

Alexandre Gouveia

Auxílio a tomada de decisão para ambientes na nuvem

Recife

20 de março de 2019

Alexandre Gouveia

Auxílio a tomada de decisão para ambientes na nuvem

Trabalho de conclusão de curso apresentado ao Curso de Bacharelado em Ciência da Computação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Carlos Julian Menezes Araújo

Universidade Federal Rural de Pernambuco

Departamento de Computação

Bacharelado em ciência da computação

Recife

20 de março de 2019

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema Integrado de Bibliotecas da UFRPE
Biblioteca Central, Recife-PE, Brasil

G719a Gouveia, Alexandre Vinicius da Silva
Auxílio a tomada de decisão para ambientes na nuvem /
Alexandre Vinicius da Silva Gouveia. – 2019.
43 f. : il.

Orientador: Carlos Julian Menezes Araujo.
Trabalho de Conclusão de Curso (Graduação) – Universidade
Federal Rural de Pernambuco, Departamento de Ciência da
Computação, Recife, BR-PE, 2019.
Inclui referências e apêndice(s).

1. Computação em nuvem 2. Processo decisório – Modelos
matemáticos 3. Processo decisório por critério múltiplo 4. Serviços
da Web I. Araujo, Carlos Julian Menezes, orient. II. Título

CDD 004



MINISTÉRIO DA EDUCAÇÃO E DO DESPORTO
UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO (UFRPE)
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

<http://www.bcc.ufrpe.br>

FICHA DE APROVAÇÃO DO TRABALHO DE CONCLUSÃO DE CURSO

Trabalho defendido por ALEXANDRE VINICIUS DA SILVA GOUVEIA às 10 horas do dia 03 de julho de 2019, na sala 07 (Auditório) CEAGRI II, como requisito para conclusão do curso de Bacharelado em Ciência da Computação da Universidade Federal Rural de Pernambuco, intitulado " **Auxílio a tomada de decisão para ambientes na nuvem**", orientado por Carlos Julian Menezes Araújo e aprovado pela seguinte banca examinadora:

Carlos Julian Menezes Araújo
DC/UFRPE

Danilo Ricardo Barbosa de Araujo
DC/UFRPE

Agradecimentos

Agradeço aos meus pais por sempre me apoiarem e orientar, ao Professor Carlos Julian Menezes por me orientar neste trabalho, aos meus amigos Jorge Candeias, Jefferson Carvalho, Raissa Camelo e Thays Silva pelo apoio e compartilhamento de experiências e aos professores do departamento por me ajudarem a chegar até aqui.

Resumo

A computação em nuvem surgiu com o intuito de oferecer serviços sob demanda, esta oferta de serviços permite uma redução no investimento de infraestrutura e manutenção das empresas. Porém com o passar do tempo o número de empresas que oferecem estes serviços tem crescido nos últimos anos. Transformando a tarefa de escolher o serviço em nuvem que mais se adequa as necessidades da empresa numa atividade que consome uma grande quantidade de tempo. Principalmente levando em consideração a quantidade de métricas que podem ser levadas em consideração como por exemplo: disponibilidade, preço, confiabilidade dentre outros. O objetivo deste trabalho é realizar uma implementação de uma solução baseada em métodos de decisão multicritério para mitigar o processo de escolha de serviços em nuvem, onde dado um conjunto de infraestruturas como serviço (IaaS) representada por métricas (por exemplo, disponibilidade e custo), o usuário pode fazer a escolha a partir de um *ranking* obtido aplicando os métodos de tomada de decisão multicritério PROMETHEE e TOPSIS. Com a finalidade de testar a ferramenta elaborada neste trabalho foram criados 4 casos utilizando-se de um estudo de caso disponível na literatura.

Palavras-chaves: Computação em nuvem, Infraestrutura como serviço, tomada de decisão, método multicritério, *PROMETHEE*, TOPSIS, MCDM, MCDA.

