

UNIVERSIDADE CANDIDO MENDES – UCAM
PROGRAMA DE PÓS-GRADUAÇÃO EM PESQUISA OPERACIONAL E
INTELIGÊNCIA COMPUTACIONAL
CURSO DE MESTRADO EM PESQUISA OPERACIONAL E INTELIGÊNCIA
COMPUTACIONAL

Orpheu de Souza Ayres

**ENSINO DE LINGUAGENS DE PROGRAMAÇÃO E ALGORITMOS PARA
ALUNOS DE CURSOS TÉCNICOS DE INFORMÁTICA: UMA ABORDAGEM
MULTICRITERIAL**

CAMPOS DOS GOYTACAZES

Julho de 2020

UNIVERSIDADE CANDIDO MENDES – UCAM
PROGRAMA DE PÓS-GRADUAÇÃO EM PESQUISA OPERACIONAL E
INTELIGÊNCIA COMPUTACIONAL
CURSO DE MESTRADO EM PESQUISA OPERACIONAL E INTELIGÊNCIA
COMPUTACIONAL

Orpheu de Souza Ayres

**ENSINO DE LINGUAGENS DE PROGRAMAÇÃO E ALGORITMOS PARA
ALUNOS DE CURSOS TÉCNICOS DE INFORMÁTICA: UMA ABORDAGEM
MULTICRITERIAL**

Dissertação apresentada ao Programa de Pós-Graduação em Pesquisa Operacional e Inteligência Computacional, da Universidade Cândido Mendes – Campos/RJ, para obtenção do grau de MESTRE EM PESQUISA OPERACIONAL E INTELIGÊNCIA COMPUTACIONAL.

Orientador: Professor Milton Erthal Jr.

CAMPOS DOS GOYTACAZES

Julho de 2020

Catlogação na Fonte

Preparada pela Biblioteca da **UCAM – CAMPOS** 015/2021

Ayres, Orpheu de Souza.

Ensino de linguagens de programação e algoritmos para os alunos de cursos técnico de informática. / Orpheu de Souza Ayres. – 2020.
136 f.

Orientador(a): Milton Erthal Júnior.

Dissertação de Mestrado em Pesquisa Operacional e Inteligência Computacional – Universidade Candido Mendes – Campos. Campos dos Goytacazes, RJ, 2020.

Referências: f. 110-115.

1. Algoritmos. 2. Multicritério. I. Erthal Júnior, Milton, orient. II. Universidade Candido Mendes – Campos. III. Título.

CDU – 004.021:65.012

Bibliotecária Responsável: Flávia Mastrogirolamo CRB 7ª-6723

ORPHEU DE SOUZA AYRES

**ENSINO DE LINGUAGENS DE PROGRAMAÇÃO E ALGORITMOS PARA
ALUNOS DE CURSOS TÉCNICOS DE INFORMÁTICA: UMA ABORDAGEM
MULTICRITERIAL**

Dissertação apresentada ao Programa de Pós-Graduação em Pesquisa Operacional e Inteligência Computacional, da Universidade Cândido Mendes – Campos/RJ, para obtenção do grau de MESTRE EM PESQUISA OPERACIONAL E INTELIGÊNCIA COMPUTACIONAL.

Aprovado em 22 de julho de 2020.

BANCA EXAMINADORA

Prof. Milton Erthal Júnior, D.Sc – Orientador.
Universidade Candido Mendes

Prof. Ítalo de Oliveira Matias, D.Sc – Coorientador.
Universidade Candido Mendes

Prof. Frederico Galaxe Paes, D.Sc.
IFF-Guarus

CAMPOS DOS GOYTACAZES, RJ

2020

Dedicatória

Dedico esta dissertação à minha amada esposa Andressa Amorim Zampirolli, pelo amor, companheirismo, carinho e apoio incondicional.

Aos meus filhos, meus tesouros, presentes de Deus, que junto com minha esposa completam uma de minhas principais missões nessa vida, nossa família.

AGRADECIMENTOS

A Deus, fonte de tudo, D'Ele provém toda a sabedoria e conhecimento; à minha família, que tanto me apoia, ao meu amigo Alfeu Garcia Júnior (Satu) por todo carinho e, é claro, pela revisão do trabalho, ao professor Milton Erthal Junior pela sua amizade, presteza e dedicação, ao professor Ítalo de Oliveira Matias pela valorosa colaboração e contribuição, aos excelentes professores do mestrado, aos colegas de mestrado pela contribuição mútua, em especial aos três amigos sempre presentes e grandes parceiros de pesquisa: Denilton Macário de Paula, André Bessa e Aline Estaneck, aos colegas de trabalho do IFF, aos irmãos da Igreja Batista, que estão sempre em oração por nós, ao IFF, que me deu apoio na realização deste, aos amigos e familiares pelo incentivo e apoio, à Universidade Cândido Mendes pelo trabalho focado na preparação de novos pesquisadores.

Para o trabalho que gostamos,
levantamo-nos cedo e fazemo-lo
com alegria.

William Shakespeare

O prazer no trabalho aperfeiçoa
a obra.

Aristóteles

Nenhum trabalho de qualidade
pode ser feito sem concentração e
auto sacrifício, esforço e dúvida.

Max Beerbohm

RESUMO

ENSINO DE LINGUAGENS DE PROGRAMAÇÃO E ALGORITMOS PARA ALUNOS DE CURSOS TÉCNICOS DE INFORMÁTICA: UMA ABORDAGEM MULTICRITERIAL

Esta pesquisa tem por objetivo selecionar e ranquear linguagem(ns) de programação ou tecnologias de ensino de algoritmo para alunos iniciantes em programação. A partir de levantamento de dados junto ao corpo docente dos Institutos Federais por meio de formulários online e reuniões, aplicou-se método de tomada de decisão multicritério AHP FUZZY TOPSIS. Utilizou-se do método de tomada de decisão multicritério AHP sobre a avaliação individual de cada professor, com média aritmética sobre os resultados na definição do vetor de pesos consistentes. Estes pesos foram utilizados no método FUZZY TOPSIS para a geração do ranking de linguagens de acordo com os critérios criados para esta finalidade. Critérios específicos foram eleitos enxergando o primeiro ano de programação separadamente; O ranqueamento obtido sobre as linguagens de programação têm destaque para, na respectiva ordem: Portugol Webstudio, Scratch e studio.code.org. Os resultados obtidos deverão contribuir para mudanças no planejamento curricular das turmas que iniciam os estudos de programação no curso técnico de informática do IFFluminense.

Palavras-chave: Lógica Fuzzy. TOPSIS. AHP FUZZY TOPSIS. Auxílio Multicritério a Decisão. Aprendizagem na Educação sobre Algoritmos de Computadores.

ABSTRACT

TEACHING PROGRAMMING LANGUAGES AND ALGORITHMS TO STUDENTS OF TECHNICAL COURSES IN INFORMATICS: A MULTICRITERIAL APPROACH

This research aims to select and rank programming language (s) or algorithm teaching technologies for students who are new to programming. Based on data collection from the faculty of the Federal Institutes through online forms and meetings, the AHP FUZZY TOPSIS multi-criteria decision-making method was applied. We used the AHP multicriteria decision-making method on the individual assessment of each teacher, with an arithmetic mean on the results in defining the vector of consistent weights. These weights were used in the FUZZY TOPSIS method to generate the ranking of languages according to the criteria created for this purpose. Specific criteria were chosen considering the first year of programming separately; The ranking obtained on the programming languages is highlighted, in the respective order: Portugol Webstudio, Scratch and studio.code.org. The results obtained should contribute to changes in the curricular planning of the classes that begin programming studies in the IFFluminense IT course.

Keywords: Fuzzy Logic. TOPSIS, AHP FUZZY TOPSIS. Multicriteria Decision Aid. Learning in Education on Computer Algorithms.

LISTA DE FIGURAS

Figura 1 Funções de pertinência para a variável altura.....	26
Figura 2 Relação de pertinência.....	27
Figura 3 Sistema Lógico Fuzzy	29
Figura 4 Valores para fuzzificação	31
Figura 5 Estrutura do método AHP.....	36
Figura 6 Exemplo do método AHP	37
Figura 7 Aplicação do método AHP.....	40
Figura 8 Distância de 3 pontos no espaço	41
Figura 9 Distância euclidiana de 2 pontos.....	41
Figura 10 Matriz de exemplos de valores de diferentes grandezas	43
Figura 11 Matriz (D) de exemplos de valores numéricos	43
Figura 12 Matriz (K) normalizada sobre o exemplo.....	44
Figura 13 Matriz de exemplo de Pesos	44
Figura 14 Matriz pesificada sobre o exemplo	45
Figura 15 Matrizes PIS e NIS com valores resultantes sobre o exemplo.....	46
Figura 16 Matriz de cálculo das distâncias com ranking final.....	47
Figura 17 Aplicação do método TOPSIS.....	47
Figura 18 Número Fuzzy Trapezoidal	50
Figura 19 Número Fuzzy Triangular	50
Figura 20 Número Fuzzy Convexo	51
Figura 21 Matriz de respostas de decisores.....	53
Figura 22 Matrizes fuzzificadas	54
Figura 23 Sugestão de valores equiparados aos usados no método AHP.....	55
Figura 24 Valores Fuzzy sugeridos com intervalo de 2 abaixo e acima.....	55
Figura 25 Matriz combinada	56
Figura 26 Matriz com valores normalizados	56
Figura 27 Matriz de pesos	57
Figura 28 Matriz com valores pesificados	57
Figura 29 : Matrizes PIS e NIS	57
Figura 30 Matriz resultante sobre a Matriz PIS	58
Figura 31 Matriz resultante sobre a Matriz NIS	58
Figura 32 Matriz com cálculo de coeficiente de proximidade	59

Figura 33 Matriz final com valores ordenados.....	59
Figura 34 Estrutura do método FUZZY TOPSIS	60
Figura 35 Matriz de respostas de decisores.....	62
Figura 36 Matrizes fuzzificadas	63
Figura 37 : Sugestão de valores equiparados aos usados no método AHP.....	64
Figura 38 Valores Fuzzy sugeridos com intervalo de 2 abaixo e acima.....	64
Figura 39 Matriz combinada	65
Figura 40 Matriz com valores normalizados	66
Figura 41 Matriz de pesos	66
Figura 42 Matriz com valores pesificados	67
Figura 43 Matrizes PIS e NIS	67
Figura 44 Matriz resultante sobre a Matriz PIS	68
Figura 45 Matriz resultante sobre a Matriz NIS	68
Figura 46 Matriz com cálculo de coeficiente de proximidade	69
Figura 47 Matriz final com valores ordenados.....	69
Figura 48 Matriz com valores crisp.....	72
Figura 49 Matriz com valores fuzzificados.....	72
Figura 50 Matriz com valores normalizados	73
Figura 51 Matriz de pesos	73
Figura 52 Matriz pesificada	74
Figura 53 Matrizes com soluções ideais positivas (PIS) e negativas (NIS)	74
Figura 54 Matriz com distâncias das alternativas para a solução ideal positiva..	74
Figura 55 Matriz com distâncias das alternativas para a solução ideal negativa	75
Figura 56 Matriz com o coeficiente de aproximação, ou distâncias globais	75
Figura 57 Método FUZZY TOPSIS com valores crisp.....	76
Figura 58 Matrizes de julgamento no método AHP por 3 decisores.....	77
Figura 59 Pesos consistentes resultantes	77
Figura 60 Matriz de julgamentos de decisores	78
Figura 61 Matriz fuzzificada de julgamentos de decisores	78
Figura 62 Matriz reduzida de decisores para critérios por alternativas	79
Figura 63 Matriz normalizada considerando o tipo de critério (Custo x Benefício)	79
.....	79
Figura 64 Matriz de pesos oriundos do método AHP	79
Figura 65 Matriz pesificada (Matriz normal x pesos)	80

Figura 66 Matrizes com soluções ideais positivas (PIS) e negativas (NIS).....	80
Figura 67 Matriz com distâncias dos pontos das alternativas em relação a PIS .	80
Figura 68 Matriz com distâncias dos pontos das alternativas em relação a NIS.	81
Figura 69 Matriz dos coeficientes de proximidades das alternativas.....	81
Figura 70 Julgamentos entre critérios dos decisores 1, 2 e 3	101
Figura 71 Geração de pesos para AHP FUZZY TOPSIS	101
Figura 72 Matriz reduzida de avaliações em critérios por alternativas	102
Figura 73 Matriz Normalizada	103
Figura 74 Matriz de pesos consistentes (AHP)	103
Figura 75 Matriz pesificada	103
Figura 76 Matriz com soluções ótimas positivas (PIS) e negativas (NIS).....	104
Figura 77 Matriz com as distâncias dos pontos para PIS.....	104
Figura 78 Matriz com as distâncias dos pontos para NIS.....	105
Figura 79 Coeficiente de proximidade e Ordenação de resultados.....	105

LISTA DE TABELAS

Tabela 1 Definição e explicação de valores associados sugeridos.....	30
Tabela 2 Valores fuzzy de sugestão.....	30
Tabela 3 Matriz com valores linguísticos.....	52
Tabela 4 Matriz com valores linguísticos.....	61
Tabela 5 Tabela de valores de conversão de expressões linguísticas para valores crisp.....	78
Tabela 6 Tabela com valores para fuzzificação	97
Tabela 7 Valores para fuzzificação	102

LISTA DE QUADROS

Quadro 1 Representação de critérios, alternativas e decisores	32
Quadro 2 Normalização sobre vetores consistentes	39
Quadro 3 Comparação de métodos FUZZY TOPSIS.....	49
Quadro 4 Quadro comparativo dos métodos TOPSIS e FUZZY TOPSIS	70
Quadro 5 Quadro com Critérios significativos	90
Quadro 6 Correspondência das representações de critérios e alternativas	94
Quadro 7 Matriz de geração de pesos	95
Quadro 8 Valores de conversão de literal para crisp.....	97

SUMÁRIO

1	INTRODUÇÃO	15
1.1	PROBLEMAS DA PESQUISA	16
1.2	RELEVÂNCIA DO TEMA	17
1.3	QUESTÕES DA PESQUISA	18
1.4	OBJETIVOS	19
1.4.1.	OBJETIVO GERAL	19
1.4.2.	OBJETIVOS ESPECÍFICOS	19
1.5	JUSTIFICATIVA	20
1.6	ESTRUTURA DO TRABALHO	21
2	REFERENCIAL TEÓRICO	22
2.1	LINGUAGENS DE PROGRAMAÇÃO E ALGORITMOS	22
2.2	LINGUAGENS – DESENVOLVIMENTO DE SOFTWARES	23
2.3	LÓGICA FUZZY	25
2.4	AUXÍLIO MULTICRITÉRIO À DECISÃO	31
2.5	CRITÉRIOS	32
2.6	MÉTODO AHP	34
2.6.1.	Descrição do Método AHP	36
2.6.1.1.	Etapa 1: Estrutura do problema em hierarquia	36
2.6.1.2.	Etapa 2: Definir prioridades entre os elementos da hierarquia	37
2.6.1.3.	Etapa 3: Avaliar a Consistência das prioridades relativas encontradas	38
2.6.1.4.	Etapa 4: Calcular a prioridade composta	39
2.7	TOPSIS	40
2.8	MÉTODO FUZZY TOPSIS	48
2.9	MÉTODO FUZZY TOPSIS COM VALORES CRISP	71
2.10	AHP FUZZY TOPSIS com Múltiplos Decisores	76
3	METODOLOGIA	82
3.1	Síntese das etapas de pesquisa	82
3.2	Desenvolvimento dos Formulários	82
3.3	ALTERNATIVAS	83
3.4	CRITÉRIOS	87
3.5	Método AHP para Determinar os Pesos dos Critérios	93

3.6	Método AHP FUZZY TOPSIS com Múltiplos Decisores	95
3.7	Institutos Federais e Universidades (fim da metodologia – estudo de caso)	99
4	Resultados	100
4.1	Estrutura Resultante	100
4.2	Dos Critérios e Pesos	100
4.3	Discussão	105
5	Conclusões	108
	REFERÊNCIAS	110
	Apêndice 1 - Formulário de Levantamento de Sugestões	116
	Apêndice 2 Formulário sobre levantamento de alternativas e de critérios.....	117
	Apêndice 3 Formulário planilha de comparação de importância entre critérios	117
	Apêndice 4 Formulário de Julgamentos paritários – Critérios vs Alternativas	120
	Apêndice 5 Plano de aula - 1º Bimestre	130
	Apêndice 6 Avaliações sobre o critério Legibilidade	132
	Anexo 1 – Fórmulas de normalização.....	135

1 INTRODUÇÃO

O Pensamento Computacional é amplamente utilizado por profissionais de diversas áreas, principalmente pelos desenvolvedores de softwares, engenheiros, cientistas, administradores, entre/ outros. Esses profissionais usam o Pensamento Computacional para a solução de problemas práticos da área de informática, sendo assim, o Pensamento Computacional não se restringe à programação, mas sim às tarefas cognitivas (PROCTOR; BLIKSTEIN, 2008). Desta forma, pode ser comparado a um algoritmo que é uma sequência detalhada de ações, para quando executado, realizar a solução de um problema ou realizar uma tarefa. Os algoritmos, que constituem a expressão do Pensamento Computacional podem ser trabalhados independentemente de linguagens de programação, uma vez que são a reprodução da lógica computacional na realização de tarefas. Estes necessitam de regras sintáticas e de estrutura formal com um universo limitado de comandos e procedimentos pré-definidos (MEDINA; FERTING, 2006).

Devido os algoritmos serem utilizados no desenvolvimento de aplicativos, capacidade de formulá-los constitui parte intrínseca e básica na formação de um desenvolvedor de programas (SOARES, 2004). A linguagem de programação, que emprega algoritmos, consiste em um método padronizado para gerar instruções a serem executadas pelo computador. Um bom programador precisa dominar e usar uma linguagem de programação e isto não é possível sem o conhecimento de algoritmos. O aprendizado de ambos deve ser simultâneo (RAPKIEWICZ; FALKEMBACH, 2006).

Para potencializar e universalizar a solução de problemas de informática, os conceitos do Pensamento Computacional devem ser ensinados desde a infância a fim de ampliar o entendimento do contexto tecnológico atual e para a construção do conhecimento (ZANETTI *et al.*, 2016).

Um curso introdutório de programação aprendido com sucesso constitui a base para o estudo eficaz dos cursos subsequentes e o domínio desta especialidade. Por outro lado, a excessiva complexidade de aprender um curso introdutório pode levar a uma queda na motivação e a baixos resultados de aprendizado em geral. A escolha de linguagem para este fim tem causado intensos debates nas últimas década (PROKOP *et al.*, 2019).

Periódicos relacionados ao computador, como o Boletim, Special Interest Group on Computer Science Education (SIGCSE) e diversos outros revelam numerosas e contínuas discussões sobre o curso de ciência da computação e sobre linguagens de programação, apropriadas para a introdução da disciplina de programação em faculdades (SMOLARSKI, 2003).

A tomada de decisão pode levar em consideração um único critério, e ser relativamente fácil, entretanto, normalmente levam em consideração diversos aspectos sobre as alternativas e, comprovadamente, as tomadas de decisões consistentes são difíceis de serem feitas pelas pessoas. Para a solução destes problemas, têm sido desenvolvidas uma série de métodos de decisão baseados em critérios múltiplos, auxiliando as melhores escolhas de forma consistente, levando em consideração os julgamentos subjetivos e os pesos de cada critério (DOLAN, 2008).

A tomada de decisão envolve ações que podem ser complexas. O Auxílio Multicritério à Decisão constituído por etapa de análise de alternativas à luz de vários critérios, possui métodos classificados como métodos multicriteriais (COSTA, 2006). O problema exposto neste trabalho, apresenta alternativas (linguagem de programação) que serão valoradas proporcionalmente aos pesos dos critérios definidos pelo método multicriterial AHP e feito o ranqueamento para as melhores escolhas utilizando-se o método FUZZY TOPSIS com múltiplos decisores sobre múltiplas alternativas em múltiplos critérios.

O objetivo deste trabalho constitui tanto, a definição de critérios específicos para a escolha de linguagens de programação, quanto o ranqueamento destas linguagens de programação, e como consequência buscar potencializar o ensino de algoritmos e programação estruturada, bem como a redução de evasão e retenção dos alunos.

1.1 PROBLEMAS DA PESQUISA

Como fundamento, a escolha de uma linguagem de programação, objeto do presente trabalho, deve-se levar em consideração a ausência de conhecimento de programação por parte dos alunos, bem como a necessidade de construir uma base de entendimento sobre programação e tecnologias que possam ser continuadas e ampliadas.

Outro fundamento importante, neste contexto sobre os Institutos Federais, reside na visão de separação do primeiro ano de introdução de algoritmos e pensamento computacional, dos anos posteriores, onde serão aprofundados conhecimentos sobre linguagens voltadas à produtividade, contemplando programação orientada a objetos, bancos de dados e redes. O universo de programação a ser trabalhado no 1º ano deste curso, deve ficar restrito a parte de programação estruturada.

A linguagem de programação impõe limites ao desenvolvimento, restrições em tipos de estruturas de controle, em estruturas de dados, nas abstrações possíveis, nas plataformas compatíveis. O conhecimento amplo dos recursos das linguagens pode reduzir as limitações, e potencializar inovações de construções de códigos (SEBESTA, 2018). Qual linguagem ou tecnologia é melhor para os anos iniciais? Esta linguagem será propícia ao desenvolvimento dos anos finais da formação do aluno? Linguagens de programação na língua nativa(materna) e/ou baseada em programação de blocos reduzem o tempo de aprendizagem? São mais fáceis de aprender? Resultam em maior clareza sobre os conceitos básicos?

A hipótese é que a boa escolha da linguagem de programação reduzirá o esforço de aprendizagem com algoritmos, fomentará o ensino, dinamizará o conteúdo e resultará em uma redução da evasão e retenção dos alunos. Quais linguagens visuais, a partir de blocos, ou na língua pátria, tendem a facilitar o processo de ensino aprendizagem e gerar um sentimento de conforto junto à linguagem utilizada.

1.2 RELEVÂNCIA DO TEMA

A busca de melhores resultados no ensino de programação de computadores constitui objeto de pesquisa de muitas instituições de ensino, algumas colocando o foco em fatores como: método, processo cognitivo, habilidades preexistentes (JENKINS, 2002).

Há muitas pesquisas sobre as vantagens e desvantagens na utilização de linguagens de programação específicas nos cursos introdutórios, com ênfase na atenção sobre publicações voltadas à análise de critérios voltados para a escolha destas linguagens introdutórias (PROKOP *et al.*, 2019).

Uma pesquisa sobre “Algorithm Learning in Computer Algorithm Education” retorna mais de dois milhões de resultados, bem como sobre “teaching algorithms”.

Notoriamente há uma parcela significativa de artigos sobre ensino de algoritmos que se traduz em uma grande área de pesquisa, entretanto não foram encontrados artigos na literatura acadêmica para buscar critérios específicos na busca de linguagens de programação que separassem o primeiro ano no ensino de algoritmos e programação, nem que utilizassem o método FUZZY TOPSIS ou, AHP FUZZY TOPSIS o ranqueamento destas linguagens.

Alunos expostos a experiência de aprendizagem direta de algoritmo a partir de uma linguagem e da aprendizagem de apenas um algoritmo apresentam, em ambos os casos, insuficiência de habilidade para projetar algoritmos. A opinião dos alunos sugere haver menos dificuldade no planejamento e solução de algoritmos diretamente, do que no uso de linguagens de programação (KORKMAZ, 2013).

A desmotivação, reprovação ou evasão no ensino de algoritmos e programação resultam da dificuldade dos alunos em assimilar as abstrações deste processo, em desenvolver raciocínio lógico, além do despreparo do aluno e a crença de ser este conhecimento extremamente difícil de ser transposto. Há a necessidade de modificar modo tradicional instrucionista para gerar motivação e interesse pela disciplina (RAPKIEWICZ *et al.*, 2006).

1.3 QUESTÕES DA PESQUISA

- Como potencializar o aprendizado de algoritmo, reduzir o tempo de aprendizagem, reduzir o tempo de correção de erros, facilitar a escrita e criação de códigos de programação, melhorar a solução, reduzir as dificuldades, reduzir a evasão e a retenção a partir de linguagens e/ ou recursos de ensino de algoritmo e linguagens de programação?
- Quais alternativas (linguagens de programação ou recursos didáticos) são comuns às experiências e conhecimentos dos docentes do Instituto?
- Quais linguagens de programação devem ser consideradas e quais devem ser descartadas?
- Alguma(s) alternativa(s) deve(m) ser elencada(s) devido a sua relevância, mesmo que não comum aos docentes?
- Quais critérios serão efetivos para uma escolha adequada às demandas introdutórias da formação de programadores?

- Quais critérios atendem as percepções dos docentes da disciplina no que diz respeito a priorizar a aprendizagem e não necessariamente a produtividade de mercado, neste momento da formação?
- Qual o peso mais adequado de cada critério?
- Qual o método mais adequado para avaliar julgamentos e ranquear as alternativas, de acordo com vários decisórios com perspectivas, experiências e conhecimentos diferentes?

1.4 OBJETIVOS

1.4.1. OBJETIVO GERAL

O objetivo desta pesquisa, realizada no Instituto Federal Fluminense no estado do Rio de Janeiro, com foco nos cursos técnicos em informática, busca identificar a(s) linguagem(ns) de programação mais adequada(s) para facilitar o processo de ensino aprendizagem sobre algoritmos, de tal forma que repercuta em melhores resultados, tanto para o docente, quanto para o discente, bem como possa resultar na redução da evasão e da retenção.

1.4.2. OBJETIVOS ESPECÍFICOS

- Criar uma fonte de discussão sobre a diferenciação dos critérios de seleção das alternativas exclusivas para o primeiro ano letivo dos cursos técnicos de informática nos Institutos Federais e Universidades;
- Consolidar os critérios e seus respectivos pesos através da metodologia AHP capaz de verificar gerar critérios consistentes e respectivos pesos;
- Utilizar um método de análise multicriterial capaz de eleger e ranquear linguagens de programação, sobre um conjunto de avaliações de docentes com possíveis características de grande dispersão das avaliações;
- Possibilitar a redução na evasão e retenção da disciplina de programação I, no que diz respeito à aprendizagem sobre algoritmos e conhecimentos introdutórios de programação de aplicativos;
- Facilitar o processo de ensino tanto para o discente quanto para o docente;

- Reduzir a evasão e retenção no curso de formação de tecnólogo em informática ofertado pelo Instituto Federal Fluminense.

1.5 JUSTIFICATIVA

Utilizar uma ferramenta ou linguagem de programação adequada, que facilite a utilização, aprendizagem, aplicação de algoritmos, identificação de erros (tanto lógicos, quanto sintáticos), maior legibilidade, melhor visualização do código, bem como maior facilidade de escrita, nesta fase inicial de aprendizagem, pode resultar em uma melhora na redução do tempo de instrução e aprendizagem, na aquisição de conhecimentos relativos a algoritmos e principalmente, possibilitar a redução de reprovações e evasões nas turmas de iniciantes.

O aprimoramento da qualidade do processo de ensino e aprendizagem na área de programação é essencial, pois, há alto índice de reprovação e evasão nas disciplinas de algoritmo e programação (DELGADO *et al.*, 2005). A aprendizagem de programação requer múltiplas habilidades e resultam em altos índices de insucesso no ensino de habilidades básicas de programação (JENKINS, 2002).

Muitas são as situações a serem observadas para realizar o ensino de programação de computadores aos alunos do curso técnico em informática. Podemos sintetizar em dois cenários: um na preparação do aluno com lógica de programação e algoritmos antes deste criar aplicativos de fato, outro, na introdução paralela destes conhecimentos com programação e a produção de aplicativos. Devido ao volume elevado de informações e disciplinas aplicadas, onze/ ou mais no ano letivo, e, ao pouco e limitado tempo, exige-se que haja uma maior didática e otimização para introduzir o conhecimento de programação, aplicando-se algoritmos simultaneamente a uma linguagem, objeto desta pesquisa.

Os Institutos Federais ofertam cursos técnicos de informática que necessitam de conhecimentos sobre algoritmos e programação de computadores. A disciplina responsável por este conhecimento é uma das que mais reprovam em todo o mundo. A dificuldade na introdução dos conceitos iniciais de programação é amplamente conhecida nos meios acadêmicos e a busca por reduzir as retenções e evasões em consequência dos resultados desta disciplina é objeto deste trabalho. Buscar a linguagem de programação adequada, que favoreça este aprendizado, nesta etapa,

pode potencializar a melhora dos resultados acadêmicos, facilitar tanto o trabalho do docente quanto a formação de melhores técnicos de programação.

1.6 ESTRUTURA DO TRABALHO

Esta pesquisa apresenta-se da seguinte forma:

Primeira parte: Apresenta a introdução situando o problema da pesquisa, a contextualização, as questões da pesquisa, os objetivos gerais e específicos, a justificativa.

Segunda Parte: Expõem o Referencial Teórico, o que os pesquisadores têm a dizer sobre os assuntos levantados no trabalho.

Terceira parte: Demonstra a aplicação passo a passo das metodologias utilizadas. Contempla tanto os valores e pesos atribuídos, os cálculos utilizados quanto as fórmulas em planilha para possibilitar a reprodução e validação dos métodos apresentados.

Quarta parte: Descreve as metodologias: AHP, TOPSIS, FUZZY TOPSIS e AHP FUZZY TOPSIS.

Quinta parte: Relata os resultados, a discussão e a conclusão de acordo com os objetivos, atendendo as questões apresentadas e sugerindo a alternativa a ser utilizada.

2 REFERENCIAL TEÓRICO

2.1 LINGUAGENS DE PROGRAMAÇÃO E ALGORITMOS

A linguagem de programação deve ser bem definida, pois é o recurso para implementar um algoritmo através de código de programação, a fim de descrevê-lo de forma compreensível ao computador, criando condições para que esse conjunto de instruções seja executado (GUDWIN, 1997).

Um algoritmo constitui uma sequência detalhada de ações, para quando executado, realizar a solução de um problema ou realizar uma tarefa, normalmente associada à computação, mas também o algoritmo é bastante utilizado em áreas como: engenharia, administração entre/ outras. Os algoritmos podem ser trabalhados independentemente de linguagens de programação, uma vez que são a reprodução da lógica computacional na realização de tarefas. Necessitam de regras sintáticas e de estrutura formal com um universo limitado de comandos e procedimentos pré-definidos (MEDINA; FERTING, 2006).

A razão dos algoritmos serem difíceis de entender para os novatos em programação está na modificação dinâmica dos valores dos elementos sobre as estruturas abstratas de dados. As animações, por conectarem conceitos abstratos a objetos e situações reais, podem facilitar o entendimento de algoritmos, no entanto, não são suficientes para o entendimento dos detalhes, sendo necessário conhecimentos extras sobre matemática básica, tecnologia de ambiente de programação, linguagem de programação e como converter o conhecimento da lógica em um programa (VÉGH; STOFFOVÁ, 2017).

Os algoritmos não necessitam de softwares para serem ensinados, entretanto, é desejável que possam ser visualizados “rodando” a partir de uma linguagem de programação ou tecnologia que pode ser tanto “online” quanto “offline”.

Todo software é resultado de programação, esta é feita a partir de linguagens, métodos de criação e desenvolvimento de códigos (utilizando-se da aplicação de algoritmos), que geram instruções entendidas pelos computadores. As linguagens de programação possuem estrutura gramatical própria, regras sintáticas e semânticas. Cada linguagem de programação exige um tempo para: criação, produção e ajustes de programas, tanto sobre os recursos de escrita do código, quanto na percepção e correção dos erros que possam surgir, bem como, sobre o volume dos códigos

gerados e sobre a qualidade e aprimoramento dos mesmos (MEDINA; FERTING, 2006).

