do sistema. Na última etapa do projeto, a listagem de requisitos prioritários servirá de base para a testagem do software em relação às funcionalidades requeridas pelos decisores. Portanto, os resultados apresentados neste estudo são fundamentais para apoiar as etapas posteriores de construção desse novo software de decisão multicritério.

Em relação às contribuições tecnológicas, o modelo implementado em MS Excel será disponibilizado na base Mendeley Data, e pode ser adaptado e reutilizado por outros pesquisadores e gestores que necessitem priorizar ou selecionar alternativas, considerando múltiplos critérios de decisão, com base em julgamentos linguísticos. 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. Essas aplicações podem envolver a priorização de requisitos ou a avaliação da qualidade de diferentes tipos, tais como softwares ERP (*Enterprise Resource Planning*), sistemas de gestão de projetos, sistemas de modelagem e gestão de processos, dentre outros. Para isso, recomenda-se que os decisores estabeleçam requisitos adequados ao contexto em questão, utilizando como base as métricas sugeridas pela ISO/IEC 25000.

REFERÊNCIAS

AFRIN, A.; SADIQ, M. An integrated approach for the selection of software requirements using fuzzy AHP and fuzzy TOPSIS method. **International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)**, p. 1094-1100, 2017.

AFZAL, N.; SADIM, M. Software requirements selection using AHP. **International Journal of Computer Science & Communication**, v. 9, p. 47–52, 2018.

AHMAD, K. S.; AHMAD, N.; TAHIR, H.;KHAN, S. Fuzzy_MoSCoW: A fuzzy based MoSCoW method for the prioritization of software requirements. **2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)**, p. 433-437, 2017.

ALBUGA, S.; ODEH, Y. Towards Prioritizing Software Business Requirements in Startups. **8th International Conference on Computer Science and Information Technology (CSIT)**, p. 257-265, 2018.

ASFORA, D. M. **Uma abordagem para a priorização de requisitos em ambientes ágeis**. 2009. 109p. Dissertação (Mestrado) - Universidade Federal de Pernambuco, Recife , 2009.

BANKS, Jerry. **Handbook of Simulation: principles, methodology, advances, applications, and practice.** Atlanta, Georgia: Emp Books, 1998.

BARBOSA, P. A. M.; PINHEIRO, P. R.; SILVEIRA, F. R. V.; FILHO, M. S. Selection and Prioritization of Software Requirements Applying Verbal Decision Analysis. **Complexity**. v. 2019, p. 20, 2019.

BELGAMO, A.; MARTINS, L. E. G. **Estudo Comparativo sobre as Técnicas de Elicitação de Requisitos do Software**. 2008. Artigo, Piracicaba-SP. Disponível em: <https://www.researchgate.net/profile/Luiz-Eduardo-Martins/publication/242602092_Estudo_Comparativo_sobre_as_Tecnicas_de_Elicitacao_de_Requisitos_do_Software/links/5513d4b60cf283ee0834923f/Estudo-Comparativo-sobre-as-Tecnicas-de-Elicitacao-de-Requisitos-do-Software.pdf>. Acesso em: 15 mai. 2022.

CHEN, C. Extensions of the TOPSIS for group decision-making under fuzzy environment. **Fuzzy sets and systems**, v. 114, n. 1, p. 1-9, 2000.

DORFMAN, M. System and Software Requirements Engineering. **IEEE Computer Society Press Tutorial**. p. 7-22, 1990.

DUARTE FILHO, N.F. **MAQSaas - Método para avaliação da qualidade em produtos SaaS**. 2011. 175 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Ciência da Computação, Universidade Federal de Minas Gerais, Belo Horizonte, 2011.

DYMOVA, L.; KACZMAREK, K.; SEVASTJANOV, P.; KULAWIK, J. A Fuzzy Multiple Criteria Decision Making Approach with a Complete User Friendly Computer Implementation. **Entropy**. v. 23, p. 1-28, 2021.

ESTRELLA, F.; RODRÍGUEZ, R.; MARTINEZ, L. A Hesitant Linguistic Fuzzy TOPSIS Approach Integrated into FLINTSTONES. **Atlantis Press**. p. 799-806, 2015.

GOMES, C. F. S. *et al*. **PROMETHEE-SAPEVO-M1 / Plataforma Computacional para Apoio à Tomada de Decisão / Processos Decisórios**. Titular: Centro de Análises de Sistemas Navais / Diretoria-Geral de Desenvolvimento Nuclear e Tecnológico da Marinha. Procurador: André Roberto dos Santos da Silva. BR 51 2020 002587 0. Depósito: 19 nov. 2020. Concessão: 24 nov. 2020a.

GOMES, C. F. S. *et al*. **Sapevo Web/ Auxílio à Decisão Multicritério/ Pesquisa Operacional - Engenharia de Produção**. Titular: Centro de Análises de Sistemas Navais / Diretoria-Geral de Desenvolvimento Nuclear e Tecnológico da Marinha. Procurador: André Roberto dos Santos da Silva. BR 51 2020 000667 1. Depósito: 14 abr. 2020. Concessão: 22 abr. 2020b.