Lista de ilustrações

Figura 1 – Modelos de computação em nuvem.	15
Figura 2 – Fluxograma da ferramenta	22
Figura 3 – Tela de login	22
Figura 4 – Tela principal do sistema	23
Figura 5 – Tela de <i>upload</i> do arquivo	24
Figura 6 – Tela de resultados	24
Figura 7 – <i>Scatter plot</i> COA x Custo	29
Figura 8 – <i>Scatter plot</i> Confiabilidade x Custo	30
Figura 9 – <i>Scatter plot</i> Disponibilidade x Custo.	31
Figura 10 – <i>Scatter plot</i> Downtime x Custo	32

Lista de tabelas

Tabela 1 – Fatores e seus respectivos níveis	25
Tabela 2 – Dados da configurações de IaaS	26
Tabela 3 – Base de dados gerada	27
Tabela 4 – Ranqueamento dos resultados para COA e custo com peso 1.	28
Tabela 5 – Ranqueamento dos resultados para confiabilidade e custo com peso 1.	29
Tabela 6 – Ranqueamento dos resultados para disponibilidade e custo com peso 1.	30
Tabela 7 – Ranqueamento dos resultados para <i>Downtime</i> e custo com peso 1.	31

Lista de abreviaturas e siglas

AHP	<i>Analytic Hierachy Process</i>
CSMIC	<i>Cloud Service Measurent Index Consortium</i>
COA	<i>Capacity-Oriented Availability</i>
CSS	<i>Cascading Style Sheets</i>
ELECTRE	<i>Elimination Et Choice Translating Reality</i>
HTML	<i>HyperText Markup Language</i>
IaaS	<i>Infrasctruture as a Service</i>
MCDM	<i>Multiple-Criteria Decision Making</i>
MCDA	<i>Multiple-Criteria Decision Analysis</i>
MTTF	<i>Mean Time To Fail</i>
MTTR	<i>Mean Time to Repair</i>
PROMETHEE	<i>Preference Ranking Organization Method for Enrichment of Evaluations</i>
TOPSIS	<i>Technique for Order Preference by Similarity to Ideal Solution</i>

Lista de símbolos

Δ Letra maiúscula grega delta

Σ Letra maiúscula grega sigma

π Constante matemática pi

λ Letra minúscula lambda

Sumário

1	Introdução	10
1.1	Contexto	10
1.2	Problema e Justificativa	11
1.3	Trabalhos Relacionados	11
1.4	Objetivos	13
1.4.1	Objetivo Geral	13
1.4.2	Objetivos Específicos	13
1.5	Estrutura do Trabalho	14
2	Referencial teórico	15
2.1	Computação em nuvem	15
2.2	Métodos MCDM	16
2.2.1	PROMETHEE II	16
2.2.2	TOPSIS	17
2.3	Métricas	19
2.3.1	<i>Mean Time To fail (MTTF) e Mean Time To repair MTTR</i>	19
2.3.2	COA	19
2.3.3	Confiabilidade	19
2.3.4	Disponibilidade	19
2.3.5	<i>Downtime</i>	20
2.4	Distância Euclidiana	20
3	Materiais e métodos	21
3.1	<i>O framework</i>	21
4	Resultados e Discussão	25
4.1	Introdução	25
4.2	Caso 1	28
4.3	Caso 2	29
4.4	Caso 3	30
4.5	Caso 4	31
4.6	Discussão	32
5	Conclusão	34
	Conclusão	34

Referências	35
Apêndices	37
APÊNDICE A Implementação do TOPSIS	38
APÊNDICE B Implementação do PROMETHEE II	40

1 Introdução

1.1 Contexto

A computação em nuvem emergiu como um paradigma de computação para fornecer recursos sob demanda (por exemplo, infraestrutura, plataforma e *software*) para usuários semelhantes a concessionárias de serviços públicos (por exemplo, água, eletricidade e gás) (SANKET, 2018). Esse paradigma tem se tornado uma solução interessante para muitas empresas que buscam escalabilidade e flexibilidade para o provimento de seus serviços. Essa mudança reflete também nas empresas que buscam reduzir custos com a aquisição e manutenção dos recursos de computação, assim como, almejam aumentar a eficiência no provimento do serviço.