2.2 LINGUAGENS – DESENVOLVIMENTO DE SOFTWARES

Como qualquer linguagem, a linguagem de programação possui estrutura gramatical própria, regras sintáticas e semânticas, impondo limites ao desenvolvimento, com variações sobre a aplicação de algoritmos sobre as estruturas de dados, de controle, sobre os recursos de programação disponíveis nesta, além da compatibilidade com a plataforma (SEBESTA, 2018).

O desenvolvimento de aplicativos, depende da plataforma e correlaciona sistema operacional, hardware, e linguagens e/ou IDE's para a programação. Há centenas de linguagens no mercado, algumas não necessitando ser instaladas fisicamente nos computadores do programador, que fazem uso de recursos de programação online oferecidos em computação em nuvem. Os recursos de hardware e software tendem a obsolescência com rapidez. A computação em nuvem é mais uma tecnologia que oferece recursos de suporte às infraestruturas de TI complexas. Traz a possibilidade de criação de ambientes virtuais para o ensino e oferece os recursos de hardware e software sob a forma de serviços (SOUSA; MOREIRA; MACHADO, 2009).

A definição dos requisitos para o desenvolvimento de programas de computador é vista como um gargalo, um ponto crítico no desenvolvimento de aplicativos, a fonte de grande parte dos problemas e falhas de softwares (THIRY; ZOUCAS; GONÇALVES, 2010). As tecnologias disponíveis hoje são bastante amplas e possuem demandas diferentes. Além das requisições do software e do hardware, tem-se as requisições do usuário deste aplicativo a ser desenvolvido, a plataforma, suas características e tecnologias exigidas para o novo aplicativo, bem como considerar que mesmo depois de desenvolvido, o aplicativo pode, e normalmente recebe uma série de melhorias e adaptações para ampliar sua qualidade na comunicabilidade, no uso de recursos, na potencialização de resultados e na correção de algoritmos (FERNANDES; MACHADO, 2018).

Há uma tecnologia que se destaca de todas as outras existentes, pela sua imensa penetração na maioria das plataformas: a Internet. A internet, constituída por uma rede de comunicação mundial, permite a integração da grande maioria dos

dispositivos, das mídias, senão todos. Faz com que diferentes plataformas acessem os mesmos conteúdos, sendo portanto, essencial que seja valorizada pelo seu significado e importância (MORAN, 1997). Os sítios, ou sites, que são locais virtuais onde as pessoas e empresas tramitam informações pela internet, utilizam a linguagem HTML que não trabalha sozinha, mas, integrada ao CSS e ao Javascript, além de/ outras tecnologias (MILETTO; BERTAGNOLLI, 2014).

Um “frame”, quadro em português, constitui uma estrutura de dados para representar uma situação, um ambiente (MINSKY, 2019), um framework, uma estrutura para realizar um estudo, um trabalho (ARKSEY; O'MALLEY, 2005). Um framework é uma estrutura alternativa em termos de níveis de processamento (CRAIK; LOCKHART, 1972).

Um framework, portanto, estabelece um ambiente com ferramentas direcionadas a realização de trabalhos específicos, muitas vezes integrando diversos recursos, aplicativos independentes, em uma mesma interface, outras, disponibilizando neste, um conjunto de recursos que de outra forma exigiria a utilização em paralelo de diversos aplicativos (PREZOTTO; BONIATI, 2014). Há frameworks para dar suporte a diversas linguagens, como por exemplo: para o PHP: Laravel; CodeIgniter; Symfony; Phalcon; CakePHP dentre/ outros, ou para o Java: Blade; Maven; Dropwizard; Hibernate e mais uma dezena de/ outros. Os frameworks são um catalisador para profissionais de programação.

O desenvolvimento de softwares ou processo de software constitui-se de um conjunto de atividades e de resultados associados, como análise de requisito, definição de funcionalidades e de restrições, projeto e implementação das especificações pela codificação e validação para garantir as funcionalidades, gerados a partir de uma diversidade de métodos de programação. A metodologia tradicional orientada a documentação, cede lugar para metodologias ágeis e “Extreme Programming” que possuem foco no cliente, na sua colaboração e feedback constante, bem como em uma abordagem incremental e dinâmica na produção de softwares com respostas rápidas às mudanças (SOARES, 2004).

As plataformas de programação são constituídas por um ambiente diversificado de tecnologias: sistema operacional, hardware compatível, ambiente integrado de desenvolvimento (IDE) e linguagens de programação, usadas para a programação de softwares nas respectivas plataformas (SOARES, 2004). Têm uma grande diversidade de sistemas operacionais para desktop como: Windows, Mac Os, Linux,

Chrome OS, para smartphones como: Android, IOS, Windows; para as IDEs como: Eclipse, Visual Studio, Sublime Text, WebStorm, NetBeans, NotePad ++, PyCharm. As linguagens como Java, C, Python, C++, C#, JavaScript são alguns exemplos dessa diversidade de tecnologias e ambientes de programação.

A programação, arte/ ou técnica na construção e formulação de algoritmos sistematicamente, constitui uma atividade voltada para a solução de problemas. Faz parte da formação básica da estrutura curricular para um desenvolvedor de programas (SOARES, 2004).

2.3 LÓGICA FUZZY

Lógica fuzzy, ou lógica difusa, ou multivalorada trabalha com a ideia de um conjunto difuso, nebuloso, não preciso ou variável. Desenvolvida em 1965 por Zadeh, tinha como objetivo trabalhar com classes de objetos que não tivessem critérios de associação precisamente definidos. Ao contrário da lógica clássica que trabalha com valores exatos, falso x verdadeiro, ligado x desligado, sim x não, 0(zero) ou 1(um), na lógica fuzzy trabalha-se com meias verdades, são considerados os valores entre o 0 e 1, ou próximos ao 1. A modelagem matemática sobre informações imprecisas é possibilitada pela teoria fuzzy, aplicada sobre dados numéricos e sobre a subjetividade para avaliação através de conceitos como: muito verdadeiro, verdadeiro, pouco verdadeiro, bom, muito bom, excelente (MARRO *et al.*, 2010).

A lógica Fuzzy que fora projetada para ser utilizada em ambientes de incertezas, constitui uma metodologia robusta para quantificar dados imprecisos, nebulosos das informações (KAHRAMAN; GÜLBAY; ULUKAN, 2006).

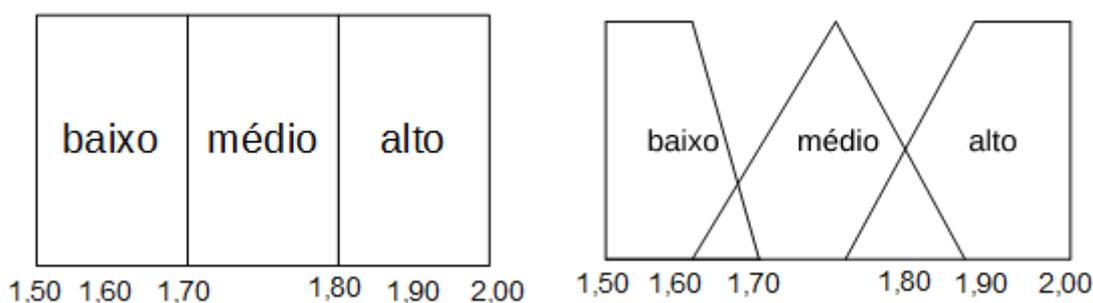
A concepção clássica da lógica fuzzy sobre a teoria de conjuntos difusos por Zadeh em 1965, diz que os valores devem atender uma polarização entre 0 e 1, ou entre sim e não. Os valores nas avaliações, na utilização desta lógica mesclada a outros métodos de multicritério como métodos AHP, TOPSIS e/ outros, determinam, a partir de regras que estes valores de julgamento, de alguma alternativa sob um critério, pertencem a um conjunto fuzzy (valor 1), ou não pertencem (valor 0). Para cada critério trabalhado, teremos um valor de referência, pico, que podem ter diferentes grandezas. Assim, pode-se ter desde pequenos valores numéricos de referência até o tamanho que for necessário, pode-se utilizar funções como parâmetros, para determinar os conjuntos de valores $f(x)$ (GOMIDE *et al.*, 1995).

Conjunto nebuloso, fuzzy ou difuso são a tradução quantitativa dos valores qualitativos dos termos linguísticos, expressões, nomes imprecisos, como ocorre quando classificamos pessoas pela estatura: baixa, média e alta. A variação da percepção entre a diferença entre o que é verdade, pouco verdade, ou muito verdade chamamos de grau de pertinência (CALACHE *et al.*, 2019).

Expressões linguísticas não são números, portanto, não podem ser calculadas, assim, deve-se substituir as expressões por valores que necessariamente não exigem um tipo e grandeza específicos, podendo, a título de exemplo, ser um valor inteiro com 1000 (mil) unidades de intervalo, ou valores reais com intervalos inferiores a 1 (um) como 0,01. Obviamente mantendo-se a coerência e a proporção.

Assim, ao fazermos uma afirmação, como: 1,90 m é uma estatura considerada alta para os homens, obviamente dependerá da percepção do avaliador, não podemos descartar 1,89 m ou 1,80 m como também sendo de estatura alta. A lógica fuzzy aceita que todas as coisas podem ter graus de pertinência. Neste contexto, uma vez determinado os limites inferiores e superiores para o pertencimento de um valor dentro de um conjunto, permite percepção de declaração falsa ou verdadeira, pertencente/ou não pertencente a um conjunto. Para essa definição são criadas regras, funções que determinam qual estatura é considerada alta dentro de um grau de pertinência. Uma pessoa que tivesse estatura de 0,95, teria 50% da altura considerada alta, $0,95/1,90 = 0,5$ ou 50%, ou seja, dividindo-se a estatura a ser comparada com o valor de referência obtemos o percentual relativo de altura para o respectivo valor e a partir da função ou regra determinar se pertencente ou não ao conjunto em questão (MARRO *et al.*, 2010).

Figura 1 Funções de pertinência para a variável altura



Fonte: adaptação de Gomide e Didwin (1994).

A diferença entre a lógica convencional e a lógica fuzzy pode ser ilustrada a partir dos gráficos abaixo. Pela lógica convencional, no Gráfico da Figura 1, a partir da observação de dois elementos $x_1 = 1,65$ e $x_2 = 1,70$, vê-se que estariam em classes diferentes, x_1 pertencendo a classe baixo e x_2 a classe médio, diferentemente do Gráfico 2 que representa a lógica fuzzy, onde tanto o x_1 quanto o x_2 têm graus de pertinências aos conjuntos fuzzy determinados.

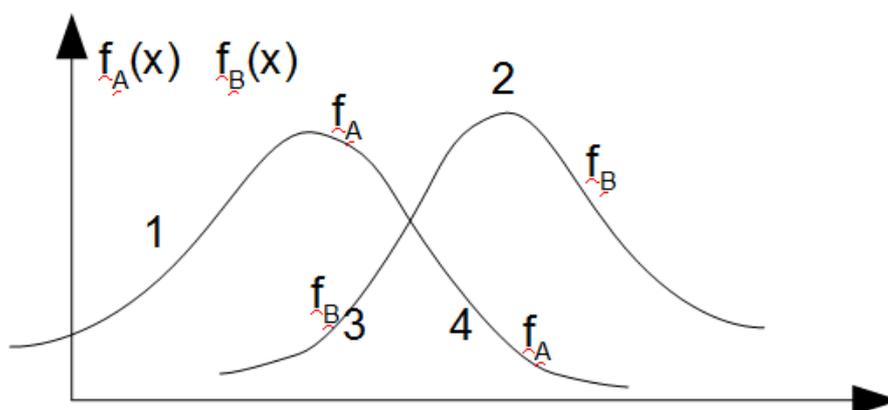
A declaração de função para os gráficos 1 e 2 podem assim ser representados:

$$f(x) = \{1 \dots 0 \dots \text{se, e somente se, } x \in A \text{ se, e somente se, } x \notin A\} \quad (1)$$

$$\mu(x) = \{100 \leq \mu(x) \leq 1 \text{ se, e somente se, } x \in A \text{ se, e somente se, } x \notin A\}$$

$$\text{se } x \text{ pertence parcialmente a } A \quad (2)$$

Figura 2 Relação de pertinência



Fonte: adaptado de Gomide *et al.* (1995).

Na função (item 2 da Figura 2) o intervalo de pertinência $[0,1]$, onde 0(zero) indica que o elemento não pertence a um determinado conjunto e 1 indica a pertinência a um conjunto, os valores intermediários indicam graus de pertinência intermediária ou parcial, dessa forma, fica determinado que uma sentença possa assumir valores parcialmente verdadeiros e parcialmente falsos, podendo um mesmo elemento apresentar grau de pertinência 0(zero) para vários conjuntos fuzzy no problema (MARRO *et al.*, 2010).

Zadeh (2015) trata deste conjunto difuso como uma classe de elementos com grau de associação contínua, que utiliza uma maneira natural de tratar problemas que tenham imprecisão quanto a definição de critérios de associação à classe. Considerando-se X os objetos, ou, espaço de pontos observáveis, sendo um elemento

qualquer de X representado por x . Utiliza um exemplo que atribui a um objeto um grau de associação variando entre 0 e 1. O conjunto fuzzy A em X , representado por $f_A(x)$, que é uma função de associação que relaciona cada ponto em X , um número real entre $[0, 1]$. O valor de $f_A(x)$ em x representa o “grau de associação” de x em A . Ao representar valores dessa função como por: $f_x(0) = 0$; $f_A(1) = 0$; $f_x(6) = 0,01$; $f_x(11) = 0,2$; $f_x(103) = 0,95$; $f_x(400) = 1$ Gerou-se uma caracterização precisa, mesmo que subjetiva, de A relacionando $f_A(x)$ como uma função em R (valores reais).

Ao contrário do valor dos sistemas lógico binários, onde o valor assumido só poderia ser 1 ou 0, na lógica fuzzy pode ser um subconjunto fuzzy de um conjunto parcialmente ordenado. Noutro sistema lógico de representação de valores: multi-valores, o valor referência de uma proposição pode ser, num intervalo de valores, um elemento de um conjunto finito de valores. Na lógica fuzzy, ou nebulosa, os valores podem ser gerados subjetivamente, a partir da linguagem humana, associando-se cada expressão linguística ao valor unitário, subconjunto fuzzy.

Observa-se que há valores tanto em $f_A(x)$ quanto em $f_B(x)$ em comum, demonstrando a interseção e união de dois conjuntos difusos (Figura 2), 1965. Os segmentos de curva 1 e 2 compõem a função de associação da união, e os 3 e 4, a interseção.

A lógica clássica tem apenas um valor para representar um conjunto, na lógica fuzzy a indeterminação na representação de valores em uma classe é representada por três valores de verdade, o limite abaixo, o valor central representando 100% de proximidade e o valor limite acima. Ou seja, os valores até o limite inferior e até o limite superior apresentam uma proximidade percentual superior a 0% e inferior a 100% do valor verdade do conjunto fuzzy, esse conjunto de valores cria a lógica de três valores de verdade (GOMIDE *et al.*, 1995).

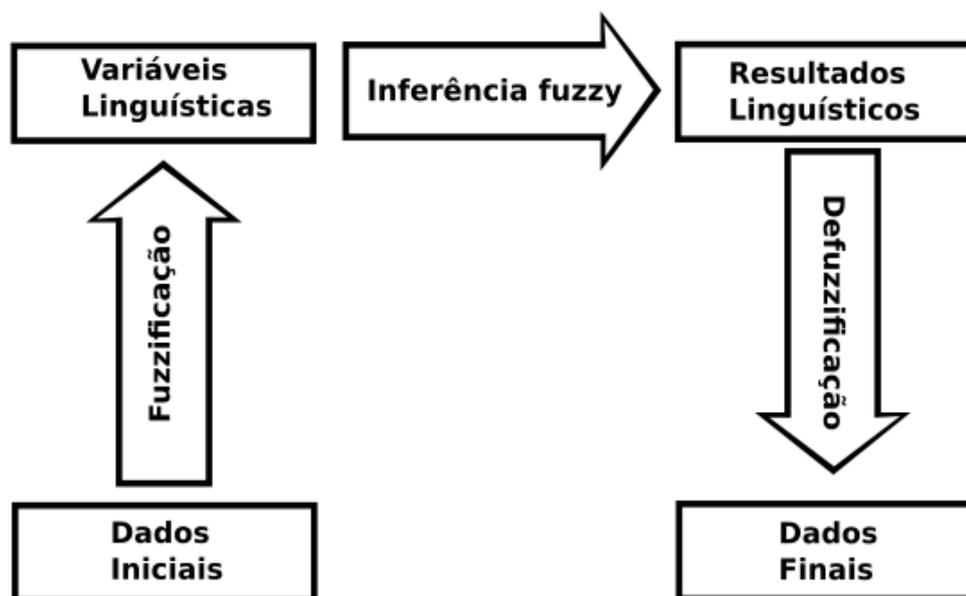
De acordo com a natureza dos dados, trabalha-se com uma grandeza, como comprimento, temperatura ou outro. A definição do conjunto fuzzy, que representa o comportamento de variação desses dados, é obtida a partir da observação do comportamento dessas grandezas, que promove a definição do limite inferior e superior de cada intervalo de um conjunto fuzzy (GOMIDE *et al.*, 1995).

Sobre a Fuzzificação

O Sistema lógico fuzzy constitui-se em três operações: Fuzzificação, Inferência e Defuzzificação. Na primeira operação os dados de entrada do problema são analisados e vistos sob a ótica de variáveis linguísticas a fim de que toda imprecisão ou incerteza seja considerada; na inferência as regras ou proposições são criadas sobre as variáveis do problema, definindo-se a validade da regra e o resultado obtido a partir dessas regras; a Defuzzificação consiste na conversão das variáveis linguísticas em números e os cálculos matemáticos para encontrar os resultados. O gráfico abaixo demonstra esse sistema (AGUADO; CANTANHEDE, 2010).

A fuzzificação dos valores ou, a geração de intervalos anteriores e posteriores ao pico, ao valor de referência considerado, possui sua variação intrinsecamente associada a natureza e a circunstâncias dos dados analisados, exige coerência no raciocínio, para que se possa validar as regras e os intervalos (MARRO *et al.*, 2010).

Figura 3 Sistema Lógico Fuzzy



Fonte: Adaptado de Aguado e Cantanhede (2010).

Para exemplificar podemos citar uma escala de valores de referência a partir de uma expressão linguística, ou notas de avaliação subjetiva, ou dados objetivos, ou ainda resultado de funções como na tabela 1, exemplificando sobre significância e sobre estatura.

Tabela 1 Definição e explicação de valores associados sugeridos

Significância/Importância	Estatutura	Valor Sugerido
Insignificante	Muito Baixa	1
Pouco significativa	Baixa	2
Média significância	Média	3
Muito significativa	Alta	4
Extrema significância	Muito Alta	5

Fonte: Adaptado de Calache *et al.* (2019)

No exemplo, foram disponibilizados cinco termos, entretanto, pode-se utilizar a quantidade de termos que for necessária e adequada a solução do problema. As expressões literais serão convertidas para valores numéricos, os valores numéricos necessitam ter o grau de pertinência definido para gerar os valores abaixo (l) e acima (u), uma referência a *less* e *up* utilizados em todos os contextos do trabalho. O conjunto de falores fuzzy serão representados na ordem [l, f, u], onde (f) representa o valor de referência, ou valor central.

Tabela 2 Valores fuzzy de sugestão

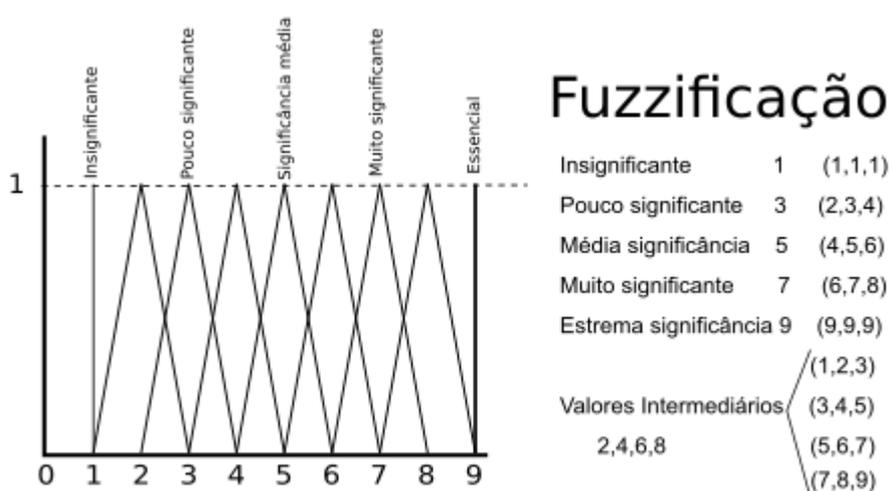
Very low			low			medium			high			Very high		
1			2			3			4			5		
l	f	u	l	f	u	l	f	u	l	f	u	l	f	u
1	1	2	1	2	3	2	3	4	3	4	5	4	5	5
l	f	u	l	f	u	l	f	u	l	f	u	l	f	u
0	1	3	1	3	5	3	5	7,5	5	7,5	9	7,5	9	10

Fonte: Adaptado de Kahraman (2008).

Na Lógica Fuzzy tradicional o intervalo de valores utilizado é binário, ou seja, deve ser zero ou um, para indicar se uma declaração é falsa ou verdadeira; na Lógica de Conjuntos Fuzzy, surge a ideia de fuzzificação a partir do conceito de grau de pertinência, valores podem ser parcialmente verdadeiros ou parcialmente falsos. A partir da função de pertinência define-se os conjuntos fuzzy que dependerão do problema a ser modelado, funções que podem ser lineares ou não lineares. A Defuzzificação é o processo para se obter uma saída numérica, cujo método mais comum é o centroide que aponta uma linha vertical ao meio de um conjunto agregado (MARRO *et al.*, 2010).

Os valores sugeridos poderiam ter outros valores com grandezas maiores ou menores, respeitando a proporcionalidade das avaliações. Todos os valores deverão ter definidos os limites inferiores e superiores, fuzzificação/defuzzificação, conforme o gráfico de pertinência sugerido no exemplo da Figura 4 que fora utilizado na aplicação do método AHP e adaptado a este método. No exemplo original a unidade 1 tem valor de equivalência e grau de importância crescente paralelamente ao aumento do valor até o valor máximo 9 (KANNAN *et al.*, 2013).

Figura 4 Valores para fuzzificação



Fonte: Adaptado de Kannan *et al.* (2013).

No exemplo, utilizou-se a variação de uma unidade para os limites inferiores e superiores, entretanto, a quantidade de unidades de variação inferior e superior ao ponto central ou de representação do conjunto fuzzy, poderia ser de mais unidades ou de valores reais dependendo da natureza do problema e dos dados encontrados, como sugerido abaixo com variação de duas unidades.

2.4 AUXÍLIO MULTICRITÉRIO À DECISÃO

A literatura sobre Auxílio Multicritério à Decisão (AMD) em inglês Multicriteria Decision Aid (MCDA), Multicriteria Decision Making (MCDM), utiliza várias disciplinas e métodos para a criação de modelos de decisão, como: teoria dos jogos, programação linear, modelos em árvore, método AHP, AHP Borda, Fuzzy entre/outras (COSTA, 2006). Há diversos métodos de auxílio à tomada de decisão por múltiplos critérios: AHP, ELECTRE, MAHP, sem, no entanto, poder afirmar que um é

melhor do que/ outro em qualquer tipo de situação. Os métodos possuem em comum fatores que os diferenciam quanto a suas propriedades: Forma como os critérios são julgados; Definição de pesos e valores (importância, preferência ou possibilidade) dos critérios ou alternativas; Utilização dos pesos ranqueando as alternativas sob os critérios (GUGLIELMETTI; MARINS; SALOMON, 2003).

As tomadas de decisões têm sua complexidade aumentada na medida que há múltiplos critérios, evidenciando a necessidade de adoção de técnicas e ferramentas para gerar melhor discernimento sobre a solução a ser alcançada pelo tomador de decisão (COSTA, 2006).

A utilização de avaliadores de alto conhecimento sobre o objeto da pesquisa constitui fator de eficácia sobre os resultados, sendo estes, responsáveis pelo grau de importância de cada critério sobre as alternativas (COSTA, 2002).

Para todos os métodos abordados e descritos no referencial teórico, será adotado uma representação dos critérios, das alternativas e dos decisores (Quadro 1).

Quadro 1 Representação de critérios, alternativas e decisores

Elementos representados	Representação
Critério 1	C1
Critério 2	C2
Critério 3	C3
Alternativa 1	A1
Alternativa 2	A2
Alternativa 3	A3
Decisor 1	D1
Decisor 2	D2
Decisor 3	D3

Fonte: próprio autor.

2.5 CRITÉRIOS

Os critérios apresentam diferentes dimensões das alternativas, podendo alguns apresentar um conflito ou até mesmo uma oposição ao outro, como ocorreria entre “custo” e “lucro”, podem trabalhar unidades diferentes como: peso, valor, volume, frequência, tamanho e tantas outras possibilidades. Devem avaliar o desempenho das alternativas verificando todas as propriedades relevantes, sem redundâncias ou

superposições propiciando a compreensão de decisores e avaliadores (COSTA, 2002).

Os múltiplos critérios aumentam a complexidade do problema nas tomadas de decisão, fazendo-se necessária a adoção de técnicas e ferramentas utilizadas pelo tomador de decisão, a fim de que alcance uma solução, tendo discernimento entre estes (COSTA, 2006).

As diferenças mais relevantes na classificação de critérios, consiste em critérios focados na transição para outras linguagens de programação e critérios direcionados as construções básicas de programação sobre a linguagem (PROKOP *et al.*, 2019).

Mesmo na pesquisa de Parker, Ottaway e Chao (2006), mais referenciada em trabalhos com propostas semelhantes, os critérios levantados não separam as alternativas entre linguagens utilizadas na introdução da programação e as de fato utilizadas para produtividade. Os critérios valorizam características não relevantes na fase inicial de formação do programador como: a capacidade de lidar com várias tecnologias, segurança de dados, rapidez de processamento, portabilidade, Suporte Orientado a Objetos dentre/ outros que, embora importantes para a produção, para o desenvolvimento de aplicativos, não fazem sentido para o aprendiz iniciante que ainda não detém os rudimentos da programação.

São critérios comumente encontrados para seleção de linguagens ou frameworks:

- 1) Popularidade, o quão conhecido e reconhecido,
- 2) Tamanho da Comunidade, maior comunidade, maiores são as discussões e as atualizações,
- 3) Suporte, qual a qualidade e facilidade no suporte,
- 4) Documentação, manual técnico e sua qualidade, disponibilidade e volume da documentação oficial e,
- 5) Disponibilidade de Recursos, Informações do framework na Internet, na Literatura e em artigos científicos.

Há que se observar que o iniciante não tem conhecimento de algoritmo, do ambiente computacional e de linguagem, muito menos para utilizar um Framework.

Os professores Freire e Prado (1999) trabalham o uso do computador como ferramenta de construção do conhecimento em um sentido amplo. Propôs uma análise sobre softwares utilizados na educação e sobre como produzir a aprendizagem, no

entanto, se limita a três critérios que não levam em consideração a condição inicial do aprendiz, nem o objetivo específico voltado para a formação de programadores.

2.6 MÉTODO AHP

O método AHP é o método de tomada de decisão mais amplamente utilizado e conhecido na busca de solução de conflitos negociados e em problemas com múltiplos critérios (MARTINS, 2009).

O método AHP possui um desejável benefício na solução de problemas de decisão, uma vez que permite trabalhar com os aspectos qualitativos e quantitativos a partir dos julgamentos, das comparações paritárias, tanto sobre os dados apresentados, quanto sobre as percepções, intuições e experiências do(s) decisor(es) (SAATY, 1994).

O método AHP permite calcular a razão de consistência entre os pares, ou, estabelecer a relação coerente entre objetos ou ideias, garantindo a proporcionalidade entre os valores de julgamento atribuídos aos elementos do problema (SAATY, 2000). Este método foi elaborado a partir de uma representação do reflexo natural da mente humana, que necessita escolher dentre muitos elementos, os classificar em grupos com propriedades comuns, repetindo esse processo, reagrupando e hierarquizando em níveis até chegar a uma decisão, formada assim a hierarquia com níveis estratificados (SAATY, 1994).

A questão a ser definida por SAATY em sua teoria, reside na definição dos pesos individuais das alternativas, ou, de como o nível mais baixo da hierarquia afeta o nível mais alto, o objetivo geral, proporcional aos pesos relativos aplicados aos critérios. SAATY com a teoria da análise hierárquica busca reduzir as falhas no processo de tomada de decisão utilizando uma sequência de comparações par a par de forma relativamente simplificada, calculável a partir de quaisquer planilhas eletrônicas, considerando a capacidade da mente humana de classificar e julgar, além de dados numéricos quantificáveis, dados subjetivos como o conhecimento, sentimentos, emoções humanas e intuição.

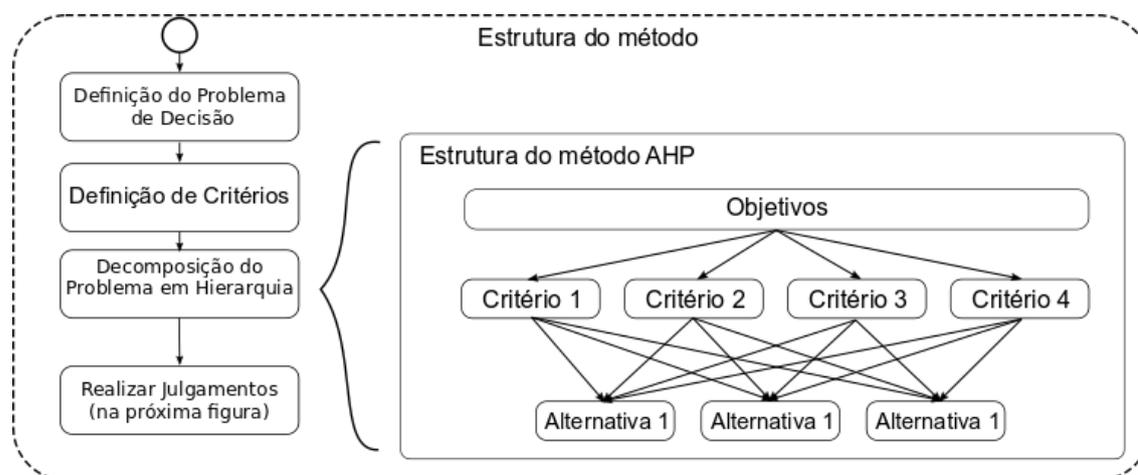
Os tomadores de decisão têm preferência pelo método AHP por este ser mais “amigável”, por ter maior facilidade no entendimento, escolha essa percebida no volume de publicações sobre o AHP muito superior a outros AMD (GUGLIELMETTI; MARINS; SALOMON, 2003).

Os métodos AHP (*Analytic Hierarchy Process*), proposto por Saaty (1977); ELECTRE (*Élimination Et Choix Traduisant la Réalité*), proposto por Roy (1968), PROMETHEE por Brans e Vincke (1985), Macbeth por Bana, Costa e Vasnick (1981) e MAHP, Multiplicative AHP por Lootsma (1993), são métodos com definições próximas, mas que diferem na em suas aplicações. Os métodos possuem em comum a forma de julgamento dos critérios, a definição dos respectivos pesos e o ranqueamento das alternativas à luz desses critérios (GUGLIELMETTI; MARINS; SALOMON, 2003).

A aplicabilidade do modelo AHP em planilhas, devido à facilidade de aplicação e potencial de solução de problemas de decisão, sem a necessidade de softwares proprietários. Favorecendo simulações, análise de sensibilidade e resultados para tomada de decisão bem embasados (PACHECO; GOLDMAN, 2016).

