

## Desenvolvimento de um software de apoio à tomada de decisão em grupo sob incerteza

### Development of a software to support group decision making under uncertainty

Viviane Ruotolo<sup>1</sup>, Tainara Silva Novaes<sup>2</sup>, Dahan Poiel Lima Schuster<sup>3</sup>, Lucyano Campos Martins<sup>4</sup>, Francisco Rodrigues Lima Junior<sup>5</sup>

#### RESUMO

Mediante a escassez de softwares de apoio à tomada de decisão multicritério em grupo, voltados a situações de incerteza e hesitação, iniciou-se o desenvolvimento de um software com interface amigável, baseado em *Hesitant Fuzzy Linguistic TOPSIS* (Technique for Order Preference by Similarity to Ideal Solution), HFL-TOPSIS. Nesse contexto, surgiu a necessidade de estabelecer as funcionalidades e outras características técnicas desejáveis a esse sistema. O presente estudo apresenta os principais resultados do processo de implementação de um novo *software* de auxílio à tomada de decisões sob incerteza. Inicialmente, apresenta-se uma aplicação do método *Fuzzy TOPSIS* para priorizar os requisitos desse novo software. Esses requisitos foram definidos e avaliados por três pesquisadores. Os resultados serviram de base para a execução da modelagem e da implementação computacional, que foram realizadas com o apoio de diversas ferramentas de desenvolvimento. O presente estudo é o primeiro a aplicar uma técnica multicritério para priorização de requisitos de software de decisão multicritério. Também é o primeiro a propor um software para tomada de decisão baseado em HFL-TOPSIS.

**PALAVRAS-CHAVE:** decisão multicritério; *Fuzzy TOPSIS*; engenharia de software; requisitos.

#### ABSTRACT

Due to the scarcity of software to support group multi-criteria decision making, aimed at situations of uncertainty and hesitation, a research group developed a new software with a friendly interface, based on the *Hesitant Fuzzy Linguistic (HFL) TOPSIS* method. In this context, the need arose to establish the functionalities and other technical characteristics desirable for this system. The present study presents the main results of the development of a new software to aid decision-making under uncertainty. Initially, an application of the *Fuzzy TOPSIS* was carried out to prioritize the requirements of this software. These requirements were defined and evaluated by three researchers. The results served as the basis for the modeling and computational implementation, which were carried out with the support of several development tools. The present study is the first to apply a multi-criteria technique for prioritizing multi-criteria decision software requirements. He is also the first to propose decision-making software based on HFL-TOPSIS.

**KEYWORDS:** multi-criteria decision; *Fuzzy TOPSIS*; software engineering; requirements.

## 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,

<sup>1</sup> Bolsista do CNPq. Universidade Tecnológica Federal do Paraná, Curitiba, Paraná, Brasil. E-mail: vivianeruotolo@alunos.utfpr.edu.br . ID Lattes: 9746632288302682.

<sup>2</sup> Voluntária. Universidade Tecnológica Federal do Paraná, Curitiba, Paraná, Brasil. E-mail: tainaranovaes@alunos.utfpr.edu.br. ID Lattes: 5943061190326456.

<sup>3</sup> Voluntária. Universidade Tecnológica Federal do Paraná, Curitiba, Paraná, Brasil. E-mail: dan.plschuster@gmail.com. ID Lattes: 7354334111411484.

<sup>4</sup> Mestre em Modelagem Computacional de Sistemas. Universidade Federal de Tocantins, Araguaína Tocantins, Brasil. E-mail: lucyanocm@gmail.com. ID Lattes: 3526769585643531.

<sup>5</sup> Docente no Programa de Pós-Graduação em Administração. Universidade Tecnológica Federal do Paraná, Curitiba, Paraná, Brasil. E-mail: frjunior@utfpr.edu.br. ID Lattes: 3024249030533602.

2016). Uma vez que essas etapas são interdependentes, é importante dedicar esforços às etapas iniciais que antecedem à etapa de 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. Essa etapa é essencial para subsidiar a etapa de desenvolvimento, principalmente nas atividades relacionadas à modelagem do software por meio da construção de diagramas de Engenharia de Software (Pressman; Maxim, 2016).