Com o passar do tempo a utilização da computação em nuvem vem crescendo dentro das empresas, e em 2016 a adoção de nuvem híbridas cresceu de 19% para 57% (COLUMBUS, 2017). A computação em nuvem tem por objetivo entregar recursos sob demanda para seus usuários. Dentre os modelos de nuvem temos: a nuvem pública em que empresas provedoras oferecem serviços através da Internet pública, tornando-o disponível para quem quiser utilizar ou comprar; na nuvem privada os serviços são disponibilizados através da Internet ou de uma rede interna, permitindo um controle adicional ou customizações de recursos dedicados sobre uma estrutura computacional hospedada na empresa. A nuvem privada também permite um nível maior de segurança. Há um terceiro modelo que é conhecido como nuvem híbrida, que combina os modelos supracitados permitindo que dados e aplicações sejam compartilhados entre si, quando a demanda flutua, a computação em nuvem híbrida permite de forma transparente escalar a infraestrutura local da empresa (MICROSOFT, 2017).

Esta questão de qual serviço em nuvem escolher pode ser incluída dentro dos problemas de tomada de decisão multicritério, onde diversos fatores devem ser considerados para que se escolha uma solução que satisfaça os critérios estabelecidos. Dentre os métodos para tomada de decisão multicritério alguns dos mais utilizados são o *Preference Ranking Organization Method for Enrichment Evaluation (PROMETHEE)* e o *Technique for Order Performance by Similarity to Ideal Solution (TOPSIS)* (KITTUR, 2015).

No mercado de serviços em nuvem podemos citar um grande número de empresas oferecendo inúmeras opções de serviço dentre as quais temos: Amazon (AWS), Embratel, Microsoft (Azure), Google (Google cloud), dentre outros. Assim, com uma grande diversidade de serviços ofertados na nuvem, a tarefa de escolher uma opção adequada se torna bastante difícil para o usuário. Adicionalmente, podemos também citar diversos critérios

associados a cada serviço, como custo, desempenho, confiabilidade, disponibilidade dentre outros, o que torna esta uma atividade ainda mais trabalhosa.

1.2 Problema e Justificativa

O problema fundamental tratado neste trabalho de graduação é abordado na seguinte pergunta: Como auxiliar o processo de seleção de serviços computacionais em nuvens considerando políticas de prestação de serviços com restrição de orçamento, garantindo níveis aceitáveis de disponibilidade e desempenho?

O crescente número de informações sobre serviços ofertados na nuvem e a necessidade de utilizá-los no processo decisório fez com que surgissem diversas pesquisas com o intuito de auxiliar gestores e usuários na tomada de decisão (GARG; VERSTEEG; BUYYA, 2013; WHAIDUZZAMAN ABDULLAH GANI; HAQUE, 2014; ALABOOL et al., 2018). Os métodos multicritérios de tomada de decisão surgiram como métodos de apoio para resolução de problemas em que existam critérios conflitantes (por exemplo, custo e confiabilidade). Assim, a tomada de decisão multicritério é uma abordagem que pode auxiliar na seleção de diferentes alternativas que consideram um conjunto de critérios e também confere ao processo clareza e transparência (SABAEI; ERKOYUNCU; ROY, 2015; KROENKE ANDRÉIA DANI, 2016).

A utilização de métodos multicritérios ocorre pelo fato de que não há, em geral, processos decisórios que atendam simultaneamente a todos os critérios em análise. Assim, considerando diversos critérios que precisam ser levados em conta no processo de seleção de serviços em nuvem, métodos de decisão multicritério se tornam uma escolha interessante na seleção desses serviços. Tais métodos também auxiliam na redução dos riscos que são associados ao processo de escolha como custo-necessidade.

1.3 Trabalhos Relacionados

O trabalho de Garg (GARG; VERSTEEG; BUYYA, 2013) teve por objetivo a criação de um *framework* utilizando o *Analytic Hierarchy Process* (AHP) para a comparação de serviços na nuvem. Esta comparação foi feita por meio de atributos propostos pelo *Cloud Service Measurent Index Consortium* (CSMIC) como custo, agilidade, segurança, desempenho, garantia(assurance) e accountability. Os autores acreditam que este *framework* representa um passo importante no que diz respeito a seleção de serviços na nuvem, e que ao utiliza-lo provedores de serviço na nuvem podem comparar sua oferta de serviço com os concorrentes.