O trabalho de Parker, Ottaway e Chao (2006), pioneiro e referência para inúmeros trabalhos voltados para o levantamento de critérios de seleção de linguagens de programação e/ou tecnologias para cursos de introdução a programação, focado no público universitário, não separa a série inicial das demais, faz uma abordagem como de um fluxo contínuo do conhecimento necessário ao desenvolvimento do programador. Isto impacta diretamente na eleição dos critérios, também não sugere pesos sobre os critérios, apenas aponta que o peso de cada critério deve ser multiplicado pelos valores atribuídos, pelo julgador, a cada linguagem, a fim de gerar uma proporção adequada de cada julgamento. Parker *et al.* (2006) apontam para o método AHP como o melhor para fazer o ranqueamento. Em termos de consistência, de fato este método é mais utilizado academicamente, entretanto, ele deve ser aplicado por um avaliador, ou um grupo que faça o julgamento de forma única, o método De Borda Multicritério apresenta maior simplicidade computacional e reiterados resultados bem-sucedidos para o ranqueamento, além de ser aplicável a julgamentos de vários avaliadores.

Figura 5 Estrutura do método AHP



Fonte: adaptado de Costa (2002).

2.6.1. DESCRIÇÃO DO MÉTODO AHP

O método de Análise Hierárquica de Processos (AHP), do pesquisador Thomas L. Saaty, no ano de 1970 constitui um método multicritério com imensidade de aplicações, amplamente utilizado para tomadas de decisão. A decomposição do problema em problemas menores até o nível de clareza para em seguida formar relações que proporcionem, à luz dos critérios, a escolha da melhor alternativa (GOMES; GONZÁLEZ; CARIGNANO, 2004).

2.6.1.1. ETAPA 1: ESTRUTURA DO PROBLEMA EM HIERARQUIA

A primeira etapa é crucial para o desenvolvimento do método, nela decomponemos o problema em três partes bem definidas (Figura 5): objetivo geral para a decisão, os critérios relativos ao problema da decisão e, as alternativas mais adequadas disponíveis.

Esta etapa fundamental permite uma melhor compreensão e avaliação do método. A compreensão de sistemas complexos é otimizada quando agrupados os elementos em conjuntos afins, estruturando-os hierarquicamente e realizando os julgamentos da importância proporcional entre estes elementos dentro de um conjunto de prioridades (SAATY, 2000).

O primeiro nível corresponde ao objetivo, a essência do problema a ser solucionado, o segundo aos critérios, filtro e responsável pelo ranking do terceiro nível, as alternativas, soluções viáveis para a solução do problema.

2.6.1.2. ETAPA 2: DEFINIR PRIORIDADES ENTRE OS ELEMENTOS DA HIERARQUIA

Uma vez criada a estrutura hierárquica, a próxima etapa no processo de análise multicritério é a análise paritária entre os critérios estabelecidos a fim de definir o grau de importância entre eles (MORAES; SANTALIESTRA, 2008).

De acordo com a habilidade humana de perceber a relação entre objetos e situações, pode-se fazer a comparação par a par de objetos similares à luz de certos critérios, e quantificar, discriminar entre os pares de acordo com sua preferência por um ou outro critério. Definiu-se uma Escala Fundamental (item 2 da Figura 7) a ser aplicada sobre estes julgamentos paritários, composta de valores de 1 a 9, com a importância proporcional ao valor aplicado, sendo 1 de pouca importância até 9 de importância absoluta (WIND; SAATY, 1980).

Compara-se critério por critério atribuindo valores dentro da escala de valor para julgamentos paritários criada por Wind e Saaty (1980), realizar julgamentos (item 3 da Figura 7). Os valores refletem a importância de um elemento em relação ao outro, onde se atribui significação aos valores: 1 de equivalência, 3 de leve importância, 5 grande importância, 7 muito grande importância, 9 absoluta importância; os valores 2, 4, 6 e 8 representam valores intermediários utilizados para ajustes.

O critério considerado menos importante terá o valor recíproco inverso (item 2 da Figura 7), ou seja, $1/n$, onde n é o valor do julgamento atribuído ao elemento de maior importância.

Figura 6 Exemplo do método AHP

	C1	C2	C3	Passo W	Passo T	C1	C2	C3	T*W	IC
C1	1/1	3/1	1/2	W1	T1	C1	1/1	1/3	T1	0
C2	1/3	1/1	1/7	W2	T2	C2	3/1	1/1	T2	RC
C3	2/1	7/1	1/1	W3	T3	C3	1/2	1/7	T3	0
3	Qde de elementos			3,9	1	Matriz transposta			Tota l	3

Fonte: próprio autor.

O critério considerado menos importante terá o valor recíproco inverso, ou seja, $1/n$, onde n é o valor do julgamento atribuído ao elemento de maior importância. Faz-

se para a par até a completude da Matriz de decisão das comparações que expressam o grau de preferência entre os objetos.

Assim na figura 6, na primeira linha, segunda coluna (C1:C2) o valor 3 corresponde ao valor de 1/3 na segunda linha, primeira coluna, obviamente linhas e colunas iguais apresentarão sempre o valor 1, uma vez que o critério será o mesmo.

Calcula-se o autovetor[o professor disse que não representa um vetor] (item 4 da Figura 7), que constitui o resumo das preferências de cada nível hierárquico, para atribuição de uma importância relativa, a partir das Matrizes comparativas julgadas (item 3 da Figura 7).

De posse do autovetor, pode-se gerar uma matriz normalizada (item 5 da Figura 7), que constitui a distribuição proporcional de cada avaliação uniformemente, ou seja, o somatório de todos os valores do vetor (Matriz unidimensional) dos pesos será igual a 100%, ao dividirmos um valor por esse total, calculamos o percentual proporcional daquele elemento em relação ao todo. No exemplo da Figura 6 (Passo W), ao somarmos 1,14, 0,36 e 2,41 obtemos 3,92 que é igual a 100% dos valores somados, ao dividirmos 1,14 por 3,92, obtemos o valor de 0,29, ou 29% do todo.

A partir do autovetor normalizado, obtém-se o λ_{\max} (item 6 da Figura 7), que constitui o maior autovalor de cada elemento julgado, a ser utilizado no cálculo final de razão de consistência. O λ_{\max} é calculado em duas etapas, primeiro faz-se o produto da Matriz transposta de julgamentos pelo respectivo valor do vetor normalizado, depois faz-se o somatório de todos os valores. No exemplo da Figura 6 temos para a primeira alternativa, a soma dos valores transpostos: $1/1 + 1/3 + 2/1$ divididos por 0,29 igual a 0,97. O somatório de todos os valores: $0,97 + 1,02 + 1,01$ resultando no $\lambda_{\max} = 3,003$.

2.6.1.3. ETAPA 3: AVALIAR A CONSISTÊNCIA DAS PRIORIDADES RELATIVAS ENCONTRADAS

O cálculo da Razão de Consistência (RC) (item 7 da Figura 7), deve resultar em um índice de valor inferior a 10% para ser considerado consistente, para tanto, depende do índice de consistência (item 7 da Figura 7) conforme a tabela de índice de consistência randômico (item 9 da Figura 7) proposto por Saaty (COSTA, 2006).

No exemplo da Figura 6 o IC é calculado pela fórmula $IC = (\lambda_{\max} - n) / (n - 1)$ onde n representa o número de alternativas. Portanto $(3,003 - 3) / (3 - 1)$ será igual a

0,0015. Por último o valor de IC é dividido pelo valor correspondente da tabela de Saaty, (item 9 da Figura 7), resultando em um valor inferior a 0,1, indicando consistência nos dados.

2.6.1.4. ETAPA 4: CALCULAR A PRIORIDADE COMPOSTA

Aplicar os vetores de prioridade de cada alternativa em relação ao vetor de prioridades dos critérios e gerar a prioridade composta para evidenciar a alternativa melhor pontuada.

Esta etapa não faz parte do rol de ações deste trabalho, uma vez que o valor de cada alternativa será avaliado por diversos docentes na metodologia FUZZY TOPSIS.

Para efeito de entendimento da conclusão do método, a fim de chegar a um resultado sobre as alternativas, o procedimento utilizado para calcular o RC dos critérios deve ser feito para as alternativas em relação a cada um dos critérios.

Cada Matriz gerada para cada alternativa, terá um vetor de pesos resultante, especificamente a Matriz de critérios por critérios será aplicada sobre todos os respectivos vetores das alternativas para gerar um vetor normalizado resultante, ou seja, dispõem-se os vetores de cada critérios sobre as alternativas, e faz-se o somatório de posição relativa do vetor da alternativa pela posição relativa do vetor de critérios corresponde.

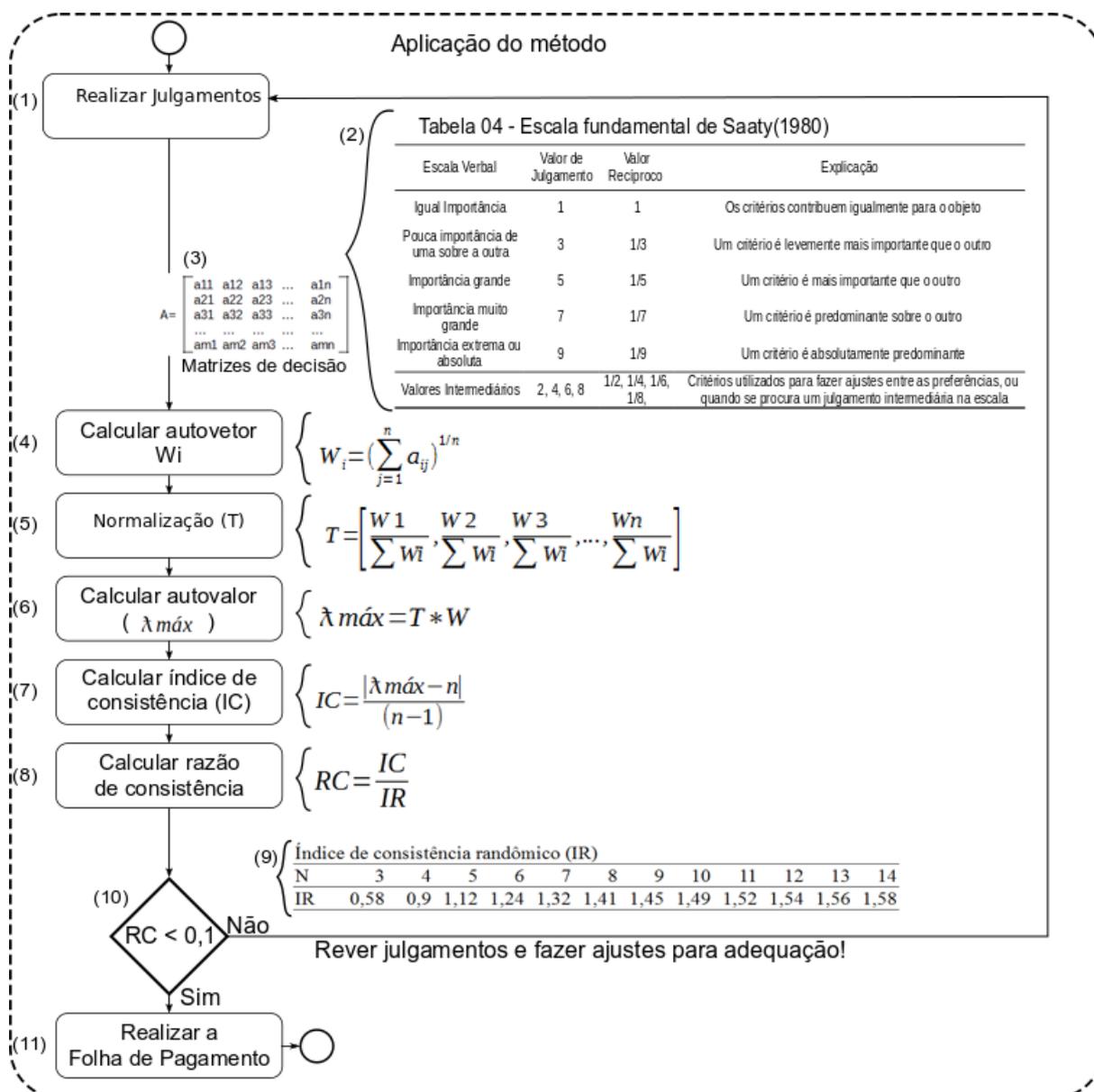
Quadro 2 Normalização sobre vetores consistentes

Folha de Pagamento sobre vetores consistentes						
Alternativas	C1	C2	C3	C4	CC	Vetor normalizado
A1	a1c1	a1c2	a1c3	a1c4	cc1	$'=a1c1*cc1+a1c2*cc2+a1c3*cc3+a1c3*cc3+a1c4*cc4$
A2	a2c1	a2c2	a2c3	a2c4	cc2	$'=a2c1*cc1+a2c2*cc2+a2c3*cc3+a2c3*cc3+a2c4*cc4$
A3	a3c1	a3c2	a3c3	a3c4	cc3	$'=a3c1*cc1+a3c2*cc2+a3c3*cc3+a3c3*cc3+a3c4*cc4$

Fonte: adaptado de Madi e Tap (2011).

Na coluna C1 temos o vetor de pesos deste critério para cada uma das alternativas e assim sucessivamente com os outros critérios, até o vetor de critérios por critérios que corresponde ao peso de cada critério correspondente. Faz-se o somatório dos vetores de cada alternativa a luz de cada critério, multiplicado pelo peso correlato.

Figura 7 Aplicação do método AHP



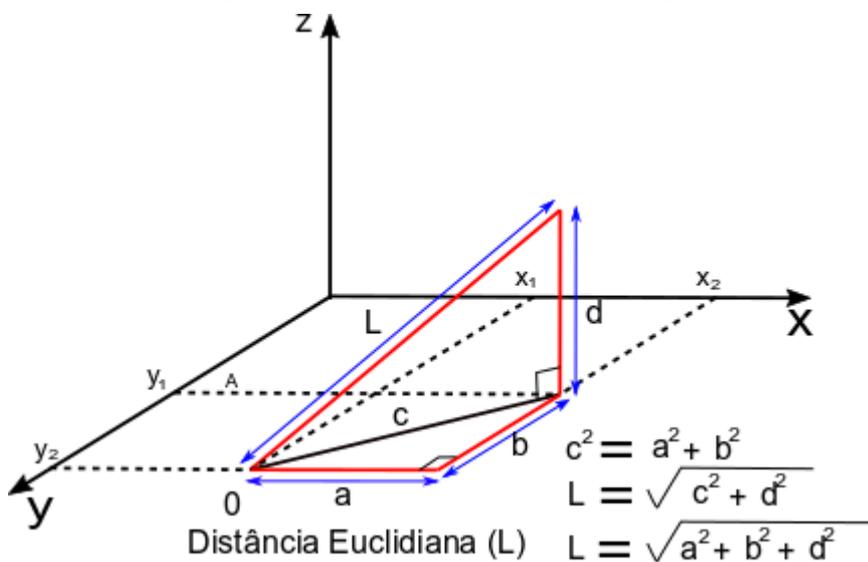
Fonte: próprio autor.

2.7 TOPSIS

O método TOPSIS (em inglês, Technique for Order Preference by Similarity to Ideal Solution) desenvolvido por Hwang e Yoon, constitui-se em um algoritmo, uma técnica de comparação de performance das alternativas. Esta metodologia trabalha com a ideia de distância entre pontos, através do teorema de Euclides (Figuras 8 e 9), aponta a melhor solução, a que mais se aproxima da solução ideal positiva e mais se distancia da solução ideal negativa. Destaca-se pela simplicidade e facilidade em avaliar ilimitadas alternativas e critérios simultaneamente, requerendo menor esforço,

tanto para a coleta de dados quanto para o processamento computacional (LIMA JUNIOR; CARPINETTI, 2014).

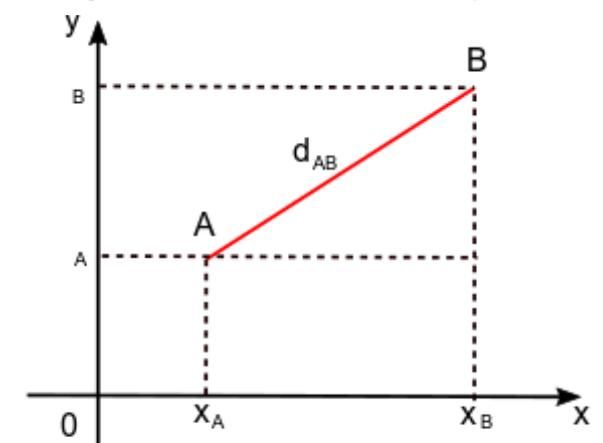
Figura 8 Distância de 3 pontos no espaço



Fonte: Lima Junior e Carpinetti (2014).

É um método de tomada de decisão multicritério caracterizado por uma Matriz de decisão, composta por critérios e alternativas. Trabalha com valores numéricos em formato crisp, valores estes, calculados utilizando a lógica aristotélica clássica (LIMA JUNIOR; CARPINETTI, 2014).

Figura 9 Distância euclidiana de 2 pontos



$$d_{AB} = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$$

Fonte: Lima Junior e Carpinetti (2014).

Uma vez montada a Matriz entre critérios e alternativas com seus respectivos valores, os passos para aplicar a metodologia TOPSIS podem ser resumidamente descritos em:

1° passo: Construção da Matriz de decisão (D), critérios, alternativas e julgamentos/valores/notas;

2° passo: Gerar a Matriz normalizada/ponderada (N), torná-los de mesma grandeza;

2.1° passo: Calcular o total a partir da raiz da soma dos quadrados de cada elemento;

2.2° passo: Gerar uma nova Matriz a partir da divisão dos valores julgados pela totalização;

3° passo: Incluir os pesos (P_i) para serem processados;

4° passo: Gerar uma nova Matriz a partir da multiplicação de cada elemento pelo respectivo peso;

5° passo: Criar o vetor de solução ideal positiva (PIS) A^+ e criar o vetor de solução ideal negativa (NIS) A^- ;

6° passo: Calcular as distâncias entre os dados das alternativas e do PIS S^+ e as distâncias entre os dados das alternativas e do NIS S^- ;

7° passo: Equacionar a melhor proximidade relativa, o melhor resultado observando o melhor e pior resultado para cada alternativa;

8° passo: Ranquear as respostas para encontrar o melhor resultado.

Para fins de exemplificação, considera-se uma Matriz de decisor D (Figura 11), composta por alternativas (linhas representadas por $A_i \rightarrow A_1, A_2, \dots, A_m$), critérios (colunas representado por $C_j \rightarrow C_1, C_2, \dots, C_n$) e, X indicando o desempenho ou, o valor, o rating da alternativa A_i sobre o critério C_j . Cada critério terá um peso correspondente, um vetor, representado por $P = (P_1, P_2, \dots, P_n)$, observando-se o mesmo número de critérios n. O peso w_j será calculado sobre o critério C_j sendo:

$i = \{1 \dots m\}$ e $j = \{1 \dots n\}$, o ponto (i, j), i representa a linha e j a coluna.

Exemplificando:

1º passo: Construção da Matriz de decisão, faz-se o levantamento de todas as notas, valores e/ ou julgamentos. Se necessário, como no exemplo abaixo, converter os valores literais em valores crisp para efetuar os cálculos. A Matriz (D) resultante, pode apresentar valores de diferentes grandezas.

Figura 10 Matriz de exemplos de valores de diferentes grandezas

	C1	C2	C3	C4
A1	10	1000	0,01	Fraco
A2	7	2500	0,05	Média
A3	3	3000	0,17	Forte
A4	1	3800	0,2	Excelente

Fonte: próprio autor.

Figura 11 Matriz (D) de exemplos de valores numéricos

	C1	C2	C3	C4
A1	10	1000	0,01	1
A2	7	2500	0,05	3
A3	3	3000	0,17	4
A4	1	3800	0,2	5

Fonte: próprio autor.

2º passo: Gerar a Matriz normalizada/ponderada (Z), torná-los de mesma grandeza (item 8 da Figura 17).

Independentemente das grandezas dos valores, faz-se a normalização para homogeneização dos dados para serem trabalhados dentro de uma escala de proporcionalidade, de tal forma que haja similaridade na grandeza. Os valores estarão todos dentro 0 e 1, mas, diferentemente do que ocorre na normalização no método AHP, o somatório não resultará no valor 1, equivalente a 100%; a razão disso acontecer reside no fato do cálculo da normalização ser baseada nas distâncias euclidianas, considerando cada valor como um ponto dentro de um plano cartesiano, ou seja, calcula as distâncias entre os pontos. Assim a normalização é baseada na divisão dos valores pela raiz da soma dos quadrados de todos os valores (item 8 da Figura 17) e não no somatório direto dos valores.

2.1º passo: Calcular o total a partir da raiz da soma dos quadrados de cada elemento.

$$r_{i1} = \sqrt{10^2 + 7^2 + 3^2 + 1^2} = 12,610$$

$$r_{i2} = \sqrt{1000^2 + 2500^2 + 3000^2 + 3800^2} = 5.539,855$$

$$r_{i1} = \sqrt{0,01^2 + 0,05^2 + 0,17^2 + 0,2^2} = 0,267$$

$$r_{i1} = \sqrt{1^2 + 3^2 + 4^2 + 5^2} = 7,141$$

$$R_{ij}[12,61; 5539,86; 0,27; 7,14]$$

2.2° passo: Gerar uma nova Matriz (K) a partir da divisão do valor do item pela totalização (item 8 da Figura 17);

FÓRMULA 1: Fórmula de normalização no método TOPSIS

$$D'_{ij} = \frac{D_{ij}}{\sqrt{\sum_{j=1}^n D_{ij}^2}}$$

Figura 12 Matriz (K) normalizada sobre o exemplo

	C1	C2	C3	C4
A1	0,79	0,18	0,04	0,14
A2	0,56	0,45	0,19	0,42
A3	0,24	0,54	0,64	0,56
A4	0,08	0,69	0,75	0,7

Fonte: próprio autor

3° passo: Incluir os pesos (Pi) para serem processados (item 9 da Figura 17);

Os pesos podem ser definidos e colocados por um decisor, ou podem ter origem em outras metodologias, como a metodologia AHP que permite a verificação consistente dos pesos, comparando-os uns com os outros.

Figura 13 Matriz de exemplo de Pesos

	P1	P2	P3	P4
Pj	3	1	4	2

Fonte: próprio autor.

4° passo: Gerar uma nova Matriz a partir da multiplicação de cada elemento pelo respectivo peso.

Uma vez feita a normalização, estes valores da Matriz (K) devem ser recalculados pelo vetor de pesos (P), atribuídos por algum decisor, para resultar em

uma nova Matriz (P_{ij}) = $[p_{ij}]_{m \times n}$, resultante da multiplicação destes valores pelos respectivos pesos: $p_{ij} = k_{ij} * r_{ij}$ (item 9 da Figura 17).

Figura 14 Matriz pesificada sobre o exemplo

	C1	C2	C3	C4
A1	2,38	0,18	0,15	0,28
A2	1,67	0,45	0,75	0,84
A3	0,71	0,54	2,54	1,12
A4	0,24	0,69	2,99	1,4

Fonte: próprio autor.

Então: $P_{11} = 3 * 0,008$, $P_{21} = 3 * 0,011$, $P_{31} = 3 * 0,026$, ... resultando em
 $P_{11} = 0,024$ $P_{21} = 0,034$, $P_{31} = 0,079$, $P_{41} = 0,238$

5º passo: Criar o vetor de solução ideal positiva (PIS) A^+ e criar o vetor de solução ideal negativa (NIS) A^- . (item 10 da Figura 17).

Há dois tipos de critérios possíveis, critério de custo e critério de benefício. Quanto menor o rating da alternativa para o critério de custo e maior para o critério de benefício melhor. Assim, em um exemplo prática sobre a compra de um bem, quanto menor o custo e maior a qualidade, conforto e/ ou recursos, melhor (KORE; RAVI; PATIL, 2017).

Neste momento a Matriz está preparada para gerar o vetor com a solução ideal que chamaremos A^+ , o vetor com a solução negativa ideal que chamaremos de A^- , conforme a descrição:

$A^+ = (p_{+1}, p_{+2}, \dots, p_{+n})$ (Indica os melhores valores)

$A^- = (p_{-1}, p_{-2}, \dots, p_{-n})$ (Indica os piores valores)

Sendo:

$p_{+j} = \{ \begin{array}{l} \text{maxi}(p_{ij}), \text{ para critério do tipo benefício e} \\ \text{mini}(p_{ij}), \text{ para critério do tipo custo} \end{array} \}$

$p_{-j} = \{ \begin{array}{l} \text{mini}(p_{ij}), \text{ para critério do tipo benefício e} \\ \text{maxi}(p_{ij}), \text{ para critério do tipo custo} \end{array} \}$

Essas funções permitem escolher os melhores e os piores desempenhos de cada critério, onde A+ e A- constituem as PIS e NIS respectivamente.

Este passo essencial é o que permitirá calcular a distância euclidiana de cada valor da alternativa em relação a PIS e a NIS.

Os valores da coluna de custo seriam invertidos, ou seja, supondo o critério C1 (primeira coluna) do tipo custo, para A+, selecionamos o menor valor que é 0,238 que é o menor e, 2,379 para A- que é o maior valor deste critério para as alternativas:

Figura 15 Matrizes PIS e NIS com valores resultantes sobre o exemplo

A+ [0,24	0,69	2,99	1,4]
A- [2,38	0,18	0,15	0,28]

Fonte: próprio autor.

6º passo: Calcular as distâncias entre os dados das alternativas e do PIS d_+ e as distâncias entre os dados das alternativas e do NIS d_- . Gera-se uma coluna/vetor para alternativas com a distância euclidiana ideal positiva s_+ e/ outra coluna ideal negativa s_- (item 11 da Figura 17).

Distâncias dos pontos: $S^+_i = p_j - p_{ij}$ e $S^-_i = p_j - p_{ij}$.

$$S^+_1 = \sqrt{(2,38 - 0,24)^2 +}$$

$$S^-_1 = \sqrt{(2,38 - 2,38)^2 +}$$

7º passo: Equacionar a melhor proximidade relativa, pela fórmula: $P = S_- / (S_- + S_+)$, considerando o melhor e pior resultado para cada alternativa (Figura 16). faz-se ao determinar a proximidade relativa P (item 12 da Figura 17), ou seja, calcular a distância levando em consideração a alternativa mais próxima da solução ideal positiva e mais afastada da solução ideal negativa através do cálculo:

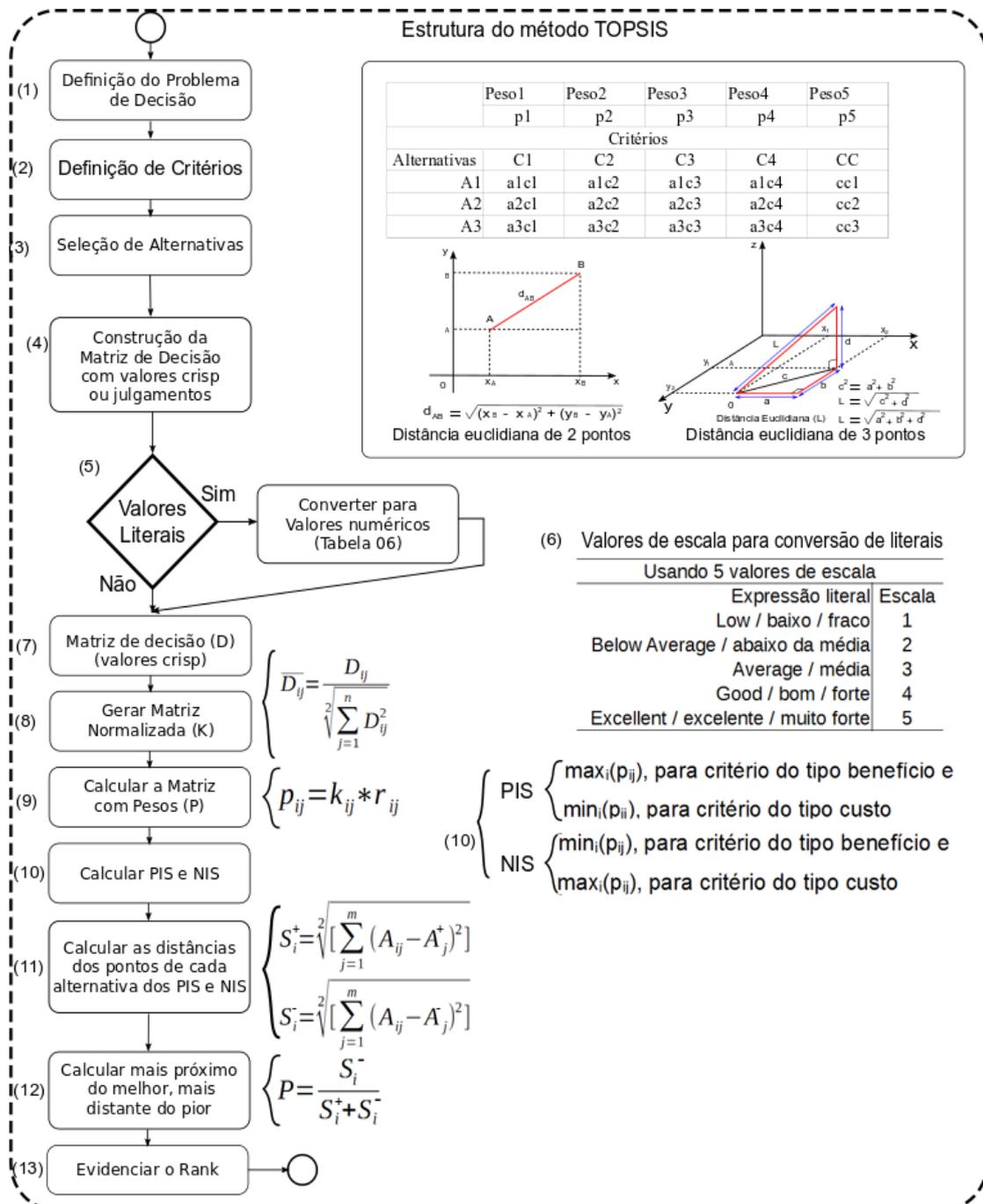
8º e último passo: Ranquear as respostas, conforme acima, para encontrar o melhor resultado. O maior resultado encontrado é o melhor e o resultado com o menor valor o pior (item 13 da Figura 17). Gerando o ranque: 1º A4; 2º A3; 3º A2 e 4º A1.

Figura 16 Matriz de cálculo das distâncias com ranking final

	S+	S-	P
A1	3,765	0	0
A2	2,728	1,12	0,291
A3	0,726	3,056	0,808
A4	0	3,765	1

Fonte: próprio autor.

Figura 17 Aplicação do método TOPSIS



Fonte: próprio autor.

2.8 MÉTODO FUZZY TOPSIS

O método FUZZY TOPSIS vem sendo utilizado tanto academicamente, quanto mercadologicamente, como poderosa ferramenta de análise de multicritério para tomada de decisão, com maior precisão para um conjunto de dados nebulosos (KORE; RAVI; PATIL, 2017).

Chen, em 2000, propõe uma técnica multicritério que estende o procedimento adotado no método TOPSIS ao conjunto fuzzy, um método que mede a distância entre dois números Fuzzy triangulares. Método com também reduzido esforço computacional, com vantagens sobre o TOPSIS por não apresentar inversões de ranking quando da inserção ou exclusão de alternativas, além de se mostrar adequado no trato de informações qualitativas e imprecisas (MADI; GARIBALDI; WAGNER, 2015).

Neste método, tanto os valores atribuídos às alternativas, quanto aos pesos aplicados sobre os critérios são constituídos a partir de variáveis linguísticas avaliadas pelos tomadores de decisão (CALACHE *et al.*, 2019).

Após a atribuição de valores numéricos correspondentes aos julgamentos com base em uma classificação linguística, defusificação, gera-se os valores limites abaixo e acima do ponto de referência, resultando no conjunto fuzzy. A ordem de classificação dos resultados sobre as alternativas, considera simultaneamente a proximidade relativa de cada uma destas alternativas com a menor distância da solução ideal positiva (PIS) e a maior distância da solução ideal negativa (NIS) (TAYLAN; ALIDRISI; KABLI, 2014).

O Quadro 3, elaborado por Kahraman (2008), mostra variações aplicação da lógica fuzzy sobre a metodologia fuzzy topsis.