Vale destacar que, durante as buscas realizadas em bases de periódicos e de patentes, não foram identificados estudos prévios que apresentem aplicações de métodos multicritério para priorização de requisitos de software de apoio à tomada de decisão multicritério. Também não foram encontrados softwares de apoio à decisão multicritério, com interface gráfica amigável, baseados em *Hesitant Fuzzy Linguistic Term Sets* (HFLTS). Esse tipo de software é bastante útil para auxiliar as empresas a encontrarem soluções racionais em problemas que envolvem a escolha ou a priorização de alternativas com base em múltiplos critérios, especialmente quando há incerteza e hesitação por parte dos decisores na definição dos valores de entrada do modelo (Beg Rashid, 2013).

Nesse contexto, o objetivo do presente estudo é desenvolver um novo software de apoio à tomada de decisão, com o apoio do *Fuzzy TOPSIS* para a priorização dos requisitos. Esse novo software emerge da necessidade de uma ferramenta de apoio à tomada de decisão multicritério em grupo, sob incerteza, para usuários não especialistas. O software desenvolvido é baseado no método *Hesitant Fuzzy Linguistic TOPSIS* (HFL-TOPSIS) (Beg Rashid, 2013) e foi nomeado *Continuum*. As seções a seguir apresentam os métodos adotados e alguns dos principais resultados obtidos.

## MATERIAIS E MÉTODOS

Esse estudo pode ser dividido em duas etapas principais: (1) definição e priorização de requisitos; e (2) modelagem e implementação computacional. A primeira etapa pode ser classificada como uma pesquisa quantitativa baseada em modelagem computacional (Novaes et al., 2022), por envolver a construção e aplicação de um modelo multicritério baseado em *Fuzzy TOPSIS* para priorização de requisitos. Ela se iniciou com o estudo do HFL-TOPSIS e um levantamento de softwares de decisão multicritério, com interface gráfica amigável, baseados em lógica *fuzzy* e/ou suas extensões. Foram consultadas as bases *Web of Science*, *Scopus*, *Springer*, *Taylor & Francis*, *IEEE-Xplore* e *Google Scholar*, e o Instituto Nacional de Propriedade Intelectual (INPI).

Em seguida, houve a definição dos requisitos funcionais e não funcionais do *software*. Para isso, mediante um *brainstorming* com três pesquisadores especialistas em HFLTS, fez-se o levantamento das funcionalidades requeridas e a priorização dos requisitos que deveriam estar contidos na primeira versão do software. Para isso, usando o *Microsoft (MS) Excel*, aplicou-se o *Fuzzy TOPSIS*, que foi escolhido por ser um dos métodos multicritério mais utilizados na literatura sobre priorização de requisitos de software e por permitir o uso de julgamentos linguísticos (Novaes et al., 2022).

Na segunda etapa, iniciou-se o processo de modelagem de *software*. A partir de um exemplo de uso do HFL-TOPSIS em *MS Excel*, foram identificadas as relações da aplicação, as quais permitiram a criação do diagrama de classes, usando a ferramenta



*Lucidchart*, para contribuir com a programação orientada à objetos do modelo de decisão em questão. Essas definições foram essenciais para identificar as ferramentas de desenvolvimento adequadas. Desse modo, foi imprescindível considerar o acesso dos usuários à aplicação, o conhecimento prévio das desenvolvedoras e a precisão numérica para os cálculos envolvidos na execução do método de tomada de decisão.

Com relação à precisão numérica, foram implementados dois protótipos com base no exemplo de uso do HFL-TOPSIS: (1) Cálculo direto do método na linguagem de programação *web* PHP; e (2) Implantação orientada à objetos do modelo de decisão e seus respectivos cálculos em *Python*. Por conseguinte, a comparação entre os dois protótipos serviu de embasamento para estabelecer o *Django*, um *framework* de programação *web* em *Python*, no *backend*. No *frontend*, utilizou-se do *TypeScript* e do *React*, uma biblioteca da linguagem *JavaScript* para páginas da *web*. Na implementação computacional, houve uma transição do protótipo em *Python* para a estrutura do *framework Django* a fim de uni-la com o *frontend* em *React*, o qual produziu o *design* idealizado no *Figma*. Na seção a seguir são apresentados os principais resultados.