O trabalho proposto por SABAEI (SABAEI; ERKOYUNCU; ROY, 2015) tem o objetivo de realizar uma revisão dos métodos de decisão multicritério, avaliá-los e reali-

zar uma análise crítica. Os autores constataram a existência de uma lacuna no que diz respeito a uma comparação entre os métodos de tomada de decisão, visto que é possível acompanhar uma descrição do diferentes trabalho conseguiram preencher esta lacuna ao apresentar uma análise dos diferentes métodos de tomada de decisão. Segundo os resultados obtidos no estudo, o método *Analytic Hierarchy Proces* (AHP) pode providenciar uma solução que melhor atenda o usuário, pôr levar em consideração as preferências.

No trabalho feito por Kroenke ([KROENKE ANDRÉIA DANI, 2016](#)), foi utilizado o método para tomada de decisão multicritério AHP para a seleção de fornecedores. Este trabalho foi desenvolvido com base em uma pesquisa de campo, por meio de um estudo de caso. A tomada de decisão foi dividida em três etapas: i) parametrização do modelo que inicia com a escolha dos funcionários que estarão ligados ao processo de tomada de decisão, ii) simulação computacional baseado no modelo AHP, em que a partir dos dados coletados será gerada uma planilha com as alternativas de serviços a partir da qual será gerado um *ranking* das alternativas, iii) os funcionários selecionados na etapa 1 irão selecionar as alternativas com base nos resultados da etapa 2. Os autores ressaltam que apesar de se utilizar um método de decisão, a palavra final sobre qual fornecedor escolher cabe ao gestor.

O estudo feito por Whaiduzzaman ([WHaiduZZAMAN ABDULLAH GANI; HAQUE, 2014](#)) teve como objetivos: realizar uma análise comparativa entre os métodos de análise de decisão multicritério (MCDM), apresentar uma taxonomia destes métodos através de uma revisão literária, analisar e resumir a seleção de serviços de computação em nuvem em diferentes cenários. Os autores concluíram que os MCDM tem um grande efeito e importância nos cenários de tomada de decisão multicritério. Este estudo realizado por Whaiduzzaman apresenta as vantagens e desvantagens bem como aplicações destes métodos analisados.

O trabalho realizado por Bougrine ([BOUGRINE, 2017](#)) aplicou o método TOPSIS para otimizar um sistema de manufatura reconfigurável. Os autores apontam que fatores ergonômicos e humanos raramente são levados em consideração para avaliação das oportunidades de reconfiguração. Para esta avaliação foram utilizados os seguintes indicadores para ergonômicos: peso manuseado pelos operadores, postura, condições de trabalho, tempo e avaliação ergonômica da atividade de reconfiguração. Quanto aos indicadores operacionais foram utilizados: a vazão de produtos, atraso do produto, tempo para manufatura do produto, utilização do maquinário, número de máquinas, número de operadores e tempo de reconfiguração. E para o estudo de caso foram avaliadas quatro configurações diferentes. O artigo conclui que os indicadores ergonômicos têm grande influência e é altamente dependente do critério peso.

O trabalho realizado por Hajkowicz ([HAJKOWICZ; HIGGINS, 2008](#)) propõe realizar uma comparação entre cinco métodos de decisão multicritério sendo esses: *PRO-*

METHEE II, *Weighted summation*, *range of value method*, *Evamix* e *compromised programming*. Para essa comparação foi utilizado um estudo de caso referente ao gerenciamento de recursos hídricos. É dito nesse trabalho que a natureza multicritério do gerenciamento de recursos hídricos torna esta área de aplicação adequada para estudos comparativos de métodos de decisão multicritério, e que ele possui relevância em outros campos de gerenciamento de recursos naturais e de tomada de decisão. Para comparar os resultados obtidos com a aplicação dos métodos foram utilizados o *Spearman's rank correlation coefficient* e o *Kendal's coefficient of concordance*. O teste foi feito de acordo com as três melhores opções, tendo em vista que alguns dos métodos selecionam apenas uma única opção. O estudo proposto encontrou uma grande concordância entre os diferentes métodos, houveram poucos casos em que as técnicas geraram resultados diferentes.