A lógica fuzzy difere do sistema lógico tradicional na medida em que a precisão corresponde a um limite de raciocínio aproximado, composto por um conjunto de relações nebulosas. Assim, enquanto nos sistemas tradicionais o valor verdade só assume dois valores, sim ou não, 1 ou 0, no sistema fuzzy o valor verdade pode ser um elemento de um conjunto finito em um intervalo (GOMIDE *et al.*, 1995).

Quadro 3 Comparação de métodos FUZZY TOPSIS

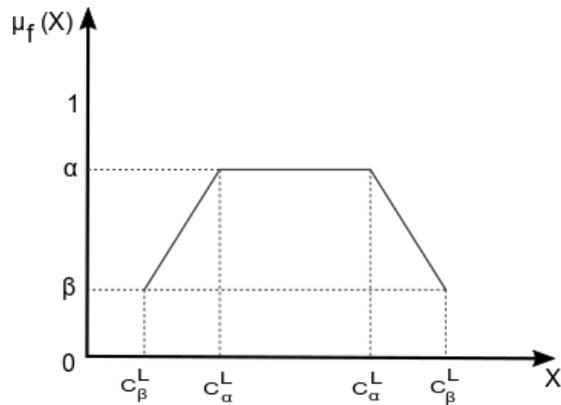
Fonte	Pesos de Atributos	Tipo de números difusos	Método de classificação	Método de Normalização
Chen and Hwang	números Fuzzys	Trapezoidal	Lee and Li's (1988) generalized mean method	Linear Normalization
Liang (1999)	números Fuzzys	Trapezoidal	Chen's (1985) ranking with maximizing set and minimizing set	Manhattan distance
Chen (2000)	números Fuzzys	Triangular	Chen (2000) proposes vertex method	Linear Normalization Modified
Chu (2002)	números Fuzzys	Triangular	Liou and Wang's (1992) ranking method of total integral value with $\alpha=1/2$	Manhattan distance
Tsaur Chang and Yen (2002)	Crisp Values	Triangular	Zhao and Govind's (1991) center of area method	Vector Normalization
Zhang and Lu (2003)	Crisp Values	Triangular	Chen's (2000) vertex method	Manhattan distance
Chu and Lin (2003)	números Fuzzys	Triangular	Kaufman and Gupta's (1988) mean of the removals method	Linear Normalization
Cha and Yung (2003)	Crisp Values	Triangular	Cha and Yung (2003) propose a fuzzy distance operator	Linear Normalization
Yang and Hung (2005)	números Fuzzys	Triangular	Chen's (2000) vertex method	Normalized fuzzy linguistic
Wang and Elhag (2006)	números Fuzzys	Triangular	Chen's (2000) vertex method	Linear Normalization
Jahanshahloo Hosseinzadeh and Izadikhah (2006)	Crisp Values	Interval data	Jahanshahloo, Hosseinzadeh e Izadikhah (2006) propose a new normlization & ranking method	

Fonte: Kahraman (2008).

Os pesos de atributos dizem respeito ao tipo de dado levantado, sendo valores linguísticos para números Fuzzy e valores numéricos para Crisp, observando que os números Fuzzy são convertidos para valores numéricos (defuzzificação) de acordo com uma tabela que deve retratar a natureza do problema observado. Os valores crisp são valores numéricos e podem representar valores de entrada de dados sobre números de diferentes grandezas, ou seja, poderiam representar unidades de medida de moeda, peso, comprimento, volume, potência ou outras unidades. Em ambos os casos os números fuzzy são compostos por um valor abaixo (*less*), um valor de referência e um valor acima (*up*).

Quanto ao tipo de números Fuzzy, o valor de referência está relacionado a função de pertinência, pode ser um intervalo, quando na forma trapezoidal, ou um valor na forma triangular, ou ainda um conjunto difuso normalizado convexo.

Figura 18 Número Fuzzy Trapezoidal

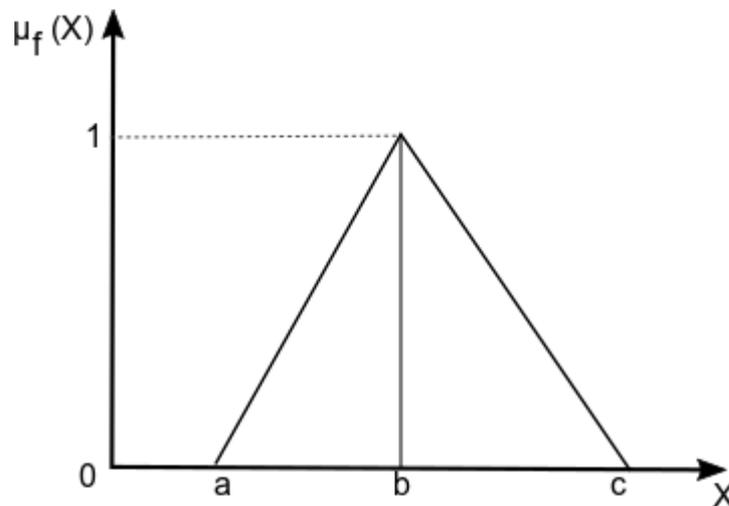


Fonte: Nádăban, Dzitac e Dzitac (2016).

Para a forma trapezoidal, os números fuzzy podem ser denotados por:

$$\mu_e(x) = \begin{cases} 0 & x < C_\beta^L \\ \frac{\alpha - \beta}{C_\alpha^L - C_\beta^L} (x - C_\beta^L) + \beta & C_\beta^L \leq x < C_\alpha^L \\ \alpha & C_\alpha^L \leq x \leq C_\beta^R \\ \frac{\alpha - \beta}{C_\alpha^R - C_\beta^R} (x - C_\beta^R) + \beta & C_\alpha^R < x \leq C_\beta^R \\ 0 & C_\beta^R < x \end{cases}$$

Figura 19 Número Fuzzy Triangular



Fonte: Kahraman (2008).

Na forma triangular o $f(x) = (a, b, c)$, seus valores de pertinência podem ser denotados pela fórmula:

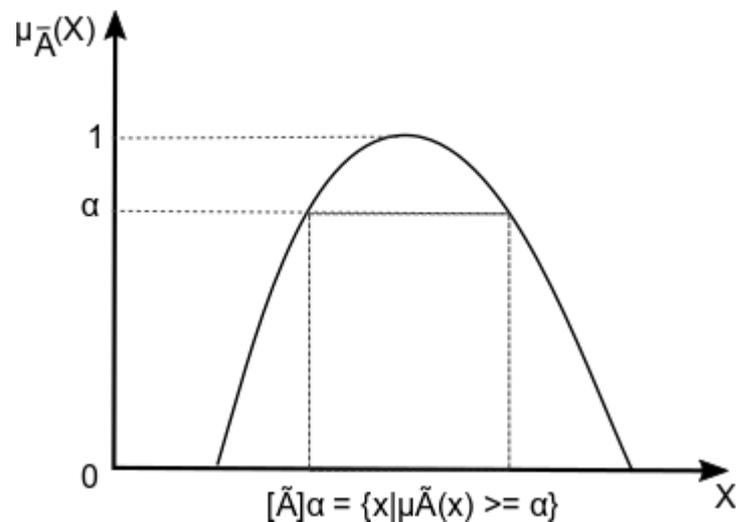
$$\mu_f(x) = \begin{cases} \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & \text{outros} \end{cases}$$

Fonte: Nădăban, Dzitac e Dzitac (2016).

O número fuzzy por intervalo de data, ou conjunto difuso normalizado convexo, está relacionado ao contínuo dentro de uma função. Um conjunto difuso \tilde{A} em X é convexo se, e somente se, para cada par de pontos x_1 e x_2 em X , a função de associação de \tilde{A} satisfizer a desigualdade

$$\mu_{\tilde{A}}(\delta x^1 + (1 - \delta)x^2) \geq \min(\mu_{\tilde{A}}(x^1), \mu_{\tilde{A}}(x^2)) \text{ onde } \delta \in [0,1].$$

Figura 20 Número Fuzzy Convexo



Fonte: Kahraman (2008).

Um número difuso \tilde{A} é um conjunto difuso normalizado convexo \tilde{A} da linha real R com função de associação contínua

Normalizar ou tornar padrão é criar uma identidade proporcional entre os elementos para uma mesma escala. A normalização para probabilidade total igual a 1 atende as normalizações lineares, onde os cálculos ficam limitados a divisão proporcional. Nas normalizações geométricas, logarítmica, por vértice e/ outras, encontra-se o padrão, mas, os cálculos são realizados por outras operações matemáticas além do somatório que resultam em uma probabilidade total diferente de 1, e no caso da lógica fuzzy há que se levar em conta o comportamento dos valores

abaixo e acima do valor de referência invertidos entre critérios do tipo benefício e do tipo custo.

Os métodos de classificação estão atrelados ao método de normalização: método da média generalizada; classificação com conjunto de maximização e conjunto de minimização; método de vértice; método de classificação do valor integral total com $\alpha=1/2$; método do centro da área; média do método de remoções; um operador de distância difusa.

Os métodos podem variar de acordo com a natureza dos dados, do problema proposto e da solução desejada, no anexo 1 são apresentadas algumas fórmulas utilizadas na aplicação do método FUZZY TOPSIS.

Passos para aplicar a metodologia FUZZY TOPSIS:

1° passo: Construção da Matriz de decisão (D), critérios, alternativas e julgamentos (item 1 da Figura 34).

Tabela 3 Matriz com valores linguísticos

Valores de escala para julgamentos			
Expressão literal	l	f	u
Muito pouco significativa	1	1	3
Pouco significativa	1	3	5
Média Significância	3	5	7
Muito Significante	5	7	9
Estrema significância	7	9	9

(l) *less* / baixo – (f) valor de referência – (u) *up* /acima

Fonte: próprio autor.

2° passo: Converter valores literais em valores fuzzy (item 2 da Figura 34), compostos pelo valor abaixo, valor de referência e valor acima (item 3 da Figura 34);

De acordo com uma tabela (item 3 da Figura 34), que corresponda à pesquisa em análise, deve-se defuzzificar, converter as avaliações em valores quantificáveis (item 2 da Figura 34), em valores numéricos de julgamento neste exemplo (Figura 22), como sugerido abaixo no intervalo entre 1 a 9, inclusive com valores intermediários, conforme exemplo:

Figura 21 Matriz de respostas de decisores

Decisor 1			
Alternativa	preço	Câmera em MP	Tempo de Bateria
Smartphone 1	Pouco significativa	Estrema significância	Pouco significativa
Smartphone 2	Média Significância	Média Significância	Estrema significância
Smartphone 3	Média Significância	Muito Significante	Muito Significante
Smartphone 4	Pouco significativa	Média Significância	Pouco significativa
Decisor 2			
Alternativa	preço	Câmera em MP	Tempo de Bateria
Smartphone 1	Média Significância	Estrema significância	Pouco significativa
Smartphone 2	Muito Significante	Média Significância	Muito Significante
Smartphone 3	Média Significância	Muito Significante	Muito Significante
Smartphone 4	Muito pouco significativa	Estrema significância	Média Significância
Decisor 3			
Alternativa	preço	Câmera em MP	Tempo de Bateria
Smartphone 1	Pouco significativa	Muito Significante	Média Significância
Smartphone 2	Média Significância	Média Significância	Estrema significância
Smartphone 3	Pouco significativa	Média Significância	Muito Significante
Smartphone 4	Média Significância	Muito Significante	Muito pouco significativa

Fonte: próprio autor.

Por tratarem de soluções próximas da percepção do mundo real, incluindo assim a incerteza, tanto a teoria dos conjuntos *fuzzy*, quanto a aplicação da lógica nebulosa, representam um enorme avanço científico para sistemas de diversos tipos, sobre/ outros métodos (OLIVEIRA *et al.*, 2014). Utiliza-se a linguagem humana natural para expressar suas percepções sobre um elemento, objeto de acordo com um prisma, um critério e depois convertesse-as em valores numéricos para cálculo.

Figura 22 Matrizes fuzzificadas

Decisor 1									
Alternativa	preço			Câmera em MP			Tempo de Bateria		
	l	f	u	l	f	U	l	f	u
Smartphone 1	2	3	4	7	9	9	2	3	4
Smartphone 2	4	5	6	4	5	6	7	9	9
Smartphone 3	4	5	6	6	7	8	6	7	8
Smartphone 4	2	3	4	4	5	6	2	3	4

Decisor 2									
Alternativa	preço			Câmera em MP			Tempo de Bateria		
	Smartphone 1	4	5	6	7	9	9	2	3
Smartphone 2	6	7	8	4	5	6	6	7	8
Smartphone 3	4	5	6	6	7	8	6	7	8
Smartphone 4	1	1	3	7	9	9	4	5	6

Decisor 3									
Alternativa	preço			Câmera em MP			Tempo de Bateria		
	Smartphone 1	2	3	4	6	7	8	4	5
Smartphone 2	4	5	6	4	5	6	7	9	9
Smartphone 3	2	3	4	4	5	6	6	7	8
Smartphone 4	4	5	6	6	7	8	1	1	3

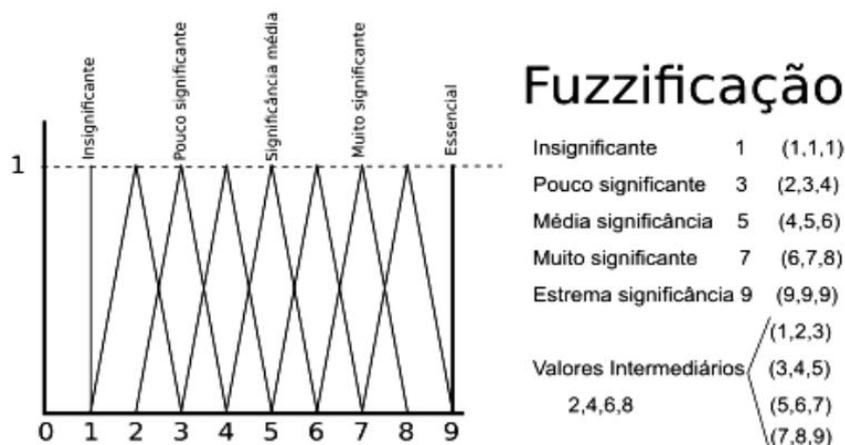
Fonte: próprio autor.

Os valores sugeridos poderiam ter grandezas maiores ou menores, respeitando a proporcionalidade das avaliações. Todos os valores foram definidos os limites inferiores (l), sempre à esquerda do valor de referência e superiores (u), sempre à direita, conforme o gráfico de pertinência sugerido no exemplo de fuzzificação (Tabela 3 e Figura 22), os valores utilizados no exemplo, fazem analogia ao método AHP adaptado a este método, entretanto, no método AHP o 1 tem valor de equivalência, neste método o grau de importância é crescente, onde o 1 tem o menor valor de importância e o 9 o maior (KANNAN *et al.*, 2013).

No método Analytic Hierarchy Process (AHP) criado pelo professor Thomas Saaty em 1970, fora utilizado um intervalo entre 1 e 9 para atribuição de valores dentro de uma escala que reflete a importância de uma alternativa a luz de um critério, onde os valores têm o seguinte significado: 1 equivalência, 3 leve importância, 5 grande importância, 7 muito grande importância, 9 absoluta importância; os valores 2, 4, 6 e 8 representam valores intermediários utilizados para ajustes, conforme 9 (na figura 7). Analogamente pode-se gerar uma tabela onde a atribuição de importância deve ser

proporcionalmente crescente, como notas de valor, assim teremos: 1 Insignificante, 3 Pouco significativa, 5 Média significância, 7 Muito significativa e 9 Absolutamente significativa. Esta classificação pode variar de acordo com a necessidade e bom senso do decisor, significando dizer que a escala e grandeza destes poderia variar, conforme figuras 23 e 24.

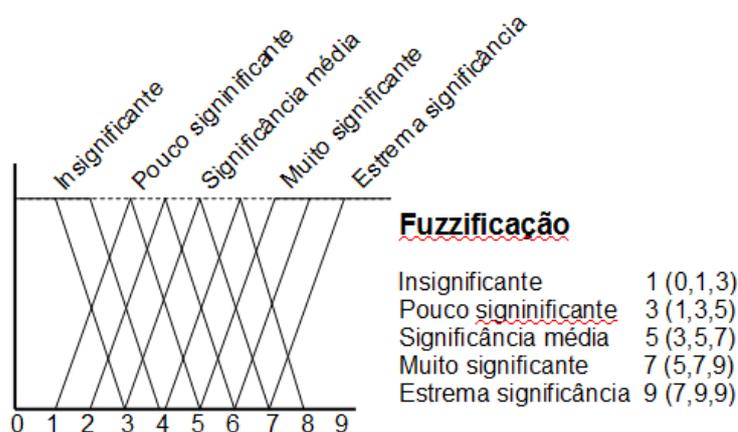
Figura 23 Sugestão de valores equiparados aos usados no método AHP



Fonte: Adaptado de LIMA JUNIOR e CARPINETTI (2014).

Neste exemplo, a fuzzificação está baseada no senso de proporcionalidade do decisor, ou do conjunto de especialistas envolvidos. Pode-se e deve-se levantar os intervalos inferiores e superiores a partir de amostragens observáveis, assim, em um exemplo para determinar a estatura baixa, média ou alta de pessoas, pode-se gerar uma coleta de opiniões de um conjunto de pessoas e a partir desse conjunto de dados encontrar os limites inferiores e superiores reconhecidos pela amostragem.

Figura 24 Valores Fuzzy sugeridos com intervalo de 2 abaixo e acima



Fonte: próprio autor.

2.1º passo: Mesclar Matrizes decisoras em uma única Matriz (itens 4 e 5 da Figura 34).

A mesclagem das Matrizes (item 4 da Figura 34) deve ter seus valores calculados diferentemente para cada posição do valor fuzzy (item 5 da Figura 34). O valor do limite inferior para cada alternativa em cada critério deverá ser o menor encontrado dentre os julgamentos dos decisores, bem com o limite superior deverá ser o maior, o valor de referência será uma média de todos os julgamentos, ou o somatório dos julgamentos dividido pela quantidade de julgadores.

Figura 25 Matriz combinada

Alternativa	preço			Câmera em MP			Tempo de Bateria		
Smartphone 1	2	3,67	6	6	8,33	9	2	3,67	6
Smartphone 2	4	5,67	8	4	5	6	6	8,33	9
Smartphone 3	2	4,33	6	4	6,33	8	6	7	8
Smartphone 4	1	3	6	4	7	9	1	3	6

Fonte: próprio autor.

3º passo: Normalizar os dados da Matriz fuzzy (item 6 da Figura 34);

3.1º passo: Para os critérios do tipo Benefício, divide-se o valor do julgamento pelo maior valor da alternativa (item 7 da Figura 34);

3.2º passo: Para os critérios do tipo Custo, divide-se o menor valor de julgamento pelos valores inversos, ou seja, 'l' (less) pelo 'u' (up) e 'u' pelo 'l' (item 7 da Figura 34);

Figura 26 Matriz com valores normalizados

Alternativa	preço			Câmera em MP			Tempo de Bateria		
	l	f	u	l	f	u	l	f	u
Smartphone 1	0,17	0,27	0,5	0,67	0,93	1	0,22	0,41	0,67
Smartphone 2	0,13	0,18	0,25	0,44	0,56	0,67	0,67	0,93	1
Smartphone 3	0,17	0,23	0,5	0,44	0,7	0,89	0,67	0,78	0,89
Smartphone 4	0,17	0,33	1	0,44	0,78	1	0,11	0,33	0,67

Fonte: próprio autor.

4º passo: Gerar uma nova Matriz a partir da aplicação dos pesos de cada critério, ou seja, da multiplicação dos valores normalizados pelo peso respectivo (itens 8 e 9 da Figura 34);

Figura 27 Matriz de pesos

	preço			Câmera em MP			Tempo de Bateria		
Pesos	7	9	9	3	5	7	5	7	9

Fonte: próprio autor.

Figura 28 Matriz com valores pesificados

Alternativa	preço			Câmera em MP			Tempo de Bateria		
Smartphone 1	1,17	2,46	4,5	2	4,63	7	1,11	2,85	6
Smartphone 2	0,88	1,59	2,25	1,33	2,78	4,67	3,33	6,48	9
Smartphone 3	1,17	2,08	4,5	1,33	3,52	6,22	3,33	5,44	8
Smartphone 4	1,17	3	9	1,33	3,89	7	0,56	2,33	6

Fonte: próprio autor.

5º passo: Montar o vetor de solução ideal positiva (PIS) A+ e criar o vetor de solução ideal negativa (NIS) A- (itens 10 e 11 da Figura 34);

Figura 29 : Matrizes PIS e NIS

(PIS) A+	1,2	3	9	2	4,6	7	3,3	6,5	9
(NIS) A-	0,9	1,6	2,3	1,3	2,8	4,7	0,6	2,3	6

Fonte: próprio autor.

6º passo: Calcular as distâncias (item 13 da Figura 34) entre os dados das alternativas e do PIS S+ e as distâncias entre os dados das alternativas e do NIS S-, a partir dos vetores de soluções ideais positivas e negativas (itens 12 e 14 da Figura 34);

Figura 30 Matriz resultante sobre a Matriz PIS

d+ Alternativa	preço	Câmera em MP	Tempo de Bateria
Smartphone 1	1,297	0	1,718
Smartphone 2	1,679	1,272	0
Smartphone 3	1,345	0,923	0,824
Smartphone 4	0	0,685	1,819

Fonte: próprio autor.

Figura 31 Matriz resultante sobre a Matriz NIS

d- Alternativa	preço	Câmera em MP	Tempo de Bateria
Smartphone 1	1,066	1,272	0,598
Smartphone 2	0	0	1,819
Smartphone 3	1,005	0,875	1,622
Smartphone 4	1,679	1,072	0

Fonte: próprio autor.

7º passo: Equacionar a melhor proximidade relativa;

7.1º passo: Totalizar as distâncias encontradas da alternativa em cada critério (item 16 da Figura 34);

Nesta etapa todos os valores relativos da alternativa a luz dos critérios já foram calculados, mas, ainda não somados para gerar uma pontuação. Faz-se o somatório (item 16 da Figura 34) para em seguida calcular a distância de cada alternativa com maior S+ ou menor S- proximidade como as soluções ideais positivas ou negativas.

7.2º Calcular a distância final de cada alternativa com maior proximidade da PIS e maior afastamento da NIS (item 17 da Figura 34);

Para calcular o melhor ponto em proximidade com solução ideal positiva e mais afastado da solução ideal positiva, divide-se o d- (somatório das distâncias negativas) pelo somatório (d+ + d-) das distâncias positivas com as distâncias negativas).

Figura 32 Matriz com cálculo de coeficiente de proximidade

Alternativa	soma d+	soma d-	$d-/ (d+ + d-)$
Smartphone 1	3,015	2,936	0,493
Smartphone 2	2,95	1,819	0,381
Smartphone 3	3,091	3,502	0,531
Smartphone 4	2,504	2,75	0,523

Fonte: próprio autor.

8º passo: Ranquear as respostas para encontrar o melhor resultado.

O resultado é encontrado a partir da ordenação dos valores resultantes da equação (item 17 da Figura 34).

Figura 33 Matriz final com valores ordenados

Alternativa	$d-/ (d+ + d-)$	ordem
Smartphone 1	0,493	3
Smartphone 2	0,381	4
Smartphone 3	0,531	1
Smartphone 4	0,523	2

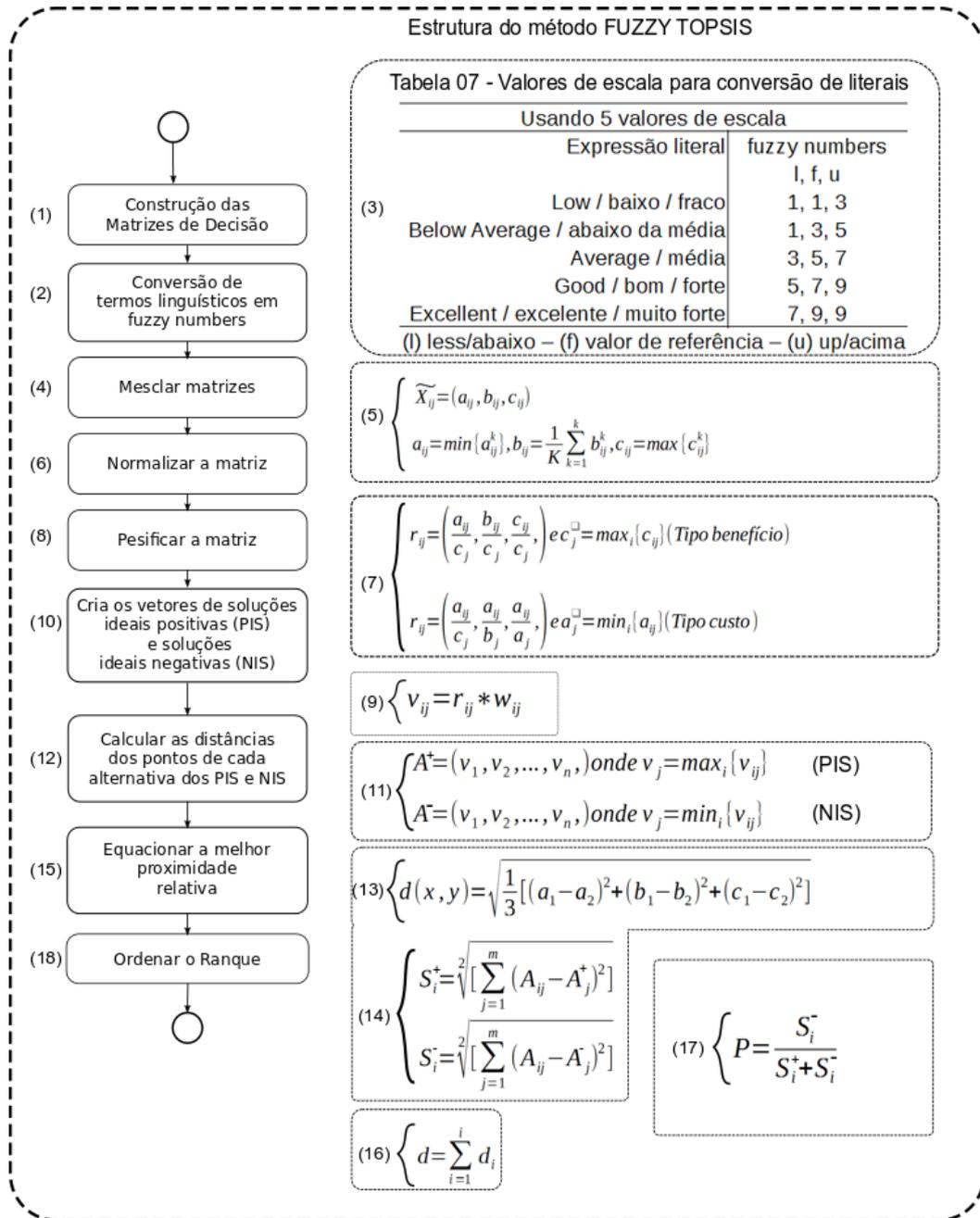
Fonte: próprio autor.

Segue um exemplo que permite visualizar tanto a fuzzificação, conversão em conjuntos fuzzy, a aplicação sobre critérios de natureza diferente, custo x benefício, a aplicação de pesos, a adaptação à lógica e os passos para o ranking utilizando o método TOPSIS.

Supondo a necessidade de comprar um celular intermediário, com preço baixo, ótima câmera e com bateria de boa duração. Supondo ainda que serão utilizadas expressões linguísticas: Insignificante, Muito pouco significativa, Pouco significativa, Média significância, Muito significativa e Extrema significância.

Para a melhor compreensão deste exemplo, foram levantados dados sobre os quesitos: preço, Câmera e bateria, com suas respectivas grandezas: reais(moeda), tamanho(pixels), e tempo(minutos) (TUDOCELULAR TECNOLOGIA LTDA, 2018).

Figura 34 Estrutura do método FUZZY TOPSIS



Fonte: próprio autor.

A ideia de se utilizar lógica fuzzy implica na percepção a partir de variáveis linguísticas (item 3 da Figura 34), entretanto, há duas possibilidades aqui de aplicar essa metodologia, uma é a conversão dos valores linguisticamente dentro de uma tabela para fuzzificação, outra é a utilização direta dos valores crisp, com algumas diferenças quanto a formação do conjunto fuzzy, como demonstrado no próximo no item 2.5.3.1 MÉTODO FUZZY TOPSIS COM VALORES CRISP. Assim como no trato

de informações quando com múltiplos decisores, também exemplificado abaixo no MÉTODO FUZZY TOPSIS COM MÚLTIPLOS DECISORES, ainda com algumas modificações de procedimento.

1º passo: Construção da Matriz de decisão (D)

Procede-se os julgamentos dos decisores (Figura 35), montam-se as Matrizes de decisão com valores linguísticos que podem variar em quantidade e qualidade, de acordo com a natureza do problema, (itens 1 e 3 da Figura 34).

Tabela 4 Matriz com valores linguísticos

Valores de escala para julgamentos			
Expressão literal	l	f	u
Muito pouco significativa	1	1	3
Pouco significativa	1	3	5
Média Significância	3	5	7
Muito Significante	5	7	9
Estrema significância	7	9	9

(l) less / baixo – (f) valor de referência – (u) up /acima

Fonte: próprio autor.

2º passo: Converter valores literais em valores fuzzy

De acordo com uma tabela (item 3 da Figura 34), que correspondente à pesquisa em análise, deve-se defuzzificar, converter as avaliações em valores quantificáveis (item 2 da Figura 34), em valores numéricos de julgamento neste exemplo (Figura 36), como sugerido abaixo no intervalo entre 1 a 9, inclusive com valores intermediários, conforme exemplo:

Figura 35 Matriz de respostas de decisores

Decisor 1			
Alternativa	preço	Câmera em MP	Tempo de Bateria
Smartphone 1	Pouco significativa	Estrema significância	Pouco significativa
Smartphone 2	Média Significância	Média Significância	Estrema significância
Smartphone 3	Média Significância	Muito Significante	Muito Significante
Smartphone 4	Pouco significativa	Média Significância	Pouco significativa
Decisor 2			
Alternativa	preço	Câmera em MP	Tempo de Bateria
Smartphone 1	Média Significância	Estrema significância	Pouco significativa
Smartphone 2	Muito Significante	Média Significância	Muito Significante
Smartphone 3	Média Significância	Muito Significante	Muito Significante
Smartphone 4	Muito pouco significativa	Estrema significância	Média Significância
Decisor 3			
Alternativa	preço	Câmera em MP	Tempo de Bateria
Smartphone 1	Pouco significativa	Muito Significante	Média Significância
Smartphone 2	Média Significância	Média Significância	Estrema significância
Smartphone 3	Pouco significativa	Média Significância	Muito Significante
Smartphone 4	Média Significância	Muito Significante	Muito pouco significativa

Fonte: próprio autor.

Por tratarem de soluções próximas da percepção do mundo real, incluindo assim a incerteza, tanto a teoria dos conjuntos fuzzy, quanto a aplicação da lógica nebulosa, representam um enorme avanço científico para sistemas de diversos tipos, sobre/ outros métodos (OLIVEIRA *et al.*, 2014). Utiliza-se a linguagem humana natural para expressar suas percepções sobre um elemento, objeto de acordo com um prisma, um critério e depois convertesse-as em valores numéricos para cálculo.

Os valores sugeridos poderiam ter grandezas maiores ou menores, respeitando a proporcionalidade das avaliações. Todos os valores foram definidos os limites inferiores (l), sempre à esquerda do valor de referência e superiores (u), sempre à direita, conforme o gráfico de pertinência sugerido no exemplo de fuzzificação (Tabela 4 e Figura 36), os valores utilizados no exemplo, fazem analogia ao método AHP adaptado a este método, entretanto, no método AHP o 1 tem valor de equivalência, neste método o grau de importância é crescente, onde o 1 tem o menor valor de importância e o 9 o maior (KANNAN *et al.*, 2013).