## RESULTADOS E DISCUSSÕES

Para definição e priorização dos requisitos para um novo software de apoio à decisão multicritério, foi composta uma equipe de três decisores. 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. Esses decisores realizaram a definição dos requisitos, baseando-se nas necessidades do projeto e no levantamento de softwares prévios, resultando no Quadro 1.

**Quadro 1 - Requisitos definidos pelos decisores**

Características	Requisitos
Compleitude funcional	R <sub>1</sub> . Permitir aos decisores a atribuição de pesos aos critérios
	R <sub>2</sub> . Permitir o uso de critérios qualitativos para avaliação de alternativas
	R <sub>3</sub> . Permitir o uso de critérios quantitativos para avaliação de alternativas
	R <sub>4</sub> . Os critérios quantitativos devem ser avaliados utilizando valores numéricos
	R <sub>5</sub> . Os pesos dos critérios devem ser avaliados usando julgamentos linguísticos
Operacionalidade	R <sub>6</sub> . Os decisores poderão escolher uma escala com 5 ou 7 termos linguísticos
Acessibilidade	R <sub>7</sub> . O software precisa ser auto explicativo em relação ao seu uso
	R <sub>8</sub> . O software deve fornecer suporte ao cadastro de usuários
Confidencialidade	R <sub>9</sub> . O administrador deve ter acesso ao cadastro e alteração de usuários
Instalabilidade	R <sub>10</sub> . O software deve funcionar em plataforma WEB
Recuperabilidade	R <sub>11</sub> . Quando houver interrupção, o software deve salvar temporariamente os dados
Modificabilidade	R <sub>12</sub> . O código deve conter comentários que expliquem as etapas implementadas
Apreensibilidade	R <sub>13</sub> . Uma figura que forneça uma visão geral das etapas de uso
	R <sub>14</sub> . Uma figura que represente a relação das teorias matemáticas usadas.
	R <sub>15</sub> . Deve apresentar os principais elementos do problema na mesma página.
Autenticidade	R <sub>16</sub> . O decisor poderá salvar os dados de entrada e os resultados calculados.

Fonte: Elaborado pelos autores (2023).

Posteriormente, cada decisor estimou os pesos dos requisitos utilizando uma escala composta por cinco termos linguísticos, os quais são associados aos seguintes números *fuzzy* triangulares ( $l, m, u$ ): “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)” (Novaes et al., 2022). Já a escala para ponderar a opinião de cada decisor foi definida com base nos mesmos termos linguísticos, porém com números *fuzzy* que variam entre 0 e 1. Os julgamentos foram convertidos em números *fuzzy*, resultando nos dados de entrada apresentados na

Tabela 1. Após aplicar o *Fuzzy TOPSIS* conforme Novaes et al. (2022), as pontuações globais ( $CCi$ ) foram obtidas, as quais serviram de base para classificação dos requisitos.