O trabalho realizado por (GRGUREVIĆ, 2017) teve por objetivo realizar uma revisão da literatura para definir quais seriam os melhores MCDA para avaliar estruturas de computação em nuvem, sendo elas dividida em: IaaS (*Infrastructure as a Service*), PaaS (*Platform as a Service*) e SaaS (*Software as a Service*). Para os autores os seguintes atributos são utilizados de forma mais recorrente: segurança, desempenho, acessibilidade, usabilidade, escalabilidade e adaptabilidade. Para todo negócio de seleção de serviço na nuvem o custo é um dos critérios mais importantes seguido por privacidade e segurança. Este trabalho conclui que nem todo MCDA é adequado para seleção de serviços na nuvem, dependendo da natureza do problema alguns métodos são mais eficientes do que outros. Por exemplo: para SaaS o AHP é o métodos mais recomendado, para PaaS o TOPSIS seria mais indicado e para IaaS todos os métodos avaliados são igualmente apropriados.

1.4 Objetivos

1.4.1 Objetivo Geral

Realizar uma implementação de uma solução baseada em métodos de decisão multicritério para mitigar o processo de escolha de serviços em nuvem. Mais especificamente dado um conjunto de infraestruturas como serviço (IaaS) representada por métricas (por exemplo, disponibilidade e custo), o usuário pode fazer a escolha a partir de um *ranking*.

1.4.2 Objetivos Específicos

1. Implementar uma abordagem baseada em algoritmos multi-critérios que auxilie os usuários de serviços na nuvem na seleção de serviços.
2. Utilizar um estudo de caso existente para demonstrar a aplicabilidade da ferramenta no processo de tomada de decisão.

1.5 Estrutura do Trabalho

O Capítulo 2 descreve os conceitos fundamentais para o entendimento deste trabalho. O Capítulo 3 é responsável por explicar os métodos utilizados bem como um passo-a-passo dos mesmos. O Capítulo 4 corresponde a parte dos resultados e interpretação dos mesmos. O Capítulo 5 contém as conclusões obtidas a partir dos resultados obtidos, limitações e trabalhos futuros.

2 Referencial teórico

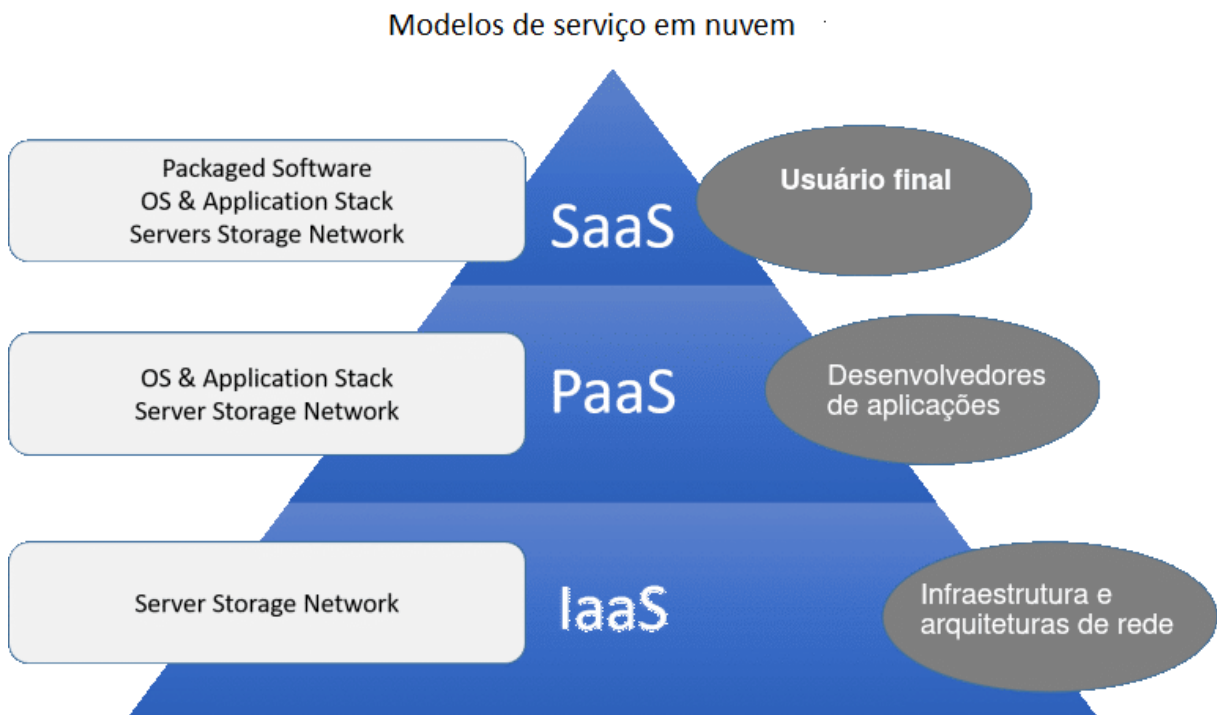
Este capítulo aborda os conceitos básicos sobre computação em nuvem e métodos de tomada de decisão multicritério. Neste sentido serão apresentados exemplos de critérios que podem ser utilizados no processo de tomada de decisão considerando sistemas computacionais na nuvem.