HAMDAN, S.; CHEAITOU, A. Supplier selection and order allocation with green criteria: An MCDM and multi-objective optimization approach. **Computers & Operations Research**. v. 81, p. 282-304, 2017.

ISO/IEC 25000:2014. **Software engineering - System and software Quality Requirements and Evaluation (SQuaRE)**. Genebra: ISO, 2014.

KAHRAMAN, C.; ÖZTAYŞI, B.; ONAR, S. Ç. A Comprehensive Literature Review of 50 Years of Fuzzy Set Theory. **International Journal of Computational Intelligence Systems**. 9, p. 3-24, 2016.

KUMARI, A. C.; SRINIVAS, K. Comparing the performance of quantum-inspired evolutionary algorithms for the solution of software requirements selection problem. **Information and Software Technology**, v. 76, p.31-64, 2016.

LIMA JUNIOR, F.R.; CARPINETTI, L.C.R.; OSIRO, L.; GANGA, G.M. Avaliação da qualidade de softwares de apoio à decisão. In: ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO, 32, 2012, Bento Gonçalves, RS. **Anais...** Bento Gonçalves, RS: ENEGEP, 2012.

LIMA JUNIOR., F.R.; CARPINETTI, L.C.R. Uma comparação entre os métodos TOPSIS e Fuzzy-TOPSIS no apoio à tomada de decisão multicritério para seleção de fornecedores. **Gestão & Produção**, v. 22, p. 17-34, 2015.

MCZARA, J.; SARKANI, S.; HOLZER, T.; EVELEIGH, T. Software requirements prioritization and selection using linguistic tools and constraint solvers—a controlled experiment. **Empirical Software Engineering Journal**, v. 20, p. 1721–1761, 2015.

MONTES, R.; SÁNCHEZ, A. M.; VILLAR, P.; HERRERA, F. A web tool to support decision making in the housing market using hesitant fuzzy linguistic term sets. **Applied Soft Computing**. v. 35, p. 949-957, 2015.

MORAIS, M.H.B.M.; JUNIOR, F.R.L. 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. **GEPROS. Gestão da Produção, Operações e Sistemas**, n. 2, p. 239-260, 2017.

MOUGOUEI, D.; GHOSE, A.; DAM, H.; FAHMIDEH, M.; POWERS, D. A Fuzzy-Based Requirement Selection Method for Considering Value Dependencies in Software Release Planning. **IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)**. p. 1-6, 2021.

NAZIM, M.; MOHAMMAD, C.W.; SADIQ, M. A Comparison Between Fuzzy AHP And Fuzzy TOPSIS Methods To Software Requirements Selection. **Alexandria Engineering Journal**, p. 10851–10870, 2022.

PACHECO, C.; GARCÍA, I.; REYES, M. Requirements elicitation techniques: a systematic literature review based on the maturity of the techniques. **IET Software**, v. 12, n. 4, p. 365–378, 1 ago. 2018.

PITANGUEIRA, A., M.; MACIEL, R.S.P.M.; BARROS, M. Software requirements selection and prioritization using SBSE approaches: A systematic review and mapping of the literature. **Journal of Systems and Software**, v. 103, p. 267-280, 2015.

PRESSMAN, R. S.; MAXIM, B. **Engenharia de Software: Uma Abordagem Profissional**. 8. ed. Grupo A, 2016.

RODRÍGUEZ, R.M.; MARTINEZ, L.; HERRERA, F. A group decision making model dealing with comparative linguistic expressions based on hesitant fuzzy linguistic term sets. **Information Sciences**, v.221, p.28-42, 2013.

SADIQ, M.; JAIN, S.K. Applying fuzzy preference relation for requirements prioritization in goal oriented requirements elicitation process. **International Journal of System Assurance Engineering and Management.** v.5, p. 711–723, 2014.

SADIQ, M.; KHAN, S.; MOHAMMAD, C. W. Selection of software requirements using TOPSIS under fuzzy environment. **International Journal of Computers and Applications**. 2020.

SADIQ, M.; PARVEEN, A.; JAIN, S.K. Software Requirements Selection with Incomplete Linguistic Preference Relations. **Business & Information Systems Engineering**, v. 63, p. 669–688, 2021.

SANTOS, M. *et al*. **TOPSIS-2NE**. Titular: Vanildo Alexandre Meirelles Vanni. BR 51 2020 001956 0. Depósito: 23 set. 2020. Concessão: 29 set. 2020.

TAHRI, M.; MAANAN, M.; TAHRI, H.; KAŠPAR, J.; PURWESTRI, R. C.; MOHAMMADI, Z.; MARUŠÁK, R. New Fuzzy-AHP Matlab based graphical user interface (GUI) for a broad range of users: Sample applications in the environmental field. **Computers & Geosciences**. v. 158, 2022.

TIWARI, S.; RATHORE, S. S.; GUPTA, A. Selecting requirement elicitation techniques for software projects. **CSI 6th International Conference on Software Engineering**. p. 1-10, 2012.

TOMINAGA, L. K. G.; MARTINS, V. W. B.; RAMPASSO, I. S.; ANHOLON, R.; SILVA, D.; PINTO, J.S.; LEAL FILHO, W.; LIMA JUNIOR, F.R. 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**, v. 22, p. 380-403, 2021.