Figura 36 Matrizes fuzzificadas

Decisor 1									
Alternativa	preço			Câmera em MP			Tempo de Bateria		
	l	f	u	l	f	u	l	f	u
Smartphone 1	2	3	4	7	9	9	2	3	4
Smartphone 2	4	5	6	4	5	6	7	9	9
Smartphone 3	4	5	6	6	7	8	6	7	8
Smartphone 4	2	3	4	4	5	6	2	3	4

Decisor 2									
Alternativa	preço			Câmera em MP			Tempo de Bateria		
	l	f	u	l	f	u	l	f	u
Smartphone 1	4	5	6	7	9	9	2	3	4
Smartphone 2	6	7	8	4	5	6	6	7	8
Smartphone 3	4	5	6	6	7	8	6	7	8
Smartphone 4	1	1	3	7	9	9	4	5	6

Decisor 3									
Alternativa	preço			Câmera em MP			Tempo de Bateria		
	l	f	u	l	f	u	l	f	u
Smartphone 1	2	3	4	6	7	8	4	5	6
Smartphone 2	4	5	6	4	5	6	7	9	9
Smartphone 3	2	3	4	4	5	6	6	7	8
Smartphone 4	4	5	6	6	7	8	1	1	3

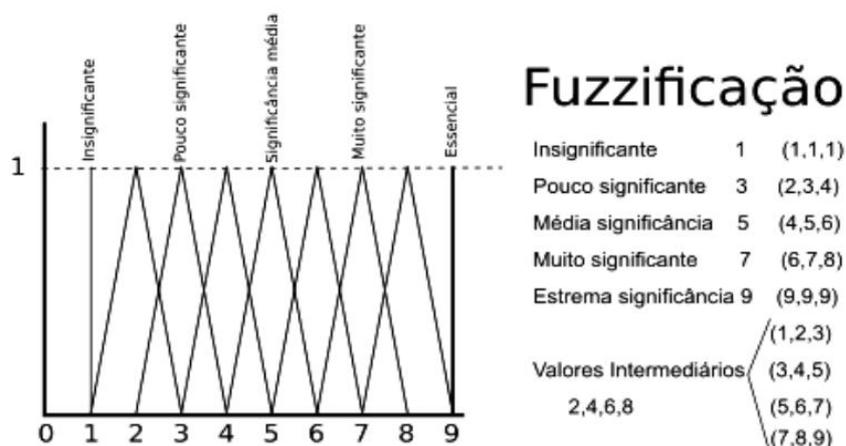
Fonte: próprio autor.

Os valores sugeridos poderiam ter grandezas maiores ou menores, respeitando a proporcionalidade das avaliações. Todos os valores foram definidos os limites inferiores (l), sempre à esquerda do valor de referência e superiores (u), sempre à direita, conforme o gráfico de pertinência sugerido no exemplo de fuzzificação (Tabela 4 e Figura 36), os valores utilizados no exemplo, fazem analogia ao método AHP adaptado a este método, entretanto, no método AHP o 1 tem valor de equivalência, neste método o grau de importância é crescente, onde o 1 tem o menor valor de importância e o 9 o maior (KANNAN *et al.*, 2013).

No método Analytic Hierarchy Process (AHP) criado pelo professor Thomas Saaty em 1970, fora utilizado um intervalo entre 1 e 9 para atribuição de valores dentro de uma escala que reflete a importância de uma alternativa a luz de um critério, onde os valores têm o seguinte significado: 1 equivalência, 3 leve importância, 5 grande importância, 7 muito grande importância, 9 absoluta importância; os valores 2, 4, 6 e 8 representam valores intermediários utilizados para ajustes, conforme na figura 37.

Analogamente pode-se gerar uma tabela onde a atribuição de importância deve ser proporcionalmente crescente, como notas de valor, assim teremos: 1 Insignificante, 3 Pouco significativa, 5 Média significância, 7 Muito significativa e 9 Absolutamente significativa. Esta classificação pode variar de acordo com a necessidade e bom senso do decisor, significando dizer que a escala e grandeza destes poderia variar, conforme figuras 23 e 24.

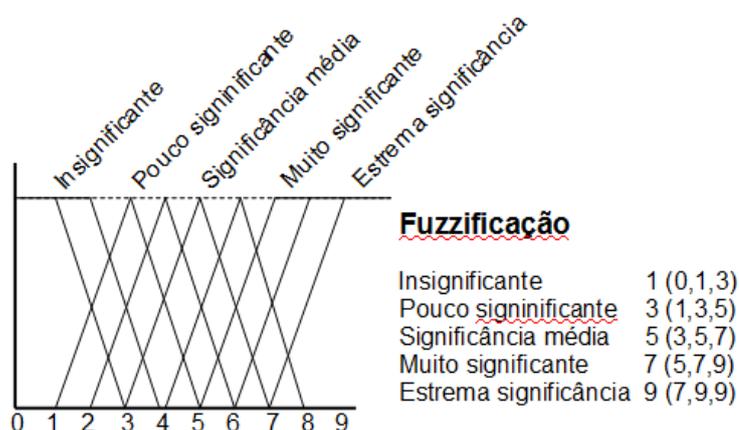
Figura 37 : Sugestão de valores equiparados aos usados no método AHP



Fonte: Kannan *et al.* (2013).

Neste exemplo, a fuzzificação está baseada no senso de proporcionalidade do decisor, ou do conjunto de especialistas envolvidos. Pode-se e deve-se levantar os intervalos inferiores e superiores a partir de amostragens observáveis, assim, em um exemplo para determinar a estatura baixa, média ou alta de pessoas, pode-se gerar uma coleta de opiniões de um conjunto de pessoas e a partir desse conjunto de dados encontrar os limites inferiores e superiores reconhecidos pela amostragem.

Figura 38 Valores Fuzzy sugeridos com intervalo de 2 abaixo e acima



Fonte: próprio autor.

2.1 – passo: Mesclar Matrizes decisoras em uma única Matriz

A mesclagem das Matrizes (item 4 da Figura 34) deve ter seus valores calculados diferentemente para cada posição do valor fuzzy (item 5 da Figura 34). O valor do limite inferior para cada alternativa em cada critério deverá ser o menor encontrado dentre os julgamentos dos decisores, bem com o limite superior deverá ser o maior, o valor de referência será uma média de todos os julgamentos, ou o somatório dos julgamentos dividido pela quantidade de julgadores.

Figura 39 Matriz combinada

Alternativa	preço			Câmera em MP			Tempo de Bateria		
Smartphone 1	2	3,67	6	6	8,33	9	2	3,67	6
Smartphone 2	4	5,67	8	4	5	6	6	8,33	9
Smartphone 3	2	4,33	6	4	6,33	8	6	7	8
Smartphone 4	1	3	6	4	7	9	1	3	6

Fonte: próprio autor.

3º passo: A normalização ou uniformização dos dados.

É natural imaginar que algo que seja benéfico deva ter os maiores valores possível, enquanto algo que tenha custos deva ter os menores. Nesta modelagem do método FUZZY TOPSIS em questão, em meio à normalização (item 6 da Figura 34), faz-se o cálculo diferenciado sobre os dados cujo critério seja do tipo benefício ou custo (item 7 da Figura 34).

Para o critério do tipo benefício divide-se o valor do julgamento pelo maior valor da alternativa (item 7 da Figura 34) para todas os valores, tanto o valor abaixo, quanto o valor de referência, quanto o valor acima deve ser dividido pelo maior valor da Matriz combinada (Matriz 10); para o critério do tipo custo divide-se o menor valor de julgamento pelos valores inversamente proporcionais à grandeza, ou seja, 'l' (less) pelo 'u' (up) e 'u' pelo 'l' (item 7 da Figura 34);

Figura 40 Matriz com valores normalizados

Alternativa	preço			Câmera em MP			Tempo de Bateria		
	l	f	u	l	f	u	l	f	u
	Smartphone 1	0,17	0,27	0,5	0,67	0,93	1	0,22	0,41
Smartphone 2	0,13	0,18	0,25	0,44	0,56	0,67	0,67	0,93	1
Smartphone 3	0,17	0,23	0,5	0,44	0,7	0,89	0,67	0,78	0,9
Smartphone 4	0,17	0,33	1	0,44	0,78	1	0,11	0,33	0,7

Fonte: próprio autor.

4º passo: A pesificação da Matriz normalizada

Os critérios podem e normalmente apresentam importâncias diferentes entre si, portanto, não devem ter o mesmo peso no julgamento. Nesta etapa os pesos são multiplicados pelos respectivos valores nos critérios (itens 8 e 9 da Figura 34) sobre a Matriz normalizada e uma nova Matriz é gerada.

Figura 41 Matriz de pesos

Pesos	[7	9	9	3	5	7	5	7	9]
-------	---	---	---	---	---	---	---	---	---	---	---

Fonte: próprio autor.

Figura 42 Matriz com valores pesificados

Alternativa	preço			Câmera em MP			Tempo de Bateria		
Smartphone 1	1,17	2,46	4,5	2	4,63	7	1,11	2,85	6
Smartphone 2	0,88	1,59	2,25	1,33	2,78	4,67	3,33	6,48	9
Smartphone 3	1,17	2,08	4,5	1,33	3,52	6,22	3,33	5,44	8
Smartphone 4	1,17	3	9	1,33	3,89	7	0,56	2,33	6

Fonte: próprio autor.

5° passo: Gerar o vetor de solução ideal positiva (PIS) A+ e o vetor de solução ideal negativa (NIS) A- (itens 10 e 11 da Figura 34);

A técnica para medir o desempenho baseada em comparativo com a solução ideal foi proposta por Hwang e Yoon sobre o método TOPSIS e adaptada para este método. O maior valor, relativo ao critério, de cada um dos pontos da Matriz pesificada é colocado no conjunto PIS, assim como, o menor valor no conjunto NIS. O objetivo é minimizar os critérios do tipo custo e maximizar os do tipo benefício.

Figura 43 Matrizes PIS e NIS

(PIS) A+	1,2	3	9	2	4,6	7	3,3	6,5	9
(NIS) A-	0,9	1,6	2,3	1,3	2,8	4,7	0,6	2,3	6

Fonte: próprio autor

6° passo: Calcular as distâncias, (item 13 da Figura 34), entre os dados das alternativas e do PIS S+ e as distâncias entre os dados das alternativas e do NIS S-, a partir dos vetores de soluções ideais positivas e negativas (itens 12 e 14 da Figura 34).

Nesta etapa, simultaneamente, a planilha se divide em duas partes, uma sobre o vetor (PIS) A+ => d+ (Figura 30 e/ outra sobre o vetor (NIS) A- => d- (Figura 31) formando duas Matrizes. Há uma redução das colunas antes fuzzificadas, são proporcionalmente aglutinados em um valor para representar o ponto avaliado para a alternativa devidamente normalizado e pesificado.

Figura 44 Matriz resultante sobre a Matriz PIS

d+	Alternativa	preço	Câmera em MP	Tempo de Bateria
	Smartphone 1	1,297	0	1,718
	Smartphone 2	1,679	1,272	0
	Smartphone 3	1,345	0,923	0,824
	Smartphone 4	0	0,685	1,819

Fonte: próprio autor

Figura 45 Matriz resultante sobre a Matriz NIS

d-	Alternativa	preço	Câmera em MP	Tempo de Bateria
	Smartphone 1	1,066	1,272	0,598
	Smartphone 2	0	0	1,819
	Smartphone 3	1,005	0,875	1,622
	Smartphone 4	1,679	1,072	0

Fonte: próprio autor

7º passo: Equacionar a melhor proximidade relativa;

7.1º passo: Totalizar as distâncias encontradas da alternativa em cada critério.

Nesta etapa todos os valores relativos da alternativa a luz dos critérios já foram calculados, mas, ainda não somados para gerar uma pontuação. Faz-se o somatório (item 16 da Figura 34) para em seguida calcular a distância de cada alternativa com maior S+ ou menor S- proximidade como as soluções ideais positivas ou negativas.

7.2º passo: Calcular a distância final de cada alternativa com maior proximidade da PIS e maior afastamento da NIS (item 17 da Figura 34);

Para calcular o melhor ponto em proximidade com solução ideal positiva e mais afastado da solução ideal positiva, divide-se o d- (somatório das distâncias negativas) pelo somatório (d+ + d-) das distâncias positivas com as distâncias negativas).

Figura 46 Matriz com cálculo de coeficiente de proximidade

Alternativa	soma d+	soma d-	$d-/ (d+ + d-)$
Smartphone 1	3,015	2,936	0,493
Smartphone 2	2,95	1,819	0,381
Smartphone 3	3,091	3,502	0,531
Smartphone 4	2,504	2,75	0,523

Fonte: próprio autor

8º passo: Ranquear as respostas para encontrar o melhor resultado.

O resultado é encontrado a partir da ordenação dos valores resultantes da equação (item 17 da Figura 34).

Figura 47 Matriz final com valores ordenados

Alternativa	$d-/ (d+ + d-)$	ordem
Smartphone 1	0,493	3
Smartphone 2	0,381	4
Smartphone 3	0,531	1
Smartphone 4	0,523	2

Fonte: próprio autor

Uma comparação entre os métodos TOPSIS e FUZZY-TOPSIS no apoio à tomada de decisão multicritério para seleção de fornecedores.

1. Quantidade de Julgamentos Requeridos: n representa o número de alternativas, m o número de critérios, cada alternativa é julgada por cada critérios resultando na multiplicação da quantidade destes mutuamente. A lógica fuzzy faz uso de uma função de pertinência triangular, sendo um valor de referência, tal qual no método TOPSIS, acrescida do valor abaixo (l) e acima (u), portanto, com dois valores a mais na coluna de critérios (m) para processar os julgamentos.

Quadro 4 Quadro comparativo dos métodos TOPSIS e FUZZY TOPSIS

Comparação entre os algoritmos dos métodos TOPSIS E FUZZY TOPSIS			
Aspectos Avaliados	TOPSIS	FUZZY TOPSIS	Conclusão
1. Quantidade de Julgamentos Requeridos	nm+m julgamentos	nm+n julgamentos (mais valores para termos linguísticos)	Diferente
2. Pontuações dos fornecedores e pesos	Expressa por valores numéricos crisp	Valores linguísticos fuzzy	Diferente
3. Desempenho global dos fornecedores (CC)	Feita em formato crisp entre [0,1]	Formato crisp entre [0,1]	Igual
4. Composição da Solução Ideal Positiva	Em cada critério, seleciona-se o melhor valor alcançado pelas alternativas	Definida como (1.0,1.0,1.0) para todos os critérios	Diferente
5. Composição da Solução Ideal Negativa	Em cada critério, seleciona-se o pior valor alcançado pelas alternativas	Definida como (0.0,0.0,0.0) para todos os critérios	Diferente
6. Cálculo das distâncias D ⁺ _i e D ⁻ _i	Método da distância euclidiana normalizada	Método vertex	Semelhante
7. Consistência dos resultados fornecidos	A inserção de novas alternativas pode causar inversão no ranking	A inserção de novas alternativas não alterou o ranking	Diferente

Fonte: Lima Junior e Carpinetti (2014).

2. Pontuações dos fornecedores e pesos: Embora se possa variar o método TOPSIS com o uso de variáveis literais, tradicionalmente os valores de julgamento utilizados são numéricos, enquanto a lógica fuzzy é baseada na utilização de variáveis linguísticas, entretanto, esta também admite uma variação do método FUZZY TOPSIS, com a utilização de dados do tipo crisp (item 1.3.10 do referencial teórico), mantida a coerência com a lógica fuzzy. A lógica fuzzy enquanto método difere de sua aplicação enquanto conceito.

3. Desempenho global dos fornecedores (CC): O coeficiente de aproximação (closeness coefficient, CCI) no método TOPSIS (item 12 da Figura 17) é idêntico ao método FUZZY TOPSIS (item 17 da Figura 34), correspondendo ao desempenho global de cada alternativa.

4. Composição da Solução Ideal Positiva: Quanto ao método TOPSIS, a melhor solução para um critério do tipo custo é inversamente proporcional a melhor solução para critério do tipo benefício (item 10 da Figura 17). Quanto ao método FUZZY TOPSIS, Chen (2000) define como um valor perfeito (PIS) = (1,1,1) e (NIS) = (0,0,0),

posteriormente Yuen (2014) define como um valor ideal um valor máximo na Matriz de decisão para (PIS) e um valor mínimo para (NIS), como o utilizado neste trabalho (item 11 da Figura 34).

Há variações sobre a aplicação do método que podem depender da normalização. Pode-se considerar a ideia original da lógica fuzzy onde os valores limites abaixo e acima estão limitados a 1 e 0 [1, 0], neste caso dependem de uma normalização linear que leve em consideração os máximos e mínimos, ou, os maiores e menores valores, de tal forma que nenhum valor normalizado ultrapasse o valor de 1, de/ outra forma, pode-se levar em consideração os valores máximos e mínimos e faz a diferenciação de critérios do tipos custo ou benefício na normalização, neste caso, a normalização é baseada em cálculos geométricos e os valores máximos podem exceder o valor de 1, por isso mesmo, os valores utilizados para PIS fazem referência aos maiores valores.

5. Composição da Solução Ideal Negativa: Da mesma forma que (item 4) apresenta as mesmas variações.

6. Cálculo das distâncias D^+ e D^- : O conceito de trabalhar como pontos em um plano cartesiano é o mesmo, entretanto, para o método FUZZY TOPSIS há três pontos por critério, fazendo com que estes sejam aglutinados no cálculo das distâncias a partir do conceito de vértice, ou, de três pontos.

7. Consistência dos resultados fornecidos: o autor deste Quadro relata resultados a partir de simulações computacionais, com inclusões e exclusões de alternativas aleatórias em ambos os modelos e concluiu que estes procedimentos afetaram os resultados sobre o método TOPSIS de forma a modificar o desempenho global das demais alternativas, causando inclusive, em alguns casos a inversão do ranque, problema este conhecido como ranking reversal, também relatados em métodos AHP e Fuzzy AHP. No método FUZZY TOPSIS o desempenho global e a classificação das alternativas permaneceram os mesmos após aplicar as mesmas simulações.

2.9 MÉTODO FUZZY TOPSIS COM VALORES CRISP

Na lógica fuzzy ou na utilização desta enquanto método, necessariamente trabalha-se com variáveis linguísticas, permitindo trabalhar sobre a incerteza do conhecimento, sobre a imprecisão dos dados (Tabela 1 do referencial teórico/lógica

fuzzy). Ocasionalmente são calculados sobre números absolutos em formato crisp, utilizando-se o conceito fuzzy (Figura 02), ou, valores abaixo e acima, considerando uma visão triangular dos dados (Figura 19), de tal forma que os valores próximos ao valor acima de um valor de referência, poderiam fazer parte de um conjunto de valores do valor inferior de/ outro conjunto de valor de referência e vice-versa.

Na utilização direta de valores crisp, a variação do método encontra-se na fase inicial. Para determinar os valores abaixo e acima do valor de referência para montar o conjunto fuzzy, define-se, respectivamente, como o menor valor sobre todos os valores encontrados para o valor abaixo de todas as alternativas e, o maior valor sobre todos os valores encontrados para o valor acima, em cada critério.

Sendo assim, no exemplo abaixo para os valores da coluna preço (Figura 48), o menor valor é o de 1500 e o maior é de 2150, sendo respectivamente passados para os valores l (*less*) e u (*up*) (Figura 49).

Figura 48 Matriz com valores crisp

Alternativa/Critério	preço	Câmera em MP	Tempo de Bateria
Celular 1	1800	48	990
Celular 2	2100	64	750
Celular 3	1950	12	1500
Celular 4	1500	64	1200
Celular 5	2000	48	1070
Celular 6	2150	12	690

Fonte: próprio autor.

Figura 49 Matriz com valores fuzzificados

Alternativa/Critério	preço			Câmera em MP			Tempo de Bateria		
Celular 1	1500	1800	2150	12	48	64	690	990	1500
Celular 2	1500	2100	2150	12	64	64	0	750	1500
Celular 3	1500	1950	2150	12	12	64	0	1500	1500
Celular 4	1500	1500	2150	12	64	64	0	1200	1500
Celular 5	1500	2000	2150	12	48	64	990	1070	1500
Celular 6	1500	2150	2150	12	12	64	750	690	1500

Fonte: próprio autor.

O Cálculo da normalização leva em consideração o tipo de critério, se de custo ou de benefício. Quando benefício, divide-se cada valor ('l', 'f' e 'u') pelo maior valor, por exemplo para o Smartphone 1, sobre o critério do tipo benefício: Câmera em MP: o valor l = $12/64 \Rightarrow 0,188$, o valor f = $48/64 \Rightarrow 0,75$ e o valor u = $64/64 \Rightarrow 1$. Sobre o critério do tipo custo: preço: para o valor l divide-se o menor valor dos três valores pelo u, assim l = $1500/2150 \Rightarrow 0,698$, para o valor f, divide-se o menor valor pelo valor de f = $1500/1800 \Rightarrow 0,833$ e o para o valor u, divide-se o menor valor pelo valor de l, u = $1500/1500$.

Figura 50 Matriz com valores normalizados

Alternativa/Critério	preço		Câmera em MP			Tempo de Bateria			
Celular 1	0,698	0,833	1	0,188	0,75	1	0,46	0,66	1
Celular 2	0,698	0,714	1	0,188	1	1	0,46	0,5	1
Celular 3	0,698	0,769	1	0,188	0,188	1	0,46	1	1
Celular 4	0,698	1	1	0,188	1	1	0,46	0,8	1
Celular 5	0,698	0,75	1	0,188	0,75	1	0,46	0,713	1
Celular 6	0,698	0,698	1	0,188	0,188	1	0,46	0,46	1

Fonte: próprio autor.

A continuidade na aplicação do método é a mesma, após a normalização (Figura 50) faz-se a pesificação (Figura 52), a definição das Matrizes soluções ideais positivas e negativas (Figura 53), o cálculo das distâncias dos pontos em relação a essas soluções ideais (Figuras 54 e 55), e finalmente o cálculo do coeficiente de aproximação (Figura 56) para permitir a ordenação.

Figura 51 Matriz de pesos

Pesos	4	5	6	2	3	4	1	2	3
-------	---	---	---	---	---	---	---	---	---

Fonte: próprio autor

Figura 52 Matriz pesificada

Alternativa/Critério	preço	Câmera em MP	Tempo de Bateria
Celular 1	2,79 4,17	6 0,38 2,25	4 0,46 1,32 3
Celular 2	2,79 3,57	6 0,38 3	4 0,46 1 3
Celular 3	2,79 3,85	6 0,38 0,56	4 0,46 2 3
Celular 4	2,79 5	6 0,38 3	4 0,46 1,6 3
Celular 5	2,79 3,75	6 0,38 2,25	4 0,46 1,43 3
Celular 6	2,79 3,49	6 0,38 0,56	4 0,46 0,92 3

Fonte: próprio autor

A Matriz de pesos da Figura 51 fora multiplicada pela Matriz com valores normalizados da Figura 50 resultando na Matriz pesificada da Figura 52.

Figura 53 Matrizes com soluções ideais positivas (PIS) e negativas (NIS)

A+	2,8 5 6 0,4 3 4 0,5 2 3
A-	2,8 3,5 6 0,4 0,6 4 0,5 0,9 3

Fonte: próprio autor.

A Matriz com soluções ideais positivas (PIS) mostra os maiores valores de referência relativo a cada critério e a Matriz com soluções ideais negativas (NIS) os menores valores relativo a cada critério.

Figura 54 Matriz com distâncias das alternativas para a solução ideal positiva

Alternativa/Critério	preço	Câmera em MP	Tempo de Bateria
Celular 1	0,48	0,43	0,39
Celular 2	0,83	0	0,58
Celular 3	0,67	1,41	0
Celular 4	0	0	0,23
Celular 5	0,72	0,43	0,33
Celular 6	0,87	1,41	0,62

Fonte: próprio autor.

A Matriz com distâncias das alternativas para PIS da Figura 54, bem como das alternativas para NIS da Figura 55, têm os valores fuzzy abaixo e acima recalculados para gerar um só valor de distância para cada alternativa.

Figura 55 Matriz com distâncias das alternativas para a solução ideal negativa

Alternativa/Critério	preço	Câmera em MP	Tempo de Bateria
Celular 1	0,392	0,974	0,231
Celular 2	0,048	1,407	0,046
Celular 3	0,207	0	0,624
Celular 4	0,873	1,407	0,393
Celular 5	0,151	0,974	0,293
Celular 6	0	0	0

Fonte: próprio autor.

Os valores com as distâncias positivas e negativas necessitam ser totalizados para gerar o coeficiente de aproximação de cada alternativa (item 14 da Figura 57).

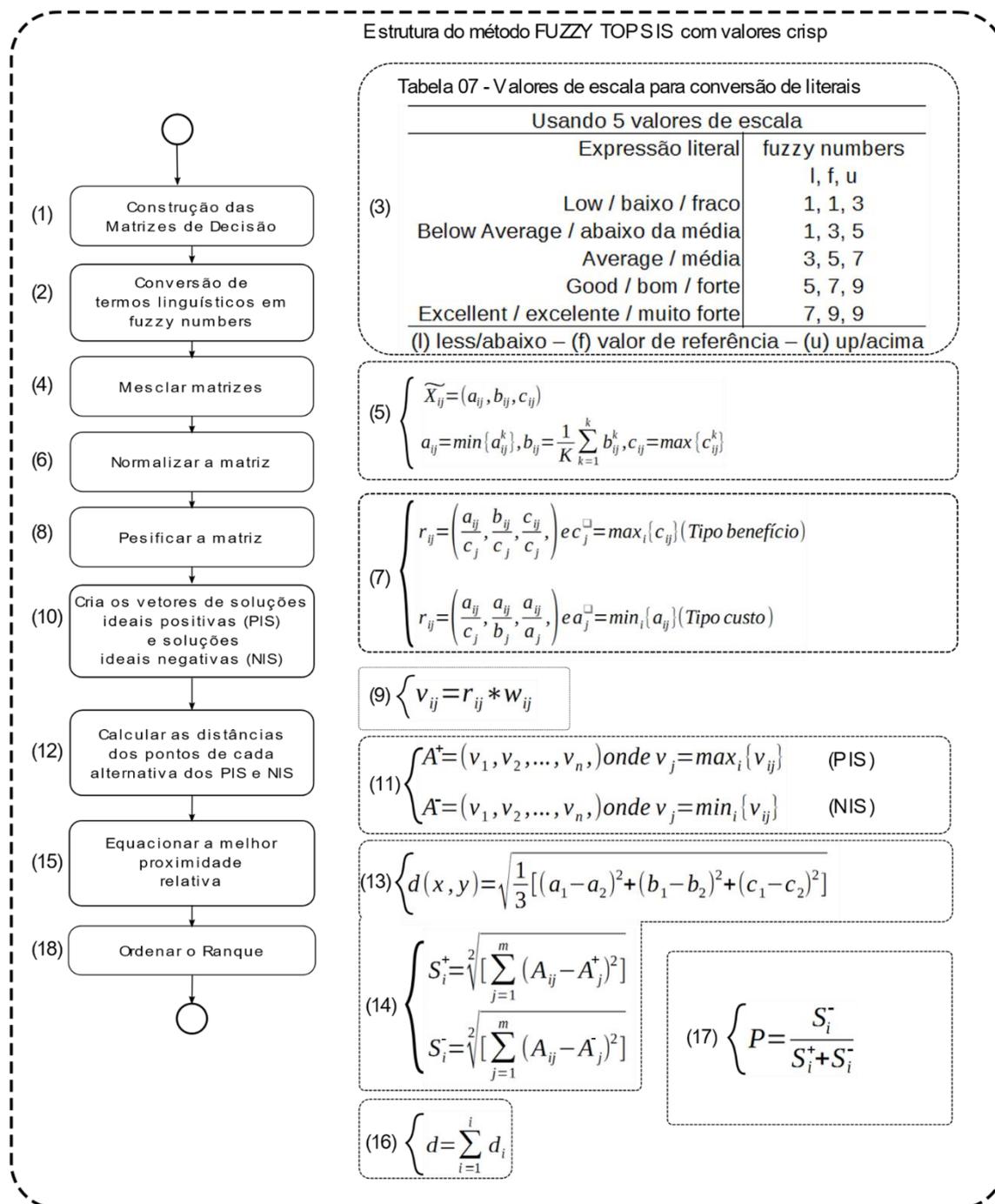
Figura 56 Matriz com o coeficiente de aproximação, ou distâncias globais

Smartphone / Atributo	S+	S-	$S^- / (S^- + S^+)$
Celular 1	1,31	1,6	0,55
Celular 2	1,4	1,5	0,52
Celular 3	2,07	0,83	0,29
Celular 4	0,23	2,67	0,92
Celular 5	1,49	1,42	0,49
Celular 6	2,9	0	0

Fonte: próprio autor.

O coeficiente de aproximação é calculado privilegiando a distância mais próxima da solução ideal positiva e mais distante da solução ideal negativa (item 17 da Figura 57).

Figura 57 Método FUZZY TOPSIS com valores crisp



Fonte: próprio autor.

2.10 AHP FUZZY TOPSIS COM MÚLTIPLOS DECISORES

A metodologia AHP, excelente pela geração de resultados consistentes item 2.6 - MÉTODO AHP, contribui com a formação dos pesos de cada critério (item 4 da Figura 7) e de seus respectivos valores fuzzy (item 2 da Figura 34).

Para a criação do valor de referência dos pesos, após terem sido verificadas as avaliações dos decisores como exemplo da Figura 45 e confirmada a consistência.

Figura 58 Matrizes de julgamento no método AHP por 3 decisores

Decisor 1			Decisor 2			Decisor 3					
	C1	C2	C3		C1	C2	C3		C1	C2	C3
C1	1	3	1/2	C1	1	1/3	3	C1	1	5	1/3
C2	1/3	1	1/7	C2	3	1	5	C2	1/5	1	1/7
C3	2	7	1	C3	1/3	1/5	1	C3	3	7	1

Fonte: Adaptado de Costa (2006)

Figura 59 Pesos consistentes resultantes

	peso Decisor 1	peso Decisor 2	peso Decisor 3	Peso de referência		
				l	f	u
C1	0,29	1	1,19	0,29	0,83	1,19
C2	0,09	2,47	0,31	0,09	0,95	2,47
C3	0,62	0,41	2,76	0,41	1,26	2,76

Fonte: próprio autor

Conforme os resultados na Figura 59, os valores abaixo (l) e acima (u) dos pesos são determinados respectivamente pelo menor e maior valores encontrados sobre cada critério. O valor de referência é igual à média dos valores de pesos de cada critério de cada julgamento.

Os julgamentos, segundo a lógica fuzzy, são feitos a partir de expressões literais (Figura 60), convertidos para valores crisp (Figura 61) a partir de uma tabela (Tabela 4), com a definição dos valores fuzzy abaixo e acima simultaneamente.

Figura 60 Matriz de julgamentos de decisores

	C 1 - A 1	C 1 - A2	C 1 - A 3	C 2 - A1	C 2 - A2	C 2 - A 3	C 3 - A 1	C 3 - A 2	C 3 - A 3
D1	medium	Very high	medium	Very high	medium	high	medium	medium	medium
D2	medium	medium	low	high	low	medium	medium	medium	low
D3	high	Very high	high	high	high	high	high	high	high
D4	low	Very high	medium	Very high	high	Very high	high	low	high
D5	low	Very high	medium	high	medium	high	medium	low	medium

Fonte: próprio autor.