**Tabela 1 – Dados de entrada e dados de saída do *Fuzzy TOPSIS* para priorização de requisitos**

Item avaliado	Dados de entrada do modelo <i>Fuzzy TOPSIS</i>									Dados de saída	
	$D_1$			$D_2$			$D_3$			$CCi$	Classificação
	l	m	u	l	m	u	l	m	u		
R <sub>1</sub>	7,5	10,0	10,0	5,0	7,5	10,0	7,5	10,0	10,0	0,673	2°
R <sub>2</sub>	7,5	10,0	10,0	7,5	10,0	10,0	7,5	10,0	10,0	0,703	1°
R <sub>3</sub>	7,5	10,0	10,0	7,5	10,0	10,0	7,5	10,0	10,0	0,703	1°
R <sub>4</sub>	5,0	7,5	10,0	0,0	2,5	5,0	2,5	5,0	7,5	0,460	11°
R <sub>5</sub>	5,0	7,5	10,0	5,0	7,5	10,0	5,0	7,5	10,0	0,608	6°
R <sub>6</sub>	5,0	7,5	10,0	5,0	7,5	10,0	5,0	7,5	10,0	0,608	6°
R <sub>7</sub>	5,0	7,5	10,0	7,5	10,0	10,0	7,5	10,0	10,0	0,665	3°
R <sub>8</sub>	5,0	7,5	10,0	2,5	5,0	7,5	2,5	5,0	7,5	0,515	9°
R <sub>9</sub>	7,5	10,0	10,0	5,0	7,5	10,0	5,0	7,5	10,0	0,645	4°
R <sub>10</sub>	7,5	10,0	10,0	7,5	10,0	10,0	0,0	2,5	5,0	0,580	7°
R <sub>11</sub>	5,0	7,5	10,0	5,0	7,5	10,0	0,0	2,5	5,0	0,514	10°
R <sub>12</sub>	5,0	7,5	10,0	7,5	10,0	10,0	5,0	7,5	10,0	0,636	5°
R <sub>13</sub>	5,0	7,5	10,0	7,5	10,0	10,0	5,0	7,5	10,0	0,636	5°
R <sub>14</sub>	5,0	7,5	10,0	2,5	5,0	7,5	2,5	5,0	7,5	0,515	9°
R <sub>15</sub>	2,5	5,0	7,5	7,5	10,0	10,0	5,0	7,5	10,0	0,580	7°
R <sub>16</sub>	2,5	5,0	7,5	5,0	7,5	10,0	5,0	7,5	10,0	0,551	8°
Peso dec.	0,75	1,0	1,0	0,5	0,75	1,0	0,5	0,75	1,0		

Fonte: Elaborado pelos autores (2023).

Os requisitos considerados como prioritários estão relacionados a funcionalidades essenciais do software, como permitir o uso de critérios qualitativos ( $R_2$ ) e qualitativos ( $R_3$ ), e possibilitar a atribuição de pesos aos critérios ( $R_1$ ).  $R_7$  também se destacou, demonstrando a importância de o software ser auto explicativo. Os requisitos com menor prioridade foram  $R_4$  e  $R_{11}$ , que se referem à tolerância a falhas e uso de valores numéricos. Os resultados foram apresentados aos decisores, que concordaram com a ordem de prioridade. Em acordo com a equipe desenvolvedora, foram incluídos na primeira versão do software os requisitos entre a 1ª e a 8ª posição.