2.1 Computação em nuvem

A computação em nuvem é o fornecimento de serviços sob demanda por meio da internet. Ela permite uma maior flexibilidade na oferta de serviços, o que permite um controle no gastos por parte do contratante.

Os serviços providos pela nuvem podem ser divididos em três grupos principais: *software* como serviço (SaaS) que provê *software* e aplicações através da Internet, Plataforma como serviço (PaaS) que entrega aos usuários um ambiente para desenvolver, gerenciar e entrega aplicações e Infraestrutura as a Serviço (IaaS) que provê acesso a recursos computacionais como servidores, *storages* e redes (BARABAS, 2016).

Figura 1 – Modelos de computação em nuvem.



Fonte: <https://www.uniprint.net/en/7-types-cloud-computing-structures/> (Editado)

2.2 Métodos MCDM

Os métodos MCDM foram criados com o intuito de auxiliar e indicar quais as melhores opções dentre um conjunto de alternativas para um determinado problema. E devido as suas características cada método pode trabalhar com atributos conflitantes.

2.2.1 PROMETHEE II

O método *PROMETHEE* criado por Brans em 1982 permite ordenar alternativas baseado no grau de preferência. O *PROMETHEE* I é um método de decisão multicritério de ordenação parcial das alternativas, sendo baseado em fluxos positivo e negativo e utiliza três tipos de resultado: preferência, indiferença e comparabilidade. O *PROMETHEE* II é um método de ordenação completa das alternativas cujo os resultados possíveis são: preferência e indiferença. Segundo (ABEDI S. ALI TORABI, 2011) o *PROMETHEE* II é um dos métodos *PROMETHEE* mais utilizados. O princípio básico do *PROMETHEE* II é baseado em uma comparação por pares de alternativas e o uso de duas informações adicionais: o peso de cada atributo e uma função de preferência fornecida pelo tomador de decisões. Os passos descritos por (ABEDI S. ALI TORABI, 2011) para se implementar o *PROMETHEE* II são:

Passo 1: Criar uma matriz de avaliação, em que o valor de cada linha representa o desempenho de cada alternativa em um determinado critério;

$$\begin{bmatrix} g_a(a) & g_b(a) & g_c(a) & \dots & g_m(a) \\ g_a(b) & g_b(b) & g_c(b) & \dots & g_m(b) \\ g_a(c) & g_b(c) & g_c(c) & \dots & g_m(c) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_a(n) & g_b(n) & g_c(n) & \dots & g_m(n) \end{bmatrix}$$

Passo 2: Calcular o grau de preferência entre as alternativas a e b para cada atributo j em um conjunto A ;

$$\text{Para cada } j \text{ em } A \quad d_j(a, b) = g_j(a) - g_j(b) \quad (2.1)$$