Figura 61 Matriz fuzzificada de julgamentos de decisores

	C1 - A1			C1 - A2			C1 - A3			C2 - A1			C2 - A2			C2 - A3			C3 - A1			C3 - A2			C3 - A3		
	I	f	u	I	f	u	I	f	u	I	f	u	I	f	u	I	f	u	I	f	u	I	f	u	I	f	u
D1	3,0	5,0	7,5	7,5	9,0	10	3,0	5,0	7,5	7,5	9,0	10	3,0	5,0	7,5	5,0	7,5	9,0	3,0	5,0	7,5	3,0	5,0	7,5	3,0	5,0	7,5
D2	3,0	5,0	7,5	3,0	5,0	7,5	1,0	3,0	5,0	5,0	7,5	9,0	1,0	3,0	5,0	3,0	5,0	7,5	3,0	5,0	7,5	3,0	5,0	7,5	1,0	3,0	5,0
D3	5,0	7,5	9,0	7,5	9,0	10	5,0	7,5	9,0	5,0	7,5	9,0	5,0	7,5	9,0	5,0	7,5	9,0	5,0	7,5	9,0	5,0	7,5	9,0	5,0	7,5	9,0
D4	1,0	3,0	5,0	7,5	9,0	10	3,0	5,0	7,5	7,5	9,0	10	5,0	7,5	9,0	7,5	9,0	10	5,0	7,5	9,0	1,0	3,0	5,0	5,0	7,5	9,0
D5	1,0	3,0	5,0	7,5	9,0	10	3,0	5,0	7,5	5,0	7,5	9,0	3,0	5,0	7,5	5,0	7,5	9,0	3,0	5,0	7,5	1,0	3,0	5,0	3,0	5,0	7,5

Fonte: próprio autor.

A tabela de valores para conversão de expressões linguísticas para números reais dependerá da natureza de cada problema.

Tabela 5 Tabela de valores de conversão de expressões linguísticas para valores crisp

Very low			low			medium			high			Very high		
1			2			3			4			5		
l	f	u	l	f	u	l	f	u	l	f	u	l	f	u
0	1,0	3,0	1,0	3,0	5,0	3,0	5,0	7,5	5,0	7,5	9,0	7,5	9,0	10

Fonte: próprio autor.

A Matriz com múltiplos decisores é uma Matriz tridimensional (decisor x critério x alternativa) (Figura 61) e sofre uma redução para uma Matriz bidimensional: alternativa por critério (Figura 62), para a aplicação do método. Nessa redução os valores de limite inferior e de limite superior para cada conjunto de respostas equacionadas são identificados como os menores (l) e maiores (u) valores encontrados em cada conjunto de valores correspondente, assim, o menor valor entre

[3,0; 3,0; 5,0; 1,0; 1,0] é 1,0; a média dos valores [5,0; 5,0; 7,5; 3,0 e 3,0] é 4,7 e, o maior valor entre [7,5; 7,5; 9,0; 5,0; 5,0] é 9,0.

Figura 62 Matriz reduzida de decisores para critérios por alternativas

	C 1			C 2			C 3		
	l	f	u	l	f	u	l	f	u
A 1	1	4,7	9	5	8,1	10	3	6	9
A 2	3	8,2	10	1	5,6	9	1	4,7	9
A 3	1	5,1	9	3	7,3	10	1	0,6	9

Fonte: próprio autor.

A normalização (Figura 63), tal qual a metodologia FUZZY TOPSIS, considera, para critérios do tipo benefício, o valor dividido pelo maior valor do critério para todos os valores fuzzy (l, f e u), e, para os critérios do tipo custo divide o menor valor do critério pelo valor invertido, ou, para o valor “l” divide-se o menor valor pelo valor “u”, para o valor “f” divide-se o menor valor pelo valor “f” e para o valor “u” divide-se o menor valor pela valor “l”.

Figura 63 Matriz normalizada considerando o tipo de critério (Custo x Benefício)

	Benefício			Benefício			Custo		
	C 1	C 1	C 1	C 2	C 2	C 2	C 3	C 3	C 3
A 1	0,1	0,47	0,9	0,5	0,81	1	0,11	0,17	0,33
A 2	0,3	0,82	1	0,1	0,56	0,9	0,11	0,21	1
A 3	0,1	0,51	0,9	0,3	0,73	1	0,11	1,67	1

Fonte: próprio autor.

Os pesos utilizados não seguem a tabela de valores de conversão, resultam dos valores gerados pela metodologia AHP.

Figura 64 Matriz de pesos oriundos do método AHP

pesos	[0,3	0,8	1,2	0,1	1	2,5	0,4	1,3	2,8]
-------	---	-----	-----	-----	-----	---	-----	-----	-----	-----	---

Fonte: próprio autor.

O procedimento seguinte é exatamente o mesmo da metodologia TOPSIS ou FUZZY TOPSIS: a pesificação (Figura 65), que consiste no produto da Matriz normalizada pela Matriz de pesos.

Figura 65 Matriz pesificada (Matriz normal x pesos)

	C 1	C 2	C 3
A 1	0,029	0,136	0,261
A 2	0,087	0,238	0,29
A 3	0,029	0,148	0,261

Fonte: próprio autor.

As soluções ótimas positivas e negativas, mantêm os maiores valores para a Matriz de soluções ideais positivas (PIS) e menores para a Matriz de soluções ideais negativas (NIS), tal qual o método FUZZY TOPSIS (Figura 66).

Figura 66 Matrizes com soluções ideais positivas (PIS) e negativas (NIS)

A+	0,1	0,2	0,3	0,1	0,2	0,3	0	0,5	0,3
A-	0	0,1	0,3	0	0,2	0,3	0	0	0,1

Fonte: próprio autor.

Cada alternativa tem as distâncias de seus pontos em relação a PIS (Figura 67), reduzindo os valores fuzzy para um valor utilizando o cálculo de distância euclidiana dos pontos, como em um plano cartesiano. Também em relação a NIS (Figura 68), para chegarem ao cálculo do coeficiente de proximidade.

Figura 67 Matriz com distâncias dos pontos das alternativas em relação a PIS

	C 1	C 2	C 3
A 1	0,07	0	0,27
A 2	0	0,08	0,24
A 3	0,06	0,04	0

Fonte: próprio autor

Figura 68 Matriz com distâncias dos pontos das alternativas em relação a NIS

		C 1	C 2	C 3
A	1	0,17	0,23	0,07
d-	A 2	0,22	0,18	0,17
A	3	0,17	0,21	0,33

Fonte: próprio autor

O coeficiente de proximidade se constitui no cálculo que permite demonstrar a menor distância para melhor solução e maior distância da pior, a partir da divisão da distância S- pela soma das distâncias S- e S+.

Figura 69 Matriz dos coeficientes de proximidades das alternativas

		S-	S+	S- / (S- + S+)
A	1	0,47	0,34	0,58
d-	A 2	0,57	0,32	0,64
A	3	0,71	0,1	0,88

Fonte: próprio autor

3 METODOLOGIA

3.1 SÍNTESE DAS ETAPAS DE PESQUISA

Etapas para alcançar os objetivos propostos:

- Construção de formulário de coleta de dados;
- Aplicação da coleta em reunião com debate junto aos docentes da disciplina;
- Análise dos dados coletados de acordo com os objetivos;
- Seleção de alternativas;
- Seleção de critérios;
- Construção de formulário de coleta de dados para aplicação de valores sobre as alternativas de acordo com os critérios;
- Tabulação de dados;
- Revisão e ajustes para adequação com a razão de consistência;
- Aplicação do método;
- Análise;
- Apresentação dos resultados.

3.2 DESENVOLVIMENTO DOS FORMULÁRIOS

O primeiro formulário foi desenvolvido para elencar os critérios e as alternativas, o qual foi aplicado à coordenação de informática do campus Quissamã do Instituto Federal Fluminense, uma vez que a pesquisa se destina a potencializar os resultados locais no ensino de algoritmos para as turmas de primeiro ano do curso técnico em informática deste, embora, atenda à demanda de todos os Institutos Federais.

O formulário, sobre levantamento de sugestões do Apêndice 1, deu início ao levantamento sobre alternativas e critérios, fora aplicado em reunião de coordenação do núcleo de informática. Posteriormente, o formulário sobre levantamento de alternativas e de critérios do Apêndice 2, fora passado, sem sugerir nenhuma linguagem para desenvolvimento para sistemas Apple, por não haver uso desse sistema operacional em escala considerável nos laboratórios deste Instituto Federal,

bem como nenhuma linguagem, a exceção do Javascript, que dependam de protocolos de desenvolvimento para a internet. Mesmo abrindo espaço para receber novos critérios, após estes terem sido discutidos e elencados em reunião com a coordenação de informática, não houve novas indicações.

O formulário planilha de comparação de importância entre critérios no Apêndice 3, fora elaborado no formato planilha, explicado pessoalmente e compartilhado no Google, aplicado segundo a metodologia AHP. Seus dados foram aplicados em planilhas previamente elaboradas com dados simulados utilizando a metodologia AHP, com o objetivo de levantar os pesos dos critérios, a serem aplicados no ranqueamento de linguagens de programação na metodologia FUZZY TOPSIS.

O Formulário de Julgamentos de Professores, no Apêndice 4, fora aplicado aos professores da rede do Instituto Federal Fluminense e do Instituto Federal do Espírito Santo. Fora enviado utilizando-se um link gerado pelo recurso do formulário Google, para as coordenações de informática de campi que tem em sua oferta cursos de informática, precedidos de ligação com contato com coordenadores dos respectivos núcleos. As respostas foram automaticamente inseridas em planilha Google. Os resultados do formulário foram copiados para planilha previamente elaborada para substituição de dados simulados e aferição de ranking sobre as alternativas utilizando a metodologia AHP FUZZY TOPSIS.

3.3 ALTERNATIVAS

As alternativas selecionadas neste trabalho se limitaram ao conhecimento dos professores envolvidos no processo de ensino do IFFluminense campus Quissamã. Esta restrição de alternativas foi definida uma vez que, considerar a experiência dos profissionais avaliadores envolvidos diretamente no objeto desta pesquisa, se constitui essencial, pois, as linguagens de programação a serem utilizadas, devem estar contidas dentro do escopo de seu conhecimento e experiência. O conhecimento e a prática docente no ensino das linguagens de programação estudadas geram as percepções necessárias quanto as características necessárias a serem avaliadas nesta pesquisa.

Como princípios para a seleção das alternativas, deve o decisor ter o conhecimento sobre a linguagem de forma não superficial, preferencialmente que tenha passado pela experiência de aprendizado e desenvolvimento com a mesma.

Com base no instrumento de coleta de dados (Apêndice 1 e 2) foram elencadas as seguintes alternativas: Java, JavaScript, PHP, C, C#, C++, Python, Objective-C, Ruby, HTML com CSS, Typescript, VB.Net, R, Perl, Assembly, Portugol WebStudio, Code.org, Scratch, studio.code.org, em um total de dezenove linguagens de programação/tecnologia de ensino de programação. Destas, foram consideradas apenas oito: C; studio.code.org; Java; Scratch; JavaScript, Portugol WebStudio, Python e C#.

A linguagem C é a mais comumente aplicada em universidades e cursos técnicos, dentre as alternativas, apresenta farta literatura e exemplos disponíveis, é uma linguagem compilada, estruturada e altamente imperativa. Apesar de uma sintaxe simples, pode ser utilizada para fazer desde pequenos programas até sistemas operacionais complexos, portanto com muitas aplicações, considerada de alto nível por ser próxima da linguagem humana (GUDWIN, 1997).

O Code.org® (2020), responsável pelo site <https://studio.code.org/courses>, é uma organização sem fins lucrativos dedicada ao ensino da ciência da computação projetando cursos próprios e de terceiros para o mundo todo em diversos idiomas. A Code.org é apoiada por: Amazon, Facebook, Google, a Infosys Foundation, Microsoft e muitas outras. Trabalha com blocos de programação e são adequadas para ensinar crianças¹.

O Scratch é uma linguagem de programação gráfica desenvolvida pelo MIT (*Massachusetts Institute of Technology*). Constitui-se em um software que se utiliza de blocos lógicos, e itens de som e imagem, para o desenvolvimento de histórias interativas, jogos e animações, usado em mais de 150 países. Foi projetado para idades entre 8 e 16 anos, mas usado por todas as idades. Possibilita uma versão instalada, não sendo apenas online² (MARJI, 2014).

O Javascript constitui uma linguagem de programação baseada em scripts e padronizada pela “ECMA International” voltada para a produção de conteúdo para a web, utilizada para manipulação de todas as propriedades dos elementos de uma página, seja de HTML ou de CSS, validar formulários, bem como realizar cálculos e processamentos que forem necessários no computador do usuário (cliente)³.

1 Disponível em: <https://studio.code.org/courses>. Acesso em: 01 jul. 2020.

2 Disponível em: <http://www.scratchbrasil.net.br/index.php/sobre-o-scratch.html>. Acesso em: 01 jul. 2020.

3 Disponível em: <https://medium.com/trainingcenter/afinal-javascript-e-ecmascript-s%C3%A3o-a-mesma-coisa-498374abbc47>. Acesso em: 01 jul. 2020.

A linguagem Java, mais utilizada entre os professores, possui uma sintaxe próxima as linguagens C e C++, na parte estruturada; multiplataforma, fora a alternativa primeiramente apresentada pelos professores do IFFluminense, até porque é atualmente utilizada nos anos letivos posteriores para a disciplina de POO (Programação Orientada a Objeto). É uma linguagem de programação orientada a objeto, o código nela produzido é compilado para um “bytecode”, código que será interpretado em uma máquina virtual intermediando a aplicação com o hardware, composta por API’s (grupo de programas de suporte com funções específicas) e bibliotecas para potencializar a programação⁴ (BURD, 2013).

O Portugal WebStudio desenvolvido e mantido pelo Laboratório de Inovação Tecnológica na Educação da UNIVALI e utilizada por diversas universidades e institutos de tecnologia, é uma pseudolinguagem, um ambiente para aprendizagem de programação, voltado para os iniciantes que falam o idioma português. Assemelha-se a linguagem C, tanto na estrutura, quanto na utilização de bibliotecas, embora muito próximo do Português, possui sintaxe própria e diferenciada. Não se a utiliza em aplicações reais, somente para ensino⁵ (STEFFENS, 2020).

A linguagem Python é uma excelente linguagem, com boa legibilidade e com suporte a múltiplos paradigmas de programação. Constitui-se em uma linguagem de programação de alto nível, interpretada, de script, imperativa, orientada a objetos, de tipagem dinâmica e forte. Apesar de um menor domínio sobre esta, e a existência de muitas versões, incompatíveis entre si, fora escolhida pela sua facilidade, versatilidade e potencial de desenvolvimento⁶ (MUELLER, 2016).

A linguagem C# (leia-se C-Sharp) é uma linguagem de programação orientada a objetos, faz parte da plataforma Microsoft .Net. Baseada nas linguagens C++ e Java, constitui-se numa linguagem de propósitos gerais e fortemente tipada, ou seja as variáveis possuem um tipo bem definido e que necessita ser informado no momento de sua declaração , também adequada as séries posteriores⁷.

Algumas alternativas, mesmo que potencialmente relevantes, foram descartadas por não estarem de acordo com os princípios pré-estabelecidos acima, a

4 Disponível em: <https://www.devmedia.com.br/guia/linguagem-java/38169>. Acesso em: 01 jul. 2020.

5 Disponível em: <http://lite.acad.univali.br/portugol/>. Acesso em: 01 jul. 2020.

6 Disponível em: <https://python.org.br/>. Acesso em: 01 jul. 2020.

7 Disponível em: <https://docs.microsoft.com/pt-br/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework>. Acesso em: 01 jul. 2020.

saber: Objective-C, Ruby, Typescript, VB.Net, R, Perl, Assembly. Outras linguagens, apesar de conhecidas pelos profissionais consultados, foram descartadas por não se adequarem ao pensamento principal de manter a maior simplicidade possível sobre a linguagem ou tecnologia, com foco em algoritmos e na programação estruturada independente do potencial de produtividade da linguagem.

A linguagem HTML não constitui propriamente uma linguagem de programação, não tem lógica de programação, não tem estruturas de controle, nem estruturas condicionais, nem de repetição de comandos, serve apenas para marcação, ou seja, para colocar os elementos na página da internet. Para posicionar estes elementos, ou dar-lhes características diferenciadas de cor, tamanho, ou organização na tela do computador ou de quaisquer dispositivos, usa-se o Cascade Style Sheet (CSS) que define as características dos objetos na página, constitui uma linguagem de estilos para alterar a apresentação dos objetos da página web. Ambas não aplicam algoritmos como desejado para a proposta⁸.

A linguagem PHP tem como pré-requisito o conhecimento de HTML e, preferencialmente, o conhecimento sobre JavaScript, e, embora poderosa, com grande potencial de programação, especialmente para web, justifica seu aprendizado apenas para os anos seguintes⁹, não para o primeiro ano, de acordo com o contexto estabelecido neste trabalho (BELÉM; MEZA; VIANNA, 2014).

A linguagem C++, embora bem conceituada, em especial pela penetração em jogos, fora descartada em função da plataforma requerida para seu desenvolvimento e do menor uso e prática por parte dos professores da instituição. A linguagem C++, nativa nas plataformas do Instituto Federal Fluminense, gera código compilado para uma plataforma específica. Caso a alternativa fosse para os 2º e 3º anos, C++ seria mais interessante por apresentar mais funcionalidades¹⁰.

Há no mercado uma quantidade substancial de sites e cursos, em grande parte gratuitos, focados na formação de programadores. Dependeria apenas do aluno ter atitude e disponibilidade para ingressar nessa formação. Algumas plataformas interativas são mais conhecidas entre os docentes do Instituto Federal como:

8 Disponível em: <https://www.devmedia.com.br/guia/programador-front-end/38792>. Acesso em: 01 jul. 2020.

9 Disponível em: <https://www.devmedia.com.br/guia/programador-php/38193>. Acesso em: 01 jul. 2020.

10 Disponível em: <https://docs.microsoft.com/pt-br/cpp/cpp/cpp-language-reference?view=vs-2019>. Acesso em: 01 jul. 2020.

code.org, codeacademy.com, ou khanacademy.org, com opções de ensino de diversas linguagens, tecnologias e conhecimentos afins, que poderiam ser usadas diretamente para ministrar as aulas, ou acessoriamente para complementá-las. A seguir apresenta uma lista com plataformas que disponibilizam as opções referidas acima:

- codewars.com
- coursera.org
- dash.generalassemb.ly
- developer.apple.com
- developers.google.com
- edx.org/course
- freecodecamp.org
- ninjadolinux.com.br/cursos-online
- ocw.mit.edu/index.htm
- odeavengers.com
- pluralsight.com/codeschool
- pt.khanacademy.org
- [scirra.com\(constructo\)](http://scirra.com(constructo))
- theodinproject.com
- udacity.com
- udemy.com
- unity.com/learn
- upskillcourses.com
- w3schools.com

3.4 CRITÉRIOS

Os critérios devem avaliar o desempenho das alternativas verificando todas as propriedades relevantes, sem redundâncias ou superposições propiciando a compreensão de decisores e avaliadores (COSTA, 2002).

Para os Institutos Federais, a essencialidade nesta etapa constitui o entendimento sobre algoritmos, o desenvolvimento dos códigos primários, a parte estrutural, necessários à programação de aplicativos nos anos seguintes.

Todos os critérios foram elaborados e avaliados a partir de questionários e debates dentro do núcleo de informática (Apêndices 1 e 3), aplicados aos três decisores. Utilizou-se a média aritmética dos resultados sobre a avaliação dos docentes para gerar o vetor de pesos de cada critério e seus limites abaixo e acima do peso de referência.

Foram selecionados cinco critérios: Legibilidade; Depuração de Erro; Curva de Aprendizagem alta; Ferramentas de escrita de código e Aplicabilidade em projetos reais, baseados nos critérios de Parker, Ottaway e Chao (2006) e Prokop *et al.* (2019).

O critério Legibilidade faz referência a facilidade para a escrita e leitura de código e o entendimento do algoritmo desenvolvido em determinada linguagem, ou seja, qual a facilidade em visualizar o que está sendo desenvolvido, a velocidade com que se percebe a estrutura do algoritmo e a simplicidade para a escrita são essenciais para o ensino e a aprendizagem.

O critério depuração de erro, visto por Parker, Ottaway e Chao (2006) como, Instalações de depuração e boas ferramentas de depuração de Prokop *et al.* (2019), diz respeito a confiabilidade da linguagem permite ao desenvolvedor mecanismos de segurança para a escrita e execução de código em plataformas distintas com um bom tempo de resposta. Durante a programação normalmente ocorrem muitos erros, principalmente no início. A facilidade em percebê-los e depurá-los constitui fator determinante para um melhor desempenho. A identificação do erro pode ser usada propositalmente para provocar percepções quanto ao funcionamento do algoritmo e dos recursos. Nesta etapa do aprendizado não se avalia os comportamentos do aplicativo em plataformas distintas, em rede/ ou sobre bancos de dados, e sim a capacidade de identificar erros com facilidade, sejam sintáticos ou semânticos.

O critério Curva de Aprendizagem, refere-se à velocidade e qualidade de evolução dos alunos sobre o entendimento e capacidade de criação de algoritmos. Este critério pode ser mensurado a partir da aplicação de um volume de atividades com dificuldade crescente a grupos diferentes. Sua importância está no quão rápido será para o discente escrever seus primeiros códigos em determinada linguagem. Para comparar as alternativas à luz desse critério os professores foram questionados quanto à facilidade dos alunos em resolver algoritmos em uma linguagem ou outra, bem como o tempo de entendimento e resposta sobre as atividades em cada linguagem.

O critério Ferramentas de escrita de código é fundamental, uma vez que o código só existe se escrito. Além da disponibilidade, a usabilidade, interface amigável e os recursos da própria ferramenta para auxiliar na escrita, devem ser considerados neste quesito a depuração e execução do código. Da mesma forma que na Disponibilidade de Material Gratuito, não necessita utilizar todos os possíveis aplicativos existentes para a escrita e execução do código. Uma vez que haja um editor eficaz disponível, de boa usabilidade, que atenda ao processo de edição, execução e depuração do código. Podem ser ferramentas para escrita do código: editores de texto, sites, IDE's, frameworks e aplicativos capazes de realizar esta tarefa.

A Aplicabilidade em Projetos Reais constitui um critério que se encontra entre o desejável e o ideal, mas, não essencial, visto que, o fato de produzir aplicativos utilizáveis na prática, torna o aprendizado mais motivante e potencialmente mais relevante para a formação do programador, embora não seja esta a expectativa nesta disciplina introdutória.

O trabalho Parker et al. (2006) é uma importante referência na área de ensino de linguagens de programação e, por este motivo, foi usado na seleção de critérios. Os critérios propostos pelo autor estão no Quadro 5:

Outro trabalho, mais recente, embora ainda sem o reconhecimento tal qual o de Parker, Ottaway e Chao (2006), foi o trabalho de Prokop *et al.* (2019), a partir de uma pesquisa sobre a definição de critérios modernos voltados para a seleção de linguagens de programação para cursos introdutórios de informática, com base em no estudo de literatura dos últimos 15 anos, faz uma classificação de dezenove critérios mais utilizados, estes critérios estão no Quadro 5:

Quadro 5 Quadro com Critérios significativos

(continua)

Parker	Prokop	Frames	Selecioneados
Custo financeiro razoável	Facilidade de aprender conceitos básicos	Popularidade, o quão conhecido e reconhecido	Legibilidade
Disponibilidade da Versão Acadêmica / Estudante	Demanda no mercado de trabalho	Tamanho da Comunidade, maior comunidade, maiores são as discussões e as atualizações	Depuração de Erro
Aceitação Acadêmica	Presença de todas as principais construções no idioma	Suporte, qual a qualidade e facilidade no suporte	Curva de Aprendizagem alta
Disponibilidade de livros didáticos	Facilidade de transição para outras linguagens de programação nos cursos subsequentes	Documentação, manual técnico e sua qualidade, disponibilidade e volume da documentação oficial e	Ferramentas de escrita de código
Estágio no ciclo de vida	O custo do compilador para estudantes	Disponibilidade de Recursos, Informações do framework na Internet, na Literatura e em artigos científicos	Aplicabilidade em projetos reais
Aceitação da indústria	Disponibilidade de livros didáticos		
Comercialização (regional e nacional)	Suporte para programação orientada a objetos		
Requisitos de sistema do aluno / versão acadêmica / completa	Custos financeiros razoáveis para criar um ambiente de aprendizado		
Dependência do sistema operacional	Comunidades online		
Proprietário / Código Aberto	Boas ferramentas de depuração		
Ambiente de desenvolvimento	Facilidade de uso do ambiente de desenvolvimento integrado (ou compilador online)		
Instalações de depuração	Estrutura do grau (Structure of degree)		
Facilidade de aprendizado de conceitos fundamentais	Independência da plataforma		
Suporte para código seguro	Perspectivas de suporte adicional dos desenvolvedores		
Recursos avançados para cursos de programação subsequentes	Instalação fácil		
Script ou idioma com todos os recursos	Disponibilidade de extensões / bibliotecas		
Suporte ao Desenvolvimento Web	Programação visual		

Quadro 5 Quadro com Critérios significativos

(conclusão)

Parker	Prokop	Frames	Selecionados
Suporta Domínio de Aplicativo de Destino	Programação segura		
Suporte à Abordagem de Ensino	Suporte ao desenvolvimento da Web		
Suporte orientado a objetos			
Disponibilidade do Suporte			
Treinamento para instrutores e funcionários			
Nível de Experiência Previsto para Alunos Entrantes			

Fonte: próprio autor.

Uma parcela significativa dos critérios de seleção de linguagens de programação encontrados, levam em consideração a programação em um sentido amplo, voltado para a produtividade e não de forma específica para a formação da base do pensamento computacional, ou, no domínio de algoritmos e programação estruturada. Fica evidente a necessidade de reduzir a imprecisão quanto a melhor estratégia de construção deste conhecimento, de identificar uma linguagem mais adequada para os anos iniciais do ensino técnico. Espera-se que as alternativas propiciem o desenvolvimento continuado do aluno e facilitem o ensino de algoritmo em conjunto com linguagem de programação, bem como possibilite a redução dos problemas de evasão e retenção desta disciplina.

Parker et al. (2006) propõem que não haja a utilização de linguagens diferentes, para os diferentes anos de formação, não contempla as necessidades específicas do aprendiz ainda não detentor dos rudimentos da programação e conceitos envolvidos. Levando-se em consideração esta divisão, ficariam de fora os critérios: Estágio no ciclo de vida; Aceitação da indústria; Comercialização; Dependência do sistema operacional; Proprietário / Código Aberto; Segurança de codificação e suporte para código seguro; Recursos avançados para cursos de programação subsequentes; Suporta Domínio de Aplicativo de Destino; Suporte orientado a objetos; Disponibilidade do Suporte.

O critério Disponibilidade de Material Gratuito apesar de constituir-se fundamental, fora encontrado com facilidade para todas as alternativas selecionadas. Sua essencialidade reside na possibilidade e facilidade de o aluno encontrar suporte

externo à aula para solução de dúvidas e evolução de conhecimento. Há uma grande diversidade de materiais de apoio possíveis, além das plataformas de ensino listadas, como o guia de referência e tutoriais da própria linguagem que podem trazer excelentes práticas de programação, bem como, vídeo aulas, livros, revistas, artigos, blogs, fóruns, comunidades e/ outros. Considera-se, por bom senso, que o aluno não necessita acessar todos os possíveis e diversos recursos didáticos de apoio. Essencialmente há que ter material de apoio e este, de boa qualidade, no que diz respeito ao auxílio na aprendizagem, portanto, basta que tenha quantidade suficiente, acessível e com boa eficácia.

A colocação Segurança de Codificação e Suporte para código seguro e orientação a objetos constituiria um excesso tanto na complexidade, quanto no volume de informação (discussão → localizar o texto base).

Quanto aos critérios Custo Financeiro Razoável e Disponibilidade da Versão Acadêmica / Estudante, todas as linguagens, nesta etapa, mesmo as pagas, são disponibilizadas gratuitamente para ensino, bem como as IDE'S que são conjunto de ferramentas para realizar o desenvolvimento de programas em uma linguagem, o quesito Custo tornou-se irrelevante, assim também todas as linguagens apresentadas possuem versões gratuitas para ensino.

O critério Aceitação Acadêmica foi desconsiderado, uma vez entendida essa aceitação de forma implícita, pois aceitável pelos próprios docentes pertencentes à pesquisa. Não há que se ponderar essa aceitabilidade e sim as características da alternativa, de forma objetiva, com pesos sobre os elementos da linguagem que afetam na potencialização do ensino de algoritmos e da própria linguagem.

O Estágio no ciclo de vida afeta a continuidade e a produtividade, mas, não baliza a escolha para linguagens iniciais, uma vez que estas não necessitam da obrigatoriedade de continuidade/ ou produtividade.

Embora essencial, os Requisitos de sistema do aluno / versão acadêmica / completa são atendidos para todas as linguagens de programação levantadas, tanto o hardware e sistemas operacionais existentes são suficientes à demanda, não justificando a utilização deste critério.

A portabilidade não constitui um problema na etapa inicial desta pesquisa, ou não é essencial sobre as alternativas elencadas, pois, algumas são oferecidas na nuvem, outras são disponibilizadas em versões compatíveis com os sistemas

operacionais existentes utilizados, conseqüentemente, não há necessidade do critério Dependência do sistema operacional, nesta.

Sobre linguagens, ao se falar em Script ou idioma com todos os recursos, faz-se referência à complexidade da linguagem, à abordagem sobre a forma de escrita e os recursos oferecidos, neste sentido, este trabalho considera essa complexidade do Script, do idioma, da sintaxe, diluídos no critério Facilidade de Aprendizado de Conceitos Fundamentais e o Ambiente de Desenvolvimento.

3.5 MÉTODO AHP PARA DETERMINAR OS PESOS DOS CRITÉRIOS

De acordo com o objetivo deste trabalho, os critérios foram avaliados por três docentes do Instituto Federal Fluminense, a partir do uso de formulários para gerar o comparativo de importância de cada critério. Essa avaliação foi feita em reunião do núcleo de informática, sob orientação para a finalidade desejada.

Cada professor julgou os critérios e as alternativas a partir de uma planilha compartilhada seguindo orientações (Apêndices 3 e 4). Comparou-se critério por critério e critérios por alternativas, atribuindo-se valores dentro da escala de valor para julgamentos paritários criada por Saaty (1990).

Os comparativos com as alternativas foram realizados por outros formulários online com a utilização do método FUZZY TOPSIS, de maior facilidade computacional e de maior precisão, além de possuir maior facilidade e flexibilidade para inserção de diversas avaliações de outros decisores sem custo de tempo significativo para o processamento de informações.

Para a demonstração dos métodos abaixo, segue o quadro com correspondência entre cada critério e entre cada alternativa, com seus respectivos valores de representação:

Quadro 6 Correspondência das representações de critérios e alternativas

Critério / Alternativa	Representação
Legibilidade	C1
Depuração de Erro	C2
Curva de Aprendizagem Alta	C3
Ferramentas de Escrita de Código	C4
Aplicabilidade em Projetos Reais	C5
Linguagem C	A1
studio.code.org	A2
Java	A3
Scratch	A4
JavaScript	A5
Portugol WebStudio	A6
Python	A7
C#	A8

Fonte: próprio autor.

A inconsistência do ser humano em seus julgamentos lhe é inerente, resultado de influências internas ou externas à tomada de decisão, com possibilidade de aumentar esta inconsistência proporcionalmente ao aumento do número de julgadores (COSTA, 2006).

Com o intuito de garantir pesos consistentes para avaliar as alternativas, usou-se a metodologia AHP. Após levantamento dos julgamentos utilizando o formulário do Apêndice 3, durante reunião com os docentes do núcleo de informática, responsáveis pelo curso e disciplinas de programação, obteve-se as avaliações dos critérios por três decisores.

Estas avaliações resultaram em um vetor de pesos para cada decisor, fez-se a média aritmética entre os pesos atribuídos por cada decisor, referente a cada critério destes para a geração do peso de referência e, para aplicação destes no ranqueamento pelo método FUZZY TOPSIS, definiu-se o valor *less* (abaixo) como o menor julgamento dentre os três e ao valor *up* (acima) o maior (Figura 56). A combinação destes vetores com a metodologia FUZZY TOPSIS resulta na metodologia AHP FUZZY TOPSIS, mais adequada para tratar um grande volume de avaliações sobre as alternativas.