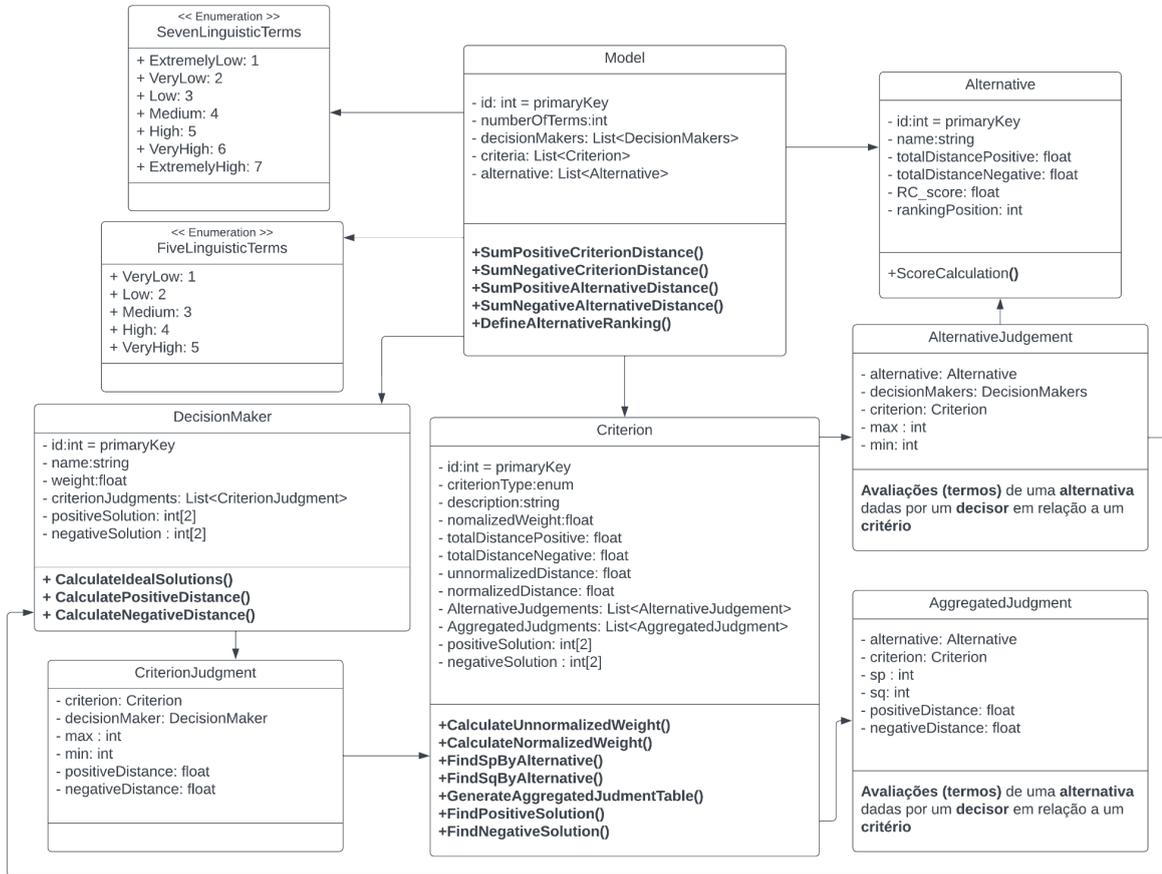
Sobre a etapa de modelagem do software, a Figura 1 apresenta o diagrama de classes desenvolvido para a criação de um modelo decisório que utiliza o HFL-TOPSIS. A classe *Model* representa o modelo com os seus decisores, critérios e alternativas, os quais compõem, respectivamente, as classes: *DecisionMaker*, *Criterion* e *Alternative*. Esses objetos são responsáveis por gerar o modelo, o qual se baseia em cálculos com os julgamentos dos decisores (*CriterionJudgment* e *AlternativeJudgment*). *SevenLinguisticTerms* e *FiveLinguisticTerms* são as escalas de julgamento disponíveis.

Posteriormente, foram implementados dois protótipos para execução dos cálculos do HFL-TOPSIS, o que permitiu identificar a diferença de execução entre eles. Como é evidenciado na Figura 2, é notável que em *Python* obteve-se uma maior precisão numérica, apesar dos resultados serem bem semelhantes e com uma diferença de duas ou três casas decimais entre os resultados. Por esse motivo, optou-se pelo *Python* e o *framework Django* para a realização matematicamente mais precisa do HFL-TOPSIS.

A primeira versão do *software* inclui as funcionalidades essenciais do projeto. Muitas funções são executadas com o *React*. O *backend* é requisitado, apenas, na criação ou alteração do modelo, em que há a necessidade de refazer os cálculos. A etapa de criação possui um total de quatro passos para construção do modelo, tais quais: (1) descrição do modelo, (2) definição do problema, (3) inserção dos julgamentos dos

decisores, e (4) ranqueamento dos resultados. Ao finalizar o terceiro passo, os dados inseridos são formatados no modelo *JavaScript Object Notation* (JSON) e enviados para a *Application Programming Interface* (API) em *Django*.

Figura 1 - Diagrama de classes desenvolvido para o software



Fonte: Elaborada pelos autores (2023).

Figura 2 - Comparação entre os protótipos em *Python* e em *PHP* para o cálculo do peso dos critérios

CRITERION WEIGHT (PHP):		CRITERION WEIGHT (PYTHON):	
CRITERION	NORMALIZED	CRITERION	NORMALIZED
Custo	0.48031486431208	Custo	0.4803148643120842
Qualidade	0.38884018940448	Qualidade	0.3888401894044807
Ocorrencia	0.13084494628344	Ocorrencia	0.13084494628343515

Fonte: Elaborado pelos autores (2023).