Passo 3: Aplicar as funções de preferência fornecidas pelo tomador de decisão, como descrito na fórmula 2.2;

$$P_j(a, b) = F_j[d_j] \quad j = 1, \dots, k \quad \& \quad \forall a, b \in A \quad (2.2)$$

Passo 4: Calcular o índice de preferência agregado para cada par de alternativas;

$$\pi(a, b) = \frac{\sum_{j=1}^k P_j(a, b)w_j}{\sum_{j=1}^k w_j} \quad \forall a, b \in A \quad (2.3)$$

Passo 5: Calcular o fluxo de superação de cada alternativa quando comparada com as outras alternativas. Este calculo é dividido em duas partes: o fluxo positivo e o fluxo negativo;

$$\phi^+(a) = \frac{1}{1-n} \sum_{x \in A} \pi(a, x) \text{ Para cada } a \quad (2.4)$$

$$\phi^-(a) = \frac{1}{1-n} \sum_{x \in A} \pi(x, a) \text{ Para cada } a \quad (2.5)$$

Passo 6: Cálculo do fluxo de superação final de cada alternativas;

$$\phi(a) = \phi^+(a) - \phi^-(a) \quad (2.6)$$

2.2.2 TOPSIS

O TOPSIS foi desenvolvido em 1981 por *Tzeng e Hwang*. O método é baseado no conceito de escolher a alternativa que possui a menor distância Euclidiana para a solução ideal e a maior distância da solução ideal negativa. A solução ideal é uma solução hipotética pela qual todos valores de atributos correspondem ao valor máximo do atributo na base de dados composta: A solução ideal negativa é uma solução hipotética onde todos os valores dos atributos correspondem ao valor mínimo do atributo (RAO, 2007), ou seja, o TOPSIS não apenas fornece uma solução próxima do valor ideal como também um valor distante dos piores valores hipotéticos.

O método TOPSIS pode ser dividido nos 8 passos descritos abaixo:

Passo 1: Criar uma matriz baseada nas informações disponíveis no formato a_{mj} onde cada linha (m) representa uma alternativa e cada coluna (j) representa o valor de um atributo;

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1j} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2j} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3j} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mj} \end{bmatrix}$$

Passo 2: Normalizar os valores da matriz gerada no passo 1;

$$R_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}} \quad (2.7)$$

Passo 3: Definir o peso que cada um dos atributos terá;

Passo 4: Construir uma matriz resultante da multiplicação dos valores normalizados e seus devidos pesos;

$$\begin{bmatrix} a_{11} \times w_1 & a_{12} \times w_2 & a_{13} \times w_3 & \dots & a_{1j} \times w_j \\ a_{21} \times w_1 & a_{22} \times w_2 & a_{23} \times w_3 & \dots & a_{2j} \times w_j \\ a_{31} \times w_1 & a_{32} \times w_2 & a_{33} \times w_3 & \dots & a_{3j} \times w_j \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} \times w_1 & a_{m2} \times w_2 & a_{m3} \times w_3 & \dots & a_{mj} \times w_j \end{bmatrix}$$

Passo 5: Obter a solução ideal positiva e a solução ideal negativa, selecionando o melhor e o pior valor para cada um dos atributos ;

$$V^+ = ((\max a_{ij}) \quad \forall \quad j \quad \text{in} \quad J', i = 1, 2, 3, \dots, m) \quad (2.8)$$

$$V^- = ((\min a_{ij}) \quad \forall \quad j \quad \text{in} \quad J', i = 1, 2, 3, \dots, m) \quad (2.9)$$

Passo 6: Obter a medida de separação positiva e negativa;

$$S_i^+ = \sqrt{\sum_{j=1}^n (V_{ij} - V_j^+)^2} \quad (2.10)$$

$$S_i^- = \sqrt{\sum_{j=1}^n (V_{ij} - V_j^-)^2} \quad (2.11)$$

Passo 7: Obter a proximidade relativa de cada alternativa a solução ideal negativa;

$$C_i = \frac{S_i^-}{S_i^+ + S_i^-} \quad (2.12)$$

Passo 8: Ordenar as alternativas de acordo com os resultados obtidos.

2.3 Métricas

Esta seção apresenta um conjunto