Quadro 7 Matriz de geração de pesos

	D1	D2	D3	Média	l	f	u
C1	r ₁₁	r ₁₂	r ₁₃	(r ₁₁ +r ₁₂ +r ₁₃)/n	Min(r ₁₁ ; r ₁₂ ; r ₁₃)	Média	Max(r ₁₁ ; r ₁₂ ; r ₁₃)
C2	r ₂₁	r ₂₂	r ₂₃	(r ₂₁ +r ₂₂ +r ₂₃)/n	Min(r ₂₁ ; r ₂₂ ; r ₂₃)	Média	Max(r ₂₁ ; r ₂₂ ; r ₂₃)
C3	r ₃₁	r ₃₂	r ₃₃	(r ₃₁ +r ₃₂ +r ₃₃)/n	Min(r ₃₁ ; r ₃₂ ; r ₃₃)	Média	Max(r ₃₁ ; r ₃₂ ; r ₃₃)

Fonte: próprio autor

l = menor valor avaliado

f = média entre os valores correspondentes dos avaliadores

u = maior valor avaliado

i = linha/critério

j = coluna/decisor

n = número de decisores

C_i = critérioD_i = decisorr_{ij} = resultado do critério C_i do Decisor D_iM = (r_{ij}+r_{ij+1}+r_{ij+2})/n (Média)

$$dp = \sqrt{\frac{\sum_{i=1}^n (r_i - M_i)^2}{n}} \text{ (Desvio padrão)}$$

dp/f% = dpi/Max(M_i) (Percentual do desvio padrão sobre o maior valor)

Para a geração dos valores fuzzy abaixo (l) e acima (u) pode-se utilizar o valor do dp (desvio padrão), subtraindo-se e somando-se à média, para gerar os respectivos valores ou, o menor e o maior valor observados, caso o desvio padrão seja muito pequeno e seja desejado um valor fuzzy com maior abrangência sobre a variação de respostas dos decisores.

3.6 MÉTODO AHP FUZZY TOPSIS COM MÚLTIPLOS DECISORES

Os formulários de levantamento de avaliações dos professores de informática foram feitos na plataforma Google online, pelo aplicativo Google *Forms*, com geração automática de respostas em uma planilha Google. Foram recebidas 21 avaliações. Estas foram copiadas em planilha previamente elaborada para serem colados substituindo os dados de simulação e, calculado o ranking das linguagens de apoio ao ensino de algoritmo.

As respostas dos decisores, originalmente apresentam-se sob a forma de expressões linguísticas (literais) conforme o Quadro 7, de acordo com a lógica fuzzy. Após o recebimento das respostas, estas são fuzzificadas, convertidas em valores numéricos, com os valores abaixo e acima, respectivamente ao lado esquerdo e direito de cada avaliação, conforme Tabela 5 de fuzzificação.

A matriz de resposta do formulário apresenta três dimensões: decisor, critério e alternativa. Para aplicar a método de ranqueamento, primeiramente faz-se a redução aglutinando as respostas dos decisores para uma matriz composta por critérios por alternativas. A diferença na aplicação desta metodologia neste contexto, basicamente incide sobre a delimitação dos limites mínimos e máximos que passam a levar em consideração as piores e melhores avaliações respectivamente aos valores de limite inferior e superior de cada conjunto fuzzy.

Neste trabalho, todos os passos foram baseados na metodologia proposta por Chen (2000) com exceção do cálculo da solução ideal PIS e NIS, que foram baseadas no trabalho de Yuen (2014).

Na escala atribuiu-se um número difuso triangular a cada termo linguístico, uma escala unidimensional, em seguida fez-se classificação (ponderação) e agregação pela média aritmética ponderada dos termos fez-se uma classificação única; Gerou-se uma Matriz de decisão de valor normalizada para obter o valor dentro do intervalo $[0,1]$; Os pesos gerados pelo método AHP foram normalizados para a metodologia FUZZY TOPSIS com seus respectivos valores abaixo e acima; Na etapa de soluções ideais baseou-se em Yuen (2014) com definição de um valor máximo na matriz de decisão (para positivo) e valor mínimo (para negativo) e não (PIS)= $(1;1;1)$ e (NIS)= $(0;.0; 0)$; Calculou-se a distância da solução ideal para cada alternativa usando o método de vértice, com base na distância euclidiana, gerando duas matrizes, uma com valores das distâncias em relação a solução ideal positiva e/ outra com em relação a solução ideal negativa; O coeficiente relativo de proximidade de cada alternativa fora calculado a partir das matrizes de distância das soluções positivas e negativas e por fim, fez a classificação das alternativas, onde, quanto maior o valor relativo de proximidade melhor.

Quadro 8 Valores de conversão de literal para crisp

Critério: Legibilidade		Critério: Ferramentas e Recursos de Escrita de Código	
Valor literal	Escala	Valor literal	Escala
Péssima Legibilidade	1	Escrita muito complexa e/ ou muito difícil	1
Difícil Legibilidade	2	Escrita complexa e/ ou difícil	2
Média Legibilidade	3	Escrita Regular	3
Fácil Legibilidade	4	Escrita simples e/ ou fácil	4
Muito fácil Legibilidade	5	Escrita muito simples e/ ou muito fácil	5
Critério: Facilidade em Depurar Erro		Critério: Aplicabilidade em Projetos Reais	
Valor literal	Escala	Valor literal	Escala
Muito difícil depurar erro	1	Não aplicável ou quase nula	1
Difícil depurar erro	2	Pouca aplicável	2
Regular depuração de erro	3	Média aplicação	3
Fácil depurar erro	4	Bastante aplicável	4
Muito Fácil depurar erro	5	Muito aplicável	5
Critério: Facilidade de Aprendizagem			
Valor literal	Escala		
Muito difícil de aprender	1		
Difícil de aprender	2		
Média aprendizagem	3		
Fácil de aprender	4		
Muito fácil de aprender	5		

Fonte: próprio autor.

Tabela 6 Tabela com valores para fuzzificação

Valores para fuzzificação														
l	f	u	l	f	u	l	f	u	l	f	u	l	f	u
0,0	1,0	3,0	1,0	3,0	5,0	3,0	5,0	7,5	5,0	7,5	9,0	7,5	9,0	10,0
	1			2			3			4			5	

Fonte: próprio autor.

Para realizar a fuzzificação utiliza-se a fórmula $a_{ij} = \min\{a_{ij}^k\}$, $b_{ij} = \frac{1}{K} \sum_{k=1}^K b_{ij}^k$, $c_{ij} = \max\{c_{ij}^k\}$, sendo k o número de decisores (respostas dos questionários), i o número de alternativas, j o número de critérios, a representando o valor *less*, b representando o valor de referência e c o valor *up*. Assim, o valor a_{ij} é o menor valor avaliado pelos decisores para a alternativa naquele critério,

respectivamente o valor de referência b_{ij} é a calculado pela média aritmética das avaliações dos decisores, e o valor de c_{ij} é o maior valor dentre as avaliações.

A matriz de decisão difusa normalizada é construída usando transformação de escala linear para transformar as várias escalas de critérios em escalas comparáveis. Para que os dados tenham a devida proporcionalidade, a normalização leva em consideração o tipo do critério, calculando novos valores em uma nova Matriz a partir das fórmulas:

$$r_{ij} = \left(\frac{a_{ij}}{c_j}, \frac{b_{ij}}{c_j}, \frac{c_{ij}}{c_j} \right) e c_j^{\square} = \max_i \{c_{ij}\} (\text{Tipobenefício})$$

$$r_{ij} = \left(\frac{a_{ij}}{c_j}, \frac{a_{ij}}{b_j}, \frac{a_{ij}}{a_j} \right) e a_j^{\square} = \min_i \{a_{ij}\} (\text{Tipocusto})$$

Uma vez normalizado os dados, estes agora devem ser colocados na proporção de sua importância. Neste ponto, os pesos calculados no método AHP são aplicados a cada valor da Matriz normalizada, gerando uma matriz pesificada, ou com o real valor de cada atributo. Para a pesificação utiliza-se a fórmula: $v_{ij} = r_{ij} * w_{ij}$.

O método FUZZY TOPSIS, baseado na distância entre pontos, necessita encontrar as soluções ótimas positivas e negativas, para calcular as distâncias dos pontos de cada alternativa destas soluções. Para o cálculo da solução ideal positiva (PIS) utiliza-se a formula: $PIS A = (v_1, v_2, \dots, v_n)$ onde $v_j = \max_i \{v_{ij}\}$ e para a solução ideal negativa (NIS) utiliza-se a fórmula: $NIS A = (v_1, v_2, \dots, v_n)$ onde $v_j = \min_i \{v_{ij}\}$.

Duas novas Matrizes são calculadas, a Matriz comparativa das distâncias dos pontos para a (PIS) A^+ , e/ outra para a (NIS) A^- . Observa-se que nesta etapa os valores fuzzy desaparecem, gerando um valor que representa a distância do ponto relativo à alternativa sobre o critério. A fórmula utilizada é a mesma:

$$d^+_{ij} = \sqrt{\left(\frac{1}{3}\right) * \left((v_{ij} - a_j)^2 + (v_{ij+1} - a_{j+1})^2 + (v_{ij+2} - a_{j+2})^2 \right)} \text{ para } A^+$$

$$d^-_{ij} = \sqrt{\left(\frac{1}{3}\right) * \left((v_{ij} - a_j)^2 + (v_{ij+1} - a_{j+1})^2 + (v_{ij+2} - a_{j+2})^2 \right)} \text{ para } A^-$$

As matrizes por si só não geram o ranqueamento. Para calcular as distâncias das alternativas, necessita-se totalizar as distâncias de cada ponto das matrizes de distância d^+ e d^- .

$$S^+ = \sum_{i=1}^i d_i^+ \quad \text{e} \quad S^- = \sum_{i=1}^i d_i^-$$

De posse do total das distâncias, calcula-se o coeficiente de proximidade CC, que permitirá enxergar a classificação das alternativas, para tanto, usa-se a fórmula:

Os valores resultantes são ordenados e geram a classificação para o ranqueamento final das alternativas.

3.7 INSTITUTOS FEDERAIS E UNIVERSIDADES (FIM DA METODOLOGIA – ESTUDO DE CASO)

Até 2019, são registradas mais de 661 unidades de ensino vinculadas a 39 Institutos Federais, também compostas por Centros Federais de Educação Tecnológica (CEFET), a Universidade Tecnológica Federal do Paraná (UTFPR), além de 22 escolas técnicas vinculadas às universidades federais e ao Colégio Pedro II (MEC, 2020). Destas unidades muitas apresentam o curso técnico em informática e a disciplina introdutória de algoritmo e programação de computadores.

Os Institutos Federais são instituições, pluricurriculares e multicampi, composto por: reitoria, campus, campus avançado, polos de inovação e polos de educação a distância, especializados em educação profissional e tecnológica (EPT), também atuam em licenciaturas, bacharelados, pós-graduação stricto sensu e mais recentemente, de doutorados (MEC, 2020).

Os institutos têm como obrigatoriedade legal garantir um mínimo de 50% de suas vagas para a oferta de cursos técnicos de nível médio, com prioridade para o ensino integrado (MEC, 2020).

Participaram do levantamento de critérios e alternativas os docentes do Instituto Federal Fluminense e do Instituto Federal do Espírito Santo, consultados a partir de reuniões do núcleo de informática e de formulário online (Apêndices: 1, 2, 3 e 4). Colaboraram na construção e preenchimento deste instrumento de coleta de dados os professores: Luiz César Ali Novaes Faria (Bacharel e Mestre em Ciências Biológicas, Graduado em Análise e Desenvolvimento de Sistemas, IFFluminense Quissamã), Daniel Vasconcelos (Doutorado em Computação, IFFluminense Quissamã), Orpheu (Tecnólogo em Processamento de Dados, Especialista em Sistemas de Multimídia e Internet, IFFluminense Quissamã) e André Bessa (Analista de Sistemas, Instituto Federal do Espírito Santo de Cachoeiro de Itapemirim) de acordo com suas experiências pessoais e percepções técnicas do problema envolvido.

4 RESULTADOS

4.1 ESTRUTURA RESULTANTE

A diferenciação do primeiro ano do curso técnico dos anos seguintes foi essencial para a delimitação dos critérios. Uma vez que o foco manteve-se sobre a construção da base do conhecimento sobre o pensamento computacional e algoritmos, independentemente das linguagens de produtividade do mercado, valorizou-se a facilidade, simplicidade e solidez na formação inicial.

Os critérios: Legibilidade; Depuração de Erro; Curva de Aprendizagem alta; Ferramentas de escrita de código e Aplicabilidade em projetos reais resultaram de levantamentos de diversos critérios, tanto a partir de pesquisas como a de Parker, Ottaway e Chao (2006), de Prokop *et al.* (2019), quanto da experiência dos professores do núcleo de informática do Instituto Federal Fluminense.

Assim também as alternativas: Linguagem C, JavaScript, studio.code.org, Python, Java, Portugol Webstudio, Scratch e C#, resultaram das linguagens mais utilizadas nos Institutos Federais e Universidades e do senso comum entre os professores do núcleo de informática, principalmente, por conterem nesse núcleo os atores a participarem do processo de ensino-aprendizagem da disciplina de programação inicial com algoritmo.

O plano de aula vigente no Instituto Federal Fluminense de Quissamã foi influenciado pelos resultados deste trabalho. No Apêndice 5 há uma amostragem do 1º bimestre resultante das discussões do núcleo de informática com aulas utilizando os recursos das três primeiras alternativas separadamente.

A estrutura para atender ao objetivo de Definir Critérios e Selecionar Linguagens de Programação Para Alunos Iniciantes em Programação pode ser vista na Figura 57, contendo o objetivo no primeiro nível, os critérios no segundo e as alternativas no terceiro.

4.2 DOS CRITÉRIOS E PESOS

Conforme a Figura 58, os critérios avaliados sob a ótica dos três decisores, segundo a metodologia AHP, resultaram nos pesos para o ranqueamento das

alternativas. Após confirmada a consistência dos julgamentos, calculou-se a partir dos vetores de pesos de cada matriz, um único vetor de pesos para cada critério.

Figura 70 Julgamentos entre critérios dos decisores 1, 2 e 3

	C1	C2	C3	C4	C5	C6		C1	C2	C3	C4	C5	C6
C1	1/1	3/1	1/4	1/1	6/1	4/1]	1/1	3/1	1/2	3/1	3/1	6/1
C2	1/3	1/1	1/4	1/2	3/1	7/1		1/3	1/1	1/6	1/2	3/1	3/1
C3	4/1	4/1	1/1	8/1	6/1	7/1		2/1	6/1	1/1	7/1	7/1	9/1
C4	1/1	2/1	1/8	1/1	3/1	4/1		1/3	2/1	1/7	1/1	1/3	3/1
C5	1/6	1/3	1/6	1/3	1/1	1/1		1/3	1/3	1/7	3/1	1/1	3/1
C6	1/4	1/7	1/7	1/4	1/1	1/1		1/6	1/3	1/9	1/3	1/3	1/1
	C1	C2	C3	C4	C5	C6							
C1	1/1	2/1	1/4	1/1	6/1	4/1]						
C2	1/2	1/1	1/4	1/2	3/1	7/1							
C3	4/1	4/1	1/1	8/1	7/1	8/1							
C4	1/1	2/1	1/8	1/1	3/1	4/1							
C5	1/6	1/3	1/7	1/3	1/1	1/1							
C6	1/4	1/7	1/8	1/4	1/1	1/1							

Os pesos para o ranqueamento foram compostos, segundo a lógica fuzzy de um valor abaixo (l), sendo este o menor valor entre os pesos do critério correspondente, um valor de referência (peso), calculado pela média aritmética dos valores correspondentes e um valor acima (u), o maior valor dentre as avaliações do critério correspondente, conforme a Figura 59.

Figura 71 Geração de pesos para AHP FUZZY TOPSIS

	Decisor 1	Decisor 2	Decisor 3	Média	l	peso	u	dp	dp/f
C1	1,619	1,513	2,08	1,737	1,513	1,737	2,08	0,246	14,17%
C2	0,978	1,046	0,794	0,939	0,794	0,939	1,046	0,107	11,36%
C3	4,185	4,391	4,174	4,25	4,174	4,25	4,391	0,1	2,34%
C4	0,382	0,372	0,723	0,492	0,372	0,492	0,723	0,163	33,18%
C5	0,329	0,322	0,297	0,316	0,297	0,316	0,329	0,014	4,38%

Fonte: próprio autor.

Considerando-se o C3 como o de maior relevância e, com os desvios padrões com baixa variação em relação à média, optou-se por considerar os menores valores observados para (l) e os maiores valores para (u), respectivamente os valores abaixo

e acima do valor de referência para a formação do número fuzzy, a fim de obter uma maior abrangência sobre o coeficiente de pertinência de cada avaliação.

Uma vez terminada a etapa das definições de critérios, alternativas e pesos, fez-se a aplicação do formulário (Apêndice 4). Passou-se o link do formulário para professores e coordenadores responsáveis por ministrar e coordenar a disciplina de programação, nos Institutos Federais. Obteve-se 21 respostas dos formulários.

As respostas do formulário foram armazenadas em formato de expressões linguísticas, conforme o formulário e adequado à lógica fuzzy. Foram copiadas para uma planilha. Utilizou-se de/ outra planilha para converter as expressões linguísticas em valores crisp de acordo com a tabela 6. Os valores dos pesos fuzzy foram calculados pela aplicação do método AHP sobre a avaliação de três decisores, estes são apresentados na figura 59.

Tabela 7 Valores para fuzzificação

Valores para fuzzificação														
Muito Difícil			Difícil			Regular/ Médio			Fácil			Muito Fácil		
0,0	1,0	3,0	1,0	3,0	5,0	3,0	5,0	7,5	5,0	7,5	9,0	7,5	9,0	10,0
	1			2			3			4			5	

Fonte: próprio autor.

A Matriz contendo a resposta dos 21 decisores, foi reduzida para uma Matriz bidimensional critérios x alternativas (Figura 60).

Figura 72 Matriz reduzida de avaliações em critérios por alternativas

	C1			C2			C3			C4			C5		
	l	f	u	l	f	u	l	f	u	l	f	u	l	f	u
A1	0	6	10	0	6	10	1	6	10	1	7	10	1	8	10
A2	0	7	10	0	7	10	3	8	10	1	7	10	0	5	10
A3	0	6	10	0	7	10	0	6	10	0	7	10	5	9	10
A4	0	8	10	0	6	10	3	8	10	1	8	10	0	5	10
A5	0	6	10	0	6	10	3	7	10	1	7	10	3	8	10
A6	0	7	10	0	7	10	3	8	10	1	8	10	0	4	10
A7	0	7	10	0	7	10	1	7	10	1	7	10	5	9	10
A8	0	7	10	0	7	10	1	6	10	1	7	10	1	8	10

Fonte: próprio autor.

A Matriz reduzida foi normalizada, de acordo com a metodologia AHP FUZZY TOPSIS resultando na Matriz normalizada (Figura 61) para gerar intervalos de 0 a 1.

Figura 73 Matriz Normalizada

	C1	C2	C3	C4	C5										
A1	0	1	1	0	1	1	0	1	0	0	1	1	0	1	1
A2	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1
A3	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1
A4	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1
A5	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1
A6	0	1	1	0	1	1	0	1	1	0	1	1	0	0	1
A7	0	1	1	0	1	1	0	1	1	0	1	1	1	1	1
A8	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1

Fonte: próprio autor.

Uma vez normalizada, a Matriz teve calculada a importância de cada avaliação proporcionalmente aos pesos de cada critério, conforme figuras 62 e 63.

Figura 74 Matriz de pesos consistentes (AHP)

pesos	1,51	1,74	2,08	0,79	0,94	1,05	4,17	4,25	4,39	0,37	0,49	0,72	0,3	0,32	0,33
-------	------	------	------	------	------	------	------	------	------	------	------	------	-----	------	------

Fonte: próprio autor.

Figura 75 Matriz pesificada

	C1	C2	C3	C4	C5										
A1	0	1,01	2,08	0	0,51	1,05	0,42	2,44	0	0,04	0,33	0,72	0,03	0,24	0,33
A2	0	1,29	2,08	0	0,69	1,05	1,25	3,44	4,39	0,72	0,34	0,72	0	0,17	0,33
A3	0	1,11	2,08	0	0,65	1,05	0	2,43	4,39	0	0,34	0,72	0,15	0,27	0,33
A4	0	1,29	2,08	0	0,6	1,05	1,25	3,56	4,39	0,04	0,37	0,72	0	0,16	0,33
A5	0	1,06	2,08	0	0,54	1,05	1,25	3,08	4,39	0,04	0,32	0,72	0,09	0,26	0,33
A6	0	1,26	2,08	0	0,69	1,05	1,25	3,5	4,39	0,04	0,39	0,72	0	0,14	0,33
A7	0	1,17	2,08	0	0,64	1,05	0,42	3,02	4,39	0,04	0,35	0,72	0,15	0,28	0,33
A8	0	1,17	2,08	0	0,63	1,05	0,42	2,72	4,39	0,04	0,32	0,72	0,03	0,26	0,33

Fonte: próprio autor.

Soluções ótimas positivas (PIS) e negativas (NIS) foram encontradas, figura 64.

Figura 76 Matriz com soluções ótimas positivas (PIS) e negativas (NIS)

A+	0	1,29	2,08	0	0,69	1,05	1,25	3,56	4,39	0,04	0,39	0,72	0,15	0,28	0,33
A-	0	1,01	2,08	0	0,51	1,05	0	2,43	0	0	0,32	0,72	0	0,14	0,33

Fonte: próprio autor.

O método AHP FUZZY TOPSIS baseia-se na ideia de visualizar as pontuações como pontos cartesianos. Criou-se duas Matrizes, uma com as distâncias positivas (d+), contendo as distâncias de cada ponto em relação a A+ na figura 65 e/ outra com as distâncias negativas (d-), contendo as distâncias de cada ponto em relação a A- na figura 65.

Figura 77 Matriz com as distâncias dos pontos para PIS

	C1	C2	C3	C4	C5
A1	0,16	0,1	2,66	0,03	0,07
A2	0	0	0,07	0,03	0,11
A3	0,11	0,02	0,98	0,04	0
d+ A4	0	0,05	0	0,01	0,11
A5	0,14	0,09	0,28	0,04	0,04
A6	0,02	0	0,04	0	0,12
A7	0,07	0,03	0,58	0,02	0
A8	0,07	0,04	0,68	0,04	0,07

Fonte: próprio autor.

As distâncias em cada critério foram somadas para gerar uma única distância da alternativa, tanto para as distâncias positivas em S+, quanto para as distâncias negativas em S-. Criou-se assim as distâncias que permitiram calcular o coeficiente de proximidade de cada alternativa. Este coeficiente foi calculado pela dividindo-se S- pelo somatório de S- com S+ na figura 67.

Figura 78 Matriz com as distâncias dos pontos para NIS

	C1	C2	C3	C4	C5
A1	0	0	0,24	0,02	0,06
A2	0,16	0,1	2,7	0,02	0,02
A3	0,06	0,08	2,54	0,01	0,12
A4	0,16	0,05	2,72	0,04	0,01
A5	0,03	0,01	2,66	0,02	0,09
A6	0,15	0,1	2,71	0,04	0
A7	0,09	0,07	2,57	0,03	0,12
A8	0,09	0,06	2,55	0,02	0,08

Fonte: próprio autor.

O último procedimento consiste na ordenação sobre os coeficientes de proximidade, apontando em ordem crescente, do melhor para o pior, ou o primeiro lugar para o de maior pontuação. O ranque resultante em ordem de melhor para pior foi: 1º Portugol WebStudio; 2º Scratch; 3º studio.code.org; 4º JavaScript; 5º Python; 6º C#; 7º Java e 8º Linguagem C.

Figura 79 Coeficiente de proximidade e Ordenação de resultados

	S-	S+	S- / (S- + S+)	Ordenação
Linguagem C	0,33	3,03	0,1	8
studio.code.org	3	0,21	0,93	3
Java	2,8	1,14	0,71	7
Scratch	2,98	0,17	0,95	2
JavaScript	2,82	0,57	0,83	4
Portugol WebStudio	3	0,17	0,95	1
Python	2,88	0,7	0,8	5
C#	2,8	0,9	0,76	6

Fonte: próprio autor.

4.3 DISCUSSÃO

Alunos expostos a experiência de aprendizagem direta de algoritmo a partir de uma linguagem e de aprender apenas um algoritmo apresentam, em ambos os casos, insuficiência de habilidade para desenvolver algoritmos. As opiniões dos alunos

sugerem haver menos dificuldade no planejamento e solução de algoritmos diretamente do que no uso de linguagens de programação (KORKMAZ, 2013).

A realidade de cada instituto de ensino pode interferir no processo de ensino da disciplina introdutória de programação de computadores, como: a boa ou má qualidade da formação pregressa do aluno, a quantidade e qualidade da estrutura de laboratórios do campus, a compatibilidade de carga horária disponível com a proposta, dentre/ outros fatores. A definição dos recursos educacionais capazes de intensificar a melhora no ensino desta disciplina dependerá do bom senso do pesquisador em identificar o problema, as variáveis e a escolha do método para buscar soluções.

Ainda sobre a complexidade, o caráter objetivo que possa ser quantificado em valores, é muitas vezes substituído pelo caráter subjetivo, como por exemplo: a beleza, a simpatia, percepções sobre as emoções, enfim, critérios que dependem de nossas percepções subjetivas (discussão).

Cognitivamente, não há evidências conclusivas da existência de aptidão dos indivíduos sobre a habilidade de programar. O fato de ter experiência ajuda, mas fatores como a motivação e o estilo de aprendizagem inadequados ao perfil individual dos alunos podem levá-los ao insucesso (Jenkins, 2002).

O aumento da aprendizagem sobre algoritmos está mais relacionada ao nível de envolvimento e interesse dos alunos do que sobre os métodos e tecnologias utilizados. O resultado da aprendizagem aumenta quando os alunos não ficam limitados as visualizações, quando aplicam na prática as atividades sobre os algoritmos visualizados (GRISSOM; MCNALLY; NAPS, 2003).

Os próximos anos do curso técnico abordam uma programação mais complexa e mais sofisticada, que envolve a utilização de inúmeros recursos inicialmente não abordados a fim de facilitar a aprendizagem da base conhecida como programação estruturada.

Para esta formação inicial não há necessidade de trabalhar conhecimentos sobre programação orientada a objetos (POO), bem como conhecimentos sobre redes e banco de dados.

A POO constitui um paradigma de programação baseado fundamentalmente no conceito de Objeto, que passa a existir a partir de instâncias de classes, que, por sua vez, são formadas por atributos e/ ou métodos (REF). Assim também, as redes e os bancos de dados podem ser vistos como objetos e ter tratamento similar.

A utilização de frameworks poderia acelerar a produção de aplicativos; entretanto, não promoveria o aprendizado suficiente para que fossem feitas manutenções e melhoramentos posteriores nestes, pois, entregariam o código pronto, parcialmente já desenvolvido, pulando etapas de construção do conhecimento.

Aplicar diferentes disciplinas a diferentes turmas simultâneas, contendo os mesmos objetivos, planos de aula e mesmos instrumentos avaliativos, poderiam nos servir para validar de forma científica a relação de ensino aprendizagem sobre cada disciplina, entretanto, não é objeto desta dissertação.

O objetivo final do curso se realiza na formação de um profissional capaz de produzir aplicativos reais e atender as demandas de programação para o mercado, no entanto, nesta etapa, o imprescindível está na preparação do discente, habilitando-o a absorver tecnologias nos anos seguintes e não necessariamente na produção real.

A título de reflexão, caso a pesquisa fosse para as turmas de 2° e 3° anos, os fatores a serem observados, além da confiabilidade, multiplataforma e orientação a objetos, poderiam ser: penetração de mercado, funcionalidade, open source, licenciamento, escalabilidade, performance, segurança e tecnologias.

5 CONCLUSÕES

A pesquisa atendeu seu objetivo de identificar e ranquear as principais linguagens e/ ou tecnologias para auxílio no ensino de algoritmo, a saber Portugol Webstudio e primeiro lugar, em segundo Scratch e studio.code.org em terceiro. O que tornou possível ranquear as melhores linguagens e/ou ferramentas didáticas foram as adequações aos critérios utilizados para se julgar as alternativas, estas por sua vez, devem em primeiro lugar ceder lugar ao aprendizado em si e não a eficiência da linguagem. A partir da opinião de 21 professores dos Institutos Federais, considerou-se, segunda a metodologia AHP FUZZY TOPSIS aplicada sobre as alternativas, a luz dos critérios.

Ao final do trabalho, após a geração do ranking, fora aplicada a metodologia AHP Borda sobre os julgamentos dos professores. Apenas o primeiro lugar se manteve. Os percentuais foram muito próximos conforme segue:

Mostrou-se essencial levar em consideração as distâncias médias de cada julgamento em relação ao PIS e ao NIS para maior precisão da análise, bem como, ficou evidente que o método FUZZY TOPSIS foi mais adequado do que o AHP Borda.

Por decisão democrática em conjunto a estes, apresentado os resultados desta pesquisa, definiu-se o Portugol WebStudio como sendo a melhor estratégia de implantação deste conhecimento, considerando o volume de conhecimento e o tempo necessário para aplicação deste.

Este trabalho foi realizado sob a premissa de que algoritmos e linguagens de programação devem ser ensinados em paralelo, assim também, considera que o aluno não deve ter conhecimento prévio a respeito da disciplina e principalmente que conhecer e dominar os preceitos da programação podem e devem ser ensinados na língua materna do aluno, preferencialmente com o uso de recursos gráficos, sem necessariamente associar este aprendizado a uma linguagem Orientada a Objetos e/ ou com manipulação de banco de dados.

O Ranking final foi:

1º Portugol WebStudio, 2º Scratch, 3º studio.code.org, 4º JavaScript, 5º Python , 6º C#, 7º Java, 8º Linguagem C.

Observa-se ainda que a utilização dos três primeiros lugares não são excludentes entre si, dotados de facilidades que permite seu uso simultâneo em partes das etapas de ensino, ratificando as respostas da pesquisa.

Por decisão conjunta ao núcleo de informática do Instituto Federal de Quissamã, apresentado os resultados desta pesquisa, definiu-se o Portugal WebStudio como sendo a melhor estratégia de implantação deste conhecimento, considerando o volume de conhecimento e o tempo necessário para aplicação deste, com conteúdos parciais em paralelo utilizando-se o Scratch e o studio.code.org, segue exemplo do plano de aula do primeiro bimestre resultante, no Apêndice 5.

Como consequência de se dividir o ensino de programação em dois momentos, fundamentado na não obrigatoriedade dos alunos do 1º ano do curso técnico de informática, de possuir conhecimento prévio em nenhuma linguagem de programação, ou algoritmo, bem como a despreocupação com o momento seguinte quanto à continuidade da linguagem e aprofundamento de temas com programação orientada a objetos, redes e banco de dados, permitiu uma análise mais voltada para o ensino do pensamento computacional inicial.

Como resultado, obteve-se uma maior percepção sobre os critérios específicos sobre a finalidade do trabalho, e o ranqueamento de algumas linguagens de programação para servir de base na formação do PPC do curso técnico de informática do Instituto Federal de Quissamã.