A Figura 3 exemplifica um desses arquivos JSON. Nesse exemplo, o modelo é composto por um decisor, um critério, uma alternativa e cinco termos linguísticos a fim de demonstrar a estrutura de armazenamento dos dados. Nesse sentido, depois do *backend* executar a sua função, é retornado ao *frontend* um JSON com a mesma estrutura. Contudo, são adicionados os dados finais que permitem o ranqueamento no último passo.

Figura 3 - Exemplo de JSON utilizado no Continuum

```
1  {
2  "id": "1dd6d183-fdf7-42e2-b645-082bc9df23ae",
3  "info": {
4    "description": "Modelinho basico",
5    "keywords": ["fornecedores"],
6    "name": "Modelinho",
7    "searchString": "modelinho modelinho basico fornecedores"
8  },
9  "decisionMakers": {
10   "id": "d755148e-644c-49b6-a668-1b20ee911554",
11   "name": "Decision1",
12   "weight": 0.4,
13   "criterion_judgments": {
14     "criterion_id": "0a211b60-ee0c-46df-af7f-0cf06e32a165",
15     "sp": 3,
16     "sq": 4
17   },
18   "positive_ideal_solution": [0, 0],
19   "negative_ideal_solution": [0, 0]
20 },
21 "criteria": {
22   "id": "ddea19be-89a7-4186-b098-f0a94107f86b",
23   "description": "Qualidade",
24   "criterion_type": "benefit",
25   "alternative_judgments": {
26     "alternative_id": "a832ed90-b798-49f3-a063-94135fdae74a",
27     "decision_maker_id": "d755148e-644c-49b6-a668-1b20ee911554",
28     "min_value": 4,
29     "max_value": 4
30   }
31 },
32 "alternatives": {
33   "id": "a832ed90-b798-49f3-a063-94135fdae74a",
34   "name": "Fornecedor A",
35   "ranking_position": 0,
36   "score": 0,
37   "total_distance_negative": 0,
38   "total_distance_positive": 0
39 },
40 "linguisticScale": 5
41 }
```

Fonte: Elaborada pelos autores (2023).

## CONCLUSÃO

Este trabalho apresentou uma síntese do processo de implementação de um novo *software* de auxílio à tomada de decisões sob incerteza. Com base no levantamento bibliográfico, pode-se afirmar que o presente estudo é o primeiro a aplicar uma técnica multicritério para priorização de requisitos de software de decisão multicritério. Ademais, o software implementado possui alto potencial de uso em diversas áreas organizacionais, como gestão da qualidade, suprimentos, logística, sustentabilidade e projetos. Por ser baseado no HFL-TOPSIS, permite a participação de um grupo de decisores e possibilita a eles o uso de termos linguísticos (ex. “baixo” e “alto”) e expressões linguísticas (ex. “entre alto e muito alto”), o que é especialmente útil para ambientes de incerteza.

Vale destacar, que, neste estudo, a validação do software em diferentes cenários foi limitada. Quando a aplicação possuir usuários ativos e usos frequentes, será possível identificar os impactos do sistema e demais necessidades em maior proporção. Outrossim, o trabalho pode ser ampliado para a execução de outros métodos multicritério em adição ao HFL-TOPSIS, como também o acréscimo de novas funcionalidades.

## Agradecimentos

Ao CNPQ (Código 409529/2021-4) pelo financiamento deste estudo.

## Disponibilidade de código

O código encontra-se em um repositório privado, pois está sendo aperfeiçoado.

## Conflito de interesse

Não há conflito de interesse.

## REFERÊNCIAS

- BEG, I.; RASHID, T. TOPSIS for Hesitant Fuzzy Linguistic Term Sets. **International Journal of Intelligent Systems**, v. 28, p.1162-1171, 2013.
- NOVAES, T.S.; RUOTOLO, V.; SCHUSTER, D.P.; MARTINS, L.C.; LIMA, F.R. **Aplicação do método Fuzzy TOPSIS para priorização dos requisitos de um software de decisão multicritério**. In: XXV Seminários em Administração (SEMEAD), São Paulo, 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, p.365–378.
- PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software**: uma abordagem profissional. 8 ed. Porto Alegre: AMGH: Porto Alegre, 2016.