REFERÊNCIAS

- AGUADO, A. G.; CANTANHEDE, M. A. **Lógica fuzzy**. 2010. Disponível em: <https://docplayer.com.br/32988870-Logica-fuzzy-alexandre-garcia-aguado-marco-andre-cantanhede.html>. Acesso em: 12 maio 2020.
- ARKSEY, H.; O'MALLEY, L. Scoping studies: towards a methodological framework. **International journal of social research methodology**, [s.l.], v. 8, n. 1, p. 19-32, 2005.
- BELÉM, P. H. A.; MEZA, E. B. M.; VIANNA, D. S. Escolha de um Framework para a linguagem de programação PHP através do método AHP clássico. *In*: CONGRESSO NACIONAL DE EXCELÊNCIA EM GESTÃO, 10., 2014. **Anais** [...]. Rio de Janeiro: Universidade Federal Fluminense, 2014. p. 33.
- BURD, B. **Java para Leigos**. Rio de Janeiro: Alta Books, 2013.
- CALACHE, L. D. D. R. *et al.* Proposta de um modelo de avaliação e de seleção de fornecedores de manutenção industrial utilizando Fuzzy-TOPSIS. **Gestão & Produção**, São Carlos, v. 26, n. 2, p. 2- 13, 2019.
- CHA, Y.; YUNG, M. Satisfaction assessment of multi-objective schedules using neural fuzzy methodology. **International journal of production research**, [s.l.], v. 41, n. 8, p. 1831-1849, 2003.
- CHEN, C. T. Extensions of the TOPSIS for group decision-making under fuzzy environment. **Fuzzy sets and systems**, [s.l.], v. 114, n. 1, p. 1-9, 2000.
- CHEN, S. J.; HWANG, C. L. **Fuzzy multiple attribute decision making methods**. Fuzzy multiple attribute decision making. Berlin: Springer, 1992. p. 289-486.
- CHU, T. C. Facility location selection using fuzzy TOPSIS under group decisions. **International journal of uncertainty, fuzziness and knowledge-based systems**, [s.l.], v. 10, n. 6, p. 687-701, 2002.
- CHU, T. C.; LIN, Y. C. A fuzzy TOPSIS method for robot selection. **The International Journal of Advanced Manufacturing Technology**, [s.l.], v. 21, n. 4, p. 284-290, 2003.
- CODE.ORG® - CODE. Disponível em: <https://studio.code.org/courses>. Acesso em: 01 jul. 2020.
- COSTA, H. G. **Auxílio multicritério à decisão: método AHP**. Rio de Janeiro: Abepro, 2006.
- COSTA, H. G. **Introdução ao método de análise hierárquica: análise multicritério no auxílio à decisão**. Niterói, RJ: HGC, 2002.

CRAIK, F. I. M.; LOCKHART, R. S. Levels of processing: A framework for memory research. **Journal of verbal learning and verbal behavior**, [s.l.], v. 11, n. 6, p. 671-684, 1972.

DELGADO, C. *et al.* Identificando competências associadas ao aprendizado de leitura e construção de algoritmos. *In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO*, 13., 2005, São Leopoldo. **Anais [...]** São Leopoldo: UNISINOS, 2005.

DOLAN, J. G. Shared decision-making—transferring research into practice: the Analytic Hierarchy Process (AHP). **Patient education and counseling**, [s.l.], v. 73, n. 3, p. 418-425, 2008.

FERNANDES, J. M.; MACHADO, R. J. **Requisitos em Projetos de Software e de Sistemas de Informação**. São Paulo: Novatec, 2018.

FREIRE, F. M. P.; PRADO, M. E. B. B. Projeto Pedagógico: Pano de fundo para escolha de um software educacional. *In: VALENTE, J. A. (org.). O computador na sociedade do conhecimento*. São Paulo: OEA_NIED/UNICAMP, 1999.

GOMES, L. F. A. M.; GONZÁLEZ, M. C. A.; CARIGNANO, C. **Tomada de decisões em cenários complexos: introdução aos métodos discretos do apoio multicritério à decisão**. São Paulo: Thomson, 2004. *E-book*.

GOMIDE, F.; GUDWIN, R. R.; TANSCHKEIT, R. Conceitos fundamentais da teoria de conjuntos fuzzy, lógica fuzzy e aplicações. *In: IFSA CONGRESS-TUTORIALS*, 1., 1995, Campinas. **Proceedings [...]** Campinas: Unicamp, 1995. p. 1-38.

GOMIDE, F. A. C.; DIDWIN, R. R. Modelagem, controle, sistemas e lógica fuzzy. **Revista da SBA controle & Automação**, [s.l.], v. 4, n. 3, p. 97-115, 1994.

GRISSOM, S.; MCNALLY, M.; NAPS, T. L. Algorithm visualization in CS education: comparing levels of student engagement. *In: ACM SYMPOSIUM ON SOFTWAREVISUALIZATION*, 1., 2003, New York. **Proceedings [...]** New York, 2003. p. 87-94.

GUDWIN, R. **Linguagens de programação**. Campinas, SP: DCA/FEEC/UNICAMP, 1997. p. 24.

GUGLIELMETTI, F.; MARINS, F.; SALOMON, V. Comparação teórica entre métodos de auxílio à tomada de decisão por múltiplos critérios. *In: ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO*, 23., 2003, Ouro Preto. **Anais [...]** Ouro Preto: UFOP, 2003.

JAHANSHALOO, G. R.; HOSSEINZADEH, L. F.; IZADIKHAH, M. An algorithmic method to extend TOPSIS for decision-making problems with interval data. **Applied mathematics and computation**, [s.l.], v. 175, n. 2, p. 1375-1384, 2006.

JENKINS, T. On the difficulty of learning to program. *In: ANNUAL CONFERENCE OF THE LTSN CENTRE FOR INFORMATION AND COMPUTER SCIENCES, 3., 2002* United Kingdom. **Proceedings** [...] United Kingdom: University of Leeds, 2002. p. 53-58.

KAHRAMAN, C.; GÜLBAY, M.; ULUKAN, Z. Applications of fuzzy capital budgeting techniques. *In: KAHRAMAN, C. Fuzzy applications in industrial engineering.* Berlin: Heidelberg, 2006. p. 177-203.

KAHRAMAN, C. **Fuzzy multi-criteria decision making: theory and applications with recent developments.** 2008. Disponível em: <https://books.google.com.br>. Acesso em: 01 jul. 2020.

KANNAN, D.; KHODAVERDI, R.; OLFAT, L.; JAFARIAN, A. Integrated fuzzy multi criteria decision making method and multi-objective programming approach for supplier selection and order allocation in a green supply chain. **Journal of Cleaner production**, [s.l.], v. 47, p. 355-367, 2013.

KORE, M. N. B.; RAVI, K.; PATIL, A. P. M. S. A simplified description of fuzzy TOPSIS method for multi criteria decision making. **International Research Journal of Engineering and Technology**, [s.l.], v. 4, n. 5, p. 2047-2050, 2017.

KORKMAZ, O. Students' difficulties in and opinions about designing algorithms according to different instructional applications. **Energy education science and technology Part B: Social and Educational Studies**, [s.l.], v. 5, n. 1, p. 209-218, 2013.

LIANG, G. S. Fuzzy MCDM based on ideal and anti-ideal concepts. **European Journal of Operational Research**, [s.l.], v. 112, n. 3, p. 682-691, 1999.

LIMA JUNIOR, F. R.; CARPINETTI, L. C. R. A comparison between Fuzzy AHP and Fuzzy TOPSIS methods to supplier selection. **Applied Soft Computing, Gestão & Produção**, [s.l.], v. 21, p. 194-209, 2014.

MADI, E. N.; TAP, A. O. Fuzzy TOPSIS method in the selection of investment boards by incorporating operational risks. *In: WORLD CONGRESS ON ENGINEERING, 1, 2011, London.* **Proceedings** [...] London: U.K, 2011. p. 291-295.

MADI, E. N.; GARIBALDI, J. M.; WAGER, C. A comparison between two types of Fuzzy TOPSIS Method. *In: IEEE International Conference on Systems, Man, and Cybernetics, 1., 2015. USA.* **Proceedings** [...] Piscataway, NJ, USA: IEEE, 2015. p. 291-297.

MARJI, M. **Aprenda a Programar com Scratch - Uma introdução visual à programação com jogos, arte, ciência e matemática.** São Paulo: Novatec, 2014.

MARRO, A. A.; SOUZA, A. D. C.; CAVALCANTE, E. D. S.; BEZERRA, G. S.; NUNES, R. **Lógica fuzzy: conceitos e aplicações.** Natal: Universidade Federal do Rio Grande do Norte (UFRN), 2010.

- MARTINS, L. de A. Uso do método de análise hierárquica (AHP) na tomada de decisão na implementação de energias renováveis em plataforma de petróleo offshore. *In: SBPO – SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL*, 41., 2009, Porto Seguro. **Anais [...]** Porto Seguro: UFG, 2009.
- MEC - MINISTÉRIO DA EDUCAÇÃO. **Pesquisa sobre Instituições da Rede Federal**. 2020. Disponível em: <http://portal.mec.gov.br/rede-federal-inicial/instituicoes> Acesso em: 25 jun. 2020.
- MEDINA, M.; FERTING, C. **Algoritmos e programação: teoria e prática**. São Paulo: Novatec Editora, 2006.
- MILETTO, E. M.; BERTAGNOLLI, S. C. **Desenvolvimento de Software II: Introdução ao Desenvolvimento Web com HTML, CSS, JavaScript e PHP-Eixo: Informação e Comunicação-Série Tekne**. São Paulo: Bookman Editora, 2014.
- MINSKY, M. A framework for representing knowledge. *In: WINSTON, P. H. The psychology of computer vision*. The society of mind. New York: Simon & Schuster, 2019.
- MORAES, E. A.; SANTALIESTRA, R. Modelo de decisão com múltiplos critérios para escolha de software de código aberto e software de código fechado. **Revista Organizações em Contexto (Online)**, [s.l.], ano 4, n. 7, p. 59-83, 2008.
- MORAN, J. M. Como utilizar a Internet na educação. **Ciência da informação**, [s.l.], v. 26, n. 2, p. 2-10, 1997.
- MUELLER, J. P. **Começando a Programar em Python Para Leigos**. São Paulo: Alta Books 2016.
- NĂDĂBAN, S.; DZITAC, S.; DZITAC, I. Fuzzy TOPSIS: a general view. **Procedia Computer Science**, [s.l.], v. 91, p. 823-831, 2016.
- OLIVEIRA, Mariângela Dutra de *et al.* Nova abordagem do índice de qualidade de água bruta utilizando a lógica fuzzy. **Engenharia Sanitária e Ambiental**, [s.l.], v. 19, n. 4, p. 361-372, 2014.
- PACHECO, M. C. R.; GOLDMAN, F. L. O AHP como um modelo matemático: uma análise de sensibilidade simples. *In: SIMPÓSIO DE EXCELÊNCIA EM GESTÃO E TECNOLOGIA - “DESENVOLVIMENTO DE COMPETÊNCIAS FRENTE AOS DESAFIOS DO AMANHÃ”*, 1., 2016. Rio de Janeiro. **Anais [...]** Rio de Janeiro: Faculdades Dom Bosco, 2016.
- PARKER, K. R.; CHAO, J. T.; OTTAWAY, T. A.; CHANG, J. A formal language selection process for introductory programming courses. **Journal of Information Technology Education: Research**, [s.l.], v. 5, n. 1, p. 133-151, 2006.
- PARKER, K. R.; OTTAWAY, T. A.; CHAO, J. T. Criteria for the selection of a programming language for introductory courses. **International Journal of Knowledge and Learning**, [s.l.], v. 2, n. 1-2, p. 119-139, 2006.

PREZOTTO, E. D.; BONIATI, B. B. **Estudo de frameworks multiplataforma para desenvolvimento de aplicações mobile híbridas**. 2014. 30f. Trabalho de Conclusão de Curso (Tecnologia em Sistemas para Internet) - Universidade Federal de Santa Maria, Santa Maria, 2014.

PROCTOR, C.; BLIKSTEIN, P. How broad is computational thinking? A longitudinal study of practices shaping learning in computer science. *In*: KAY, J.; LUCKIN, R. (Eds.). **Rethinking Learning in the Digital Age: Making the Learning Sciences Count**, London, UK: International Society of the Learning Sciences, 2018.

PROKOP, Y., TROFIMENKO, E.; SEVERIN, N.; BUKATA, N. **An Analysis of Criteria for Choosing a First Programming Language in Universities**. 2019. Disponível em: 10.22152/programming-journal. Acesso em: 12 jul. 2020.

RAPKIEWICZ, C. E. *et al.* Estratégias pedagógicas no ensino de algoritmos e programação associadas ao uso de jogos educacionais. **RENOTE-Revista Novas Tecnologias na Educação**, [s.l.], v. 4, n. 2, p. 1-11, 2006.

SAATY, T. L. **Fundamentals of decision making and priority theory with the analytic hierarchy process**. New York: RWS publications, 2000.

SAATY, T. L. How to make a decision: the analytic hierarchy process. **European journal of operational research**, [s.l.], v. 48, n. 1, p. 9-26, 1990.

SAATY, T. L. How to make a decision: the analytic hierarchy process. **Interfaces**, [s.l.], v. 24, n. 6, p. 19-43, 1994.

SEBESTA, R. W. **Conceitos de Linguagens de Programação-11**. São Paulo: Bookman Editora, 2018.

SMOLARSKI, D. C. A first course in computer science: Languages and goals. **Teaching Mathematics and Computer Science**, [s.l.], v. 1, n. 1, p. 137-152, 2003.

SOARES, M. S. Comparação entre metodologias Ágeis e tradicionais para o desenvolvimento de software. **INFOCOMP Journal of Computer Science**, [s.l.], v. 3, n. 2, p. 8-13, 2004.

SOUSA, F. R. C.; MOREIRA, L. O.; MACHADO, J. C. Computação em nuvem: Conceitos, tecnologias, aplicações e desafios. *In*: SANTOS NETO, P A. **II Escola Regional de Computação Ceará, Maranhão e Piauí (ERCEMAPI)**. Ceará: UFC, 2009. p. 150-175. Disponível em: https://www.researchgate.net/profile/Javam-Machado/publication/237644729_Computacao_em_Nuvem_Conceitos_Tecnologias_Aplicacoes_e_Desafios/links/56044f4308aea25fce3121f3/Computacao-em-Nuvem-Conceitos-Tecnologias-Aplicacoes-e-Desafios.pdf. Acesso em: 01 jul. 2020.

STEFFENS, A. **PORTUGOL STUDIO**. 2020. Disponível em: <http://lite.acad.univali.br/portugol/>. Acesso em: 01 jul. 2020.

TAYLAN, O.; ALIDRISI, H.; KABLI, M. A multi-criteria decision-making approach that combines fuzzy topsis and DEA methodologies. **South African Journal of Industrial Engineering**, [s.l.], v. 25, n. 3, p. 39-56, 2014.

THIRY, M.; ZOUCAS, M.; GONÇALVES, A. R. Q. Promovendo a Aprendizagem de Engenharia de Requisitos de Software através de um Jogo Educativo. *In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO-SBIE*, 1., 2010, São José. **Anais [...]** São José: Universidade do Vale do Itajaí (UNIVALI), 2010.

TUDO CELULAR LTDA. **Pesquisa sobre valores de preço, câmera em MP e tempo de bateria de celulares**. 2018. Disponível em: <http://www.tudocelular.com/compare>. Acesso em: 01 jul. 2020.

TSAUR, S. H.; CHANG, T. Y.; YEN, C. H. The evaluation of airline service quality by fuzzy MCDM. **Tourism management**, [s.l.], v. 23, n. 2, p. 107-115, 2002.

VÉGH, L.; STOFFOVÁ, V. Algorithm animations for teaching and learning the main ideas of basic sortings. **Informatics in Education**, [s.l.], v. 16, n. 1, p. 121-140, 2017.

YANG, T.; HUNG, C. C. Multi-attribute decision making methods for plant layout design problem. **Robotics Comput. Integrated Manuf.**, [s.l.], v. 23, p. 126-137, 2007.

YUEN, K. K. F. Combining compound linguistic ordinal scale and cognitive pairwise comparison in the rectified fuzzy TOPSIS method for group decision making. **Fuzzy Optimization and Decision Making**, [s.l.], v. 13, n. 1, p. 105-130, 2014.

ZADEH, L. A. Fuzzy logic - a personal perspective. **Fuzzy sets and systems**, [s.l.], v. 281, p. 4-20, 2015.

ZANETTI, H.; BORGES, M.; RICARTE, I. Pensamento computacional no ensino de programação: Uma revisão sistemática da literatura brasileira. *In: BRAZILIAN SYMPOSIUM ON COMPUTERS IN EDUCATION*, 1., 2016, Rio de Janeiro. **Anais [...]** Rio de Janeiro: SBIE 2016. p. 21.

ZHANG, G.; LU, J. An integrated group decision-making method dealing with fuzzy preferences for alternatives and individual judgments for selection criteria. **Group Decision and Negotiation**, [s.l.], v. 12, n. 6, p. 501-515, 2003.

WANG, Y. M.; ELHAG, T. M. S. Fuzzy TOPSIS method based on alpha level sets with an application to bridge risk assessment. **Expert systems with applications**, [s.l.], v. 31, n. 2, p. 309-319, 2006.

WIND, Y.; SAATY, T. L. Marketing applications of the analytic hierarchy process. **Management science**, [s.l.], v. 26, n. 7, p. 641-658, 1980.

Apêndice 1 - Formulário de Levantamento de Sugestões

O público iniciante na aprendizagem de algoritmo e linguagem de programação do curso técnico em informática do IFF nos é conhecido, bem como suas dificuldades e demandas.

Este formulário tem por objetivo identificar alternativas de linguagens de programação e critérios para sua escolha, a serem utilizadas em dissertação de mestrado.

Solicito que preencham com as sugestões de linguagens para o primeiro ano e com os critérios que julgam importantes para a escolha destas linguagens

Obrigado!

Linguagens

Crítérios

Apêndice 2 Formulário sobre levantamento de alternativas e de critérios

Escolha de linguagens para 1º ano

Escolha de melhores linguagens para ensino e aplicação de algoritmo no 1º ano do curso técnico de informática.

Linguagens/recursos que poderiam ou deveriam ser escolhidas para serem ensinadas no primeiro ano, afim de potencializar a aprendizagem de algoritmos e introdução a programação no 1º ano de informática?

- C
- Java
- [Studio.code.org](https://studio.code.org) ou
- [Studio.cod.org](https://studio.cod.org) + Scratch
- [Studio.code.org](https://studio.code.org) + Scratch + Portugol Webstudio
- Scratch
- Portugol WebStudio
- HTML/CSS/JS
- JavaScript
- Python
- C#
- C++

Informe alguma(s) linguagem(ns), recurso(s) ou tecnologia(s) que considera essencial que não está listada acima.

Texto de resposta longa

Seção 2 de 2

Levantamento de Critérios para seleção de Linguagens de apoio ao ensino de algoritmo



Liste os critérios para que sejam pontuadas e ranqueadas as alternativas a luz destes critérios

Informe os critérios que julgar necessários para a seleção das linguagens

Texto de resposta longa

Apêndice 3 Formulário planilha de comparação de importância entre critérios

COMPARAÇÃO DE IMPORTÂNCIA ENTRE CRITÉRIOS

Informe em uma das colunas o grau de importância de cada critério em relação ao outro

Número	Termo
1	equivalente / igual importância
3	pouco mais importante
5	maior importância
7	bem mais importante
9	Muito mais importante - essencial
2,4,6,8	Valores intermediários de importância

Legibilidade		Depuração de Erro
Legibilidade		Curva de Aprendizagem alta
Legibilidade		Disponibilidade de Material Gratuito
Legibilidade		Ferramentas de escrita de código
Legibilidade		Aplicabilidade em projetos reais
Depuração de Erro		Curva de Aprendizagem alta
Depuração de Erro		Disponibilidade de Material Gratuito
Depuração de Erro		Ferramentas de escrita de código
Depuração de Erro		Aplicabilidade em projetos reais
Curva de Aprendizagem alta		Disponibilidade de Material Gratuito
Curva de Aprendizagem alta		Ferramentas de escrita de código
Curva de Aprendizagem alta		Aplicabilidade em projetos reais
Disponibilidade de Material Gratuito		Ferramentas de escrita de código
Disponibilidade de Material Gratuito		Aplicabilidade em projetos reais
Ferramentas de escrita de código		Aplicabilidade em projetos reais

Caso o valor seja diferente de 1, deverá ser colocado ao lado do critério mais importante.

Caso o valor seja igual a 1, poderá ser colocado ao lado de qualquer dos dois critérios

Escala Verbal	Valor de Julgamento	Valor Recíproco	Explicação
Igual Importância	1	1	Os critérios contribuem igualmente para o objeto
Pouca importância de uma sobre a outra	3	1/3	Um critério é levemente mais importante que o outro
Importância grande	5	1/5	Um critério é mais importante que o outro
Importância muito grande	7	1/7	Um critério é predominante sobre o outro
Importância extrema ou absoluta	9	1/9	Um critério é absolutamente predominante
Valores Intermediários	2, 4, 6, 8	1/2, 1/4, 1/6, 1/8,	Critérios utilizados para fazer ajustes entre as preferências, ou quando se procura um julgamento intermediária na escala

Apêndice 5 Plano de aula - 1º Bimestre

Aula	Conteúdo	Descrição
1	Apresentação e Integração	Atividades sociais de integração e apresentação trabalhos de alunos dos anos anteriores
2	"Apresentação / contextualização desta (avaliação com peso 1,3,5,7 para prova por bimestre + atividades com pesos variáveis)"	Apresentação de laboratórios, objetivos, metodologia, formas avaliativas, Calendário letivo, login e senha, formação de grupo whatsapp, Pasta de aulas e de trabalhos compartilhada (repositório comum)
3	Instrução – dinâmica com “baralho / organizar”; receita de bolo; direções	Atividades lúdicas na criação e execução de algoritmos utilizando os próprios alunos e os recursos de uma sala de aula padrão
4	code.org: Sequências (curso 1 [4-6 anos]: Fases 3 a 7 e 16)	Desenvolver a percepção de uma sequência de comandos e criação de histórias digitais
5	Csu – corrida (formato de gincana valendo serenata de amor para ganhadores) Atividade em grupo {Integração com Geografia e Educação Física}	Atividade lúdica com integração de conteúdos interdisciplinares e despertar para empresas de startup e soluções de ti
6	code.org: Sequências (curso 2 [>6 anos]: Fases 3 e 4)	Reforço do conceito de sequência de comandos
7	CSU de desenho (conceito de LP e compilação) (CSU pág. 109)	Atividades lúdicas para aproximação e conceitos
8	Scratch (Movimento / Aparência) - Criar uma Animação {Integração com Filosofia}	Introdução do ambiente de programação em blocos com scratch, ainda com sequência de comandos, fazendo animações com conteúdo integrado à filosofia
9	Portugol - escreva() e operadores aritméticos	Introdução à programação em linhas de comando com o portugol utilizando sequências de comandos e operações matemáticas, conceito de variáveis
10	Scratch (Variáveis e Cálculo)	Trabalhando com variáveis e operações matemáticas
11	Portugol - tipos de dados, variáveis e leia()	Conceitos de tipos de dados, recebendo valores do teclado
12	Portugol - resolução de exercícios {Integração com Matemática}	Exercícios com operações matemáticas integrados à disciplina de matemática
13	Scratch - variáveis locais e globais; Portugol - Variáveis locais e globais	Expandindo o conceito de variáveis, exercícios de fixação
14	Revisão para a P1	Revisão de conteúdos práticos e teóricos para avaliação
15	Avaliação prova (31/03)	Avaliação escrita

16	Correção da prova	Correção de avaliação com alunos para dissolução de dúvidas e fixação de conteúdos
17	code.org (curso 2 [>6 anos]: Fases 6, 7 e 8)	Introdução dos conceitos de controles de repetição (laços)
18	Scratch: Laços e controles (introdução a condicionais)	Utilização de laços no scratch e introdução de controles condicionais
19	Atividade Letiva dos Povos Originários	Exercícios de fixação
20	Avaliação	Avaliação bimestral

D 14	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10			
D 15	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	3,0	5,0	7,5	5,0	7,5	9,0	5,0	7,5	9,0	7,5	9,0	10	7,5	9,0	10
D 16	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10
D 17	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	1,0	3,0	5,0	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10
D 18	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10
D 19	0,0	1,0	3,0	0,0	1,0	3,0	0,0	1,0	3,0	0,0	1,0	3,0	0,0	1,0	3,0	0,0	1,0	3,0	0,0	1,0	3,0	0,0	1,0	3,0
D 20	1,0	3,0	5,0	1,0	3,0	5,0	1,0	3,0	5,0	1,0	3,0	5,0	1,0	3,0	5,0	1,0	3,0	5,0	1,0	3,0	5,0	1,0	3,0	5,0
D 21	3,0	5,0	7,5	3,0	5,0	7,5	3,0	5,0	7,5	3,0	5,0	7,5	3,0	5,0	7,5	3,0	5,0	7,5	3,0	5,0	7,5	3,0	5,0	7,5

Avaliação sobre o critério Facilidade de Aprendizagem

	C3 - A1			C3 - A2			C3 - A3			C3 - A4			C3 - A5			C3 - A6			C3 - A7			C3 - A8				
	I	f	u	I	f	u	I	f	u	I	f	u	I	f	u	I	f	u	I	f	u	I	f	u	I	f
D 1	1,0	3,0	5,0	7,5	9,0	10	1,0	3,0	5,0	7,5	9,0	10	5,0	7,5	9,0	5,0	7,5	9,0	3,0	5,0	7,5	1,0	3,0	5,0		
D 2	1,0	3,0	5,0	3,0	5,0	7,5	0,0	1,0	3,0	7,5	9,0	10	3,0	5,0	7,5	3,0	5,0	7,5	3,0	5,0	7,5	3,0	5,0	7,5		
D 3	3,0	5,0	7,5	3,0	5,0	7,5	1,0	3,0	5,0	5,0	7,5	9,0	3,0	5,0	7,5	5,0	7,5	9,0	5,0	7,5	9,0	3,0	5,0	7,5		
D 4	1,0	3,0	5,0	7,5	9,0	10	1,0	3,0	5,0	5,0	7,5	9,0	5,0	7,5	9,0	5,0	7,5	9,0	3,0	5,0	7,5	3,0	5,0	7,5		
D 5	1,0	3,0	5,0	7,5	9,0	10	3,0	5,0	7,5	7,5	9,0	10	3,0	5,0	7,5	5,0	7,5	9,0	3,0	5,0	7,5	3,0	5,0	7,5		
D 6	5,0	7,5	9,0	7,5	9,0	10	3,0	5,0	7,5	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	3,0	5,0	7,5		
D 7	1,0	3,0	5,0	7,5	9,0	10	1,0	3,0	5,0	7,5	9,0	10	5,0	7,5	9,0	5,0	7,5	9,0	3,0	5,0	7,5	1,0	3,0	5,0		
D 8	1,0	3,0	5,0	5,0	7,5	9,0	0,0	1,0	3,0	7,5	9,0	10	5,0	7,5	9,0	5,0	7,5	9,0	1,0	3,0	5,0	1,0	3,0	5,0		
D 9	3,0	5,0	7,5	3,0	5,0	7,5	3,0	5,0	7,5	5,0	7,5	9,0	3,0	5,0	7,5	5,0	7,5	9,0	5,0	7,5	9,0	5,0	7,5	9,0		
D 10	1,0	3,0	5,0	7,5	9,0	10	1,0	3,0	5,0	5,0	7,5	9,0	3,0	5,0	7,5	7,5	9,0	10	5,0	7,5	9,0	3,0	5,0	7,5		
D 11	1,0	3,0	5,0	7,5	9,0	10	3,0	5,0	7,5	5,0	7,5	9,0	3,0	5,0	7,5	5,0	7,5	9,0	3,0	5,0	7,5	3,0	5,0	7,5		
D 12	5,0	7,5	9,0	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10		
D 13	3,0	5,0	7,5	7,5	9,0	10	5,0	7,5	9,0	7,5	9,0	10	5,0	7,5	9,0	7,5	9,0	10	7,5	9,0	10	5,0	7,5	9,0		
D 14	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10		
D 15	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10		
D 16	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10		
D 17	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10		
D 18	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10		
D 19	3,0	5,0	7,5	3,0	5,0	7,5	3,0	5,0	7,5	3,0	5,0	7,5	3,0	5,0	7,5	7,5	9,0	10	3,0	5,0	7,5	3,0	5,0	7,5		
D 20	5,0	7,5	9,0	5,0	7,5	9,0	5,0	7,5	9,0	5,0	7,5	9,0	5,0	7,5	9,0	7,5	9,0	10	5,0	7,5	9,0	5,0	7,5	9,0		
D 21	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10	7,5	9,0	10		

Avaliação sobre o Critério: Ferramentas e Recursos de Escrita de Código

	C4 - A1			C4 - A2			C4 - A3			C4 - A4			C4 - A5			C4 - A6			C4 - A7			C4 - A8		
	I	f	u	I	f	u	I	f	u	I	f	u	I	f	u	I	f	u	I	f	u	I	f	u
D 1	3,0	5,0	7,5	5,0	7,5	9,0	3,0	5,0	7,5	5,0	7,5	9,0	3,0	5,0	7,5	5,0	7,5	9,0	3,0	5,0	7,5	3,0	5,0	7,5
D 2	3,0	5,0	7,5	3,0	5,0	7,5	3,0	5,0	7,5	3,0	5,0	7,5	3,0	5,0	7,5	3,0	5,0	7,5	3,0	5,0	7,5	3,0	5,0	7,5
D 3	3,0	5,0	7,5	3,0	5,0	7,5	3,0	5,0	7,5	7,5	9,0	10	3,0	5,0	7,5	5,0	7,5	9,0	3,0	5,0	7,5	3,0	5,0	7,5
D 4	5,0	7,5	9,0	5,0	7,5	9,0	3,0	5,0	7,5	5,0	7,5	9,0	3,0	5,0	7,5	5,0	7,5	9,0	3,0	5,0	7,5	3,0	5,0	7,5

Anexo 1 – Fórmulas de normalização

Normalização Linear 1 - pelo maior valor	
Tipo	Fórmula
Benefício	$\hat{X}_{ij} = \frac{X_{ij}}{X_j^{Max}}$
Custo (Não Benefício)	$\hat{X}_{ij} = 1 - \frac{X_{ij}}{X_j^{Max}}$

Normalização Linear 2 - pelo maior valor para benefício e menor valor para o custo	
Tipo	Fórmula
Benefício	$\hat{X}_{ij} = \frac{X_{ij}}{X_j^{Max}}$
Custo (Não Benefício)	$\hat{X}_{ij} = \frac{X_j^{Min}}{X_{ij}}$

Normalização Linear 3 - pelo maior e menor valor (Max e Min)	
Tipo	Fórmula
Benefício	$\hat{X}_{ij} = \frac{X_{ij} - X_j^{Min}}{X_j^{Max} - X_j^{Min}}$
Custo (Não Benefício)	$\hat{X}_{ij} = \frac{X_j^{Max} - X_{ij}}{X_j^{Max} - X_j^{Min}}$

Normalização Linear 4 - pela soma	
Tipo	Fórmula
Benefício	$\hat{X}_{ij} = \frac{X_{ij}}{\sqrt{\sum_{i=1}^m X_{ij}^2}}$
Custo (Não Benefício)	$\hat{X}_{ij} = \frac{1}{\sum_{i=1}^m \left(\frac{1}{X_{ij}}\right)}$

Normalização Vetorial	
Tipo	Fórmula
Benefício	$\acute{X}_{ij} = \frac{X_{ij}}{\sum_{i=1}^m X_{ij}}$
Custo (Não Benefício)	$\acute{X}_{ij} = \frac{X_{ij}}{\sqrt{\sum_{i=1}^m X_{ij}^2}}$

Normalização de precisão aprimorada -	
Tipo	Fórmula
Benefício	$\acute{X}_{ij} = 1 - \frac{X_j^{Max} - X_{ij}}{\sum_{i=1}^m (X_j^{Max} - X_{ij})}$
Custo (Não Benefício)	$\acute{X}_{ij} = 1 - \frac{X_{ij} - X_j^{Min}}{\sum_{i=1}^m (X_{ij} - X_j^{Min})}$

Normalização Logarítmica	
Tipo	Fórmula
Benefício	$\acute{X}_{ij} = \frac{\log X_{ij}}{\log(\prod_{i=1}^m X_{ij})}$
Custo (Não Benefício)	$\acute{X}_{ij} = 1 - \frac{\log X_{ij}}{m - 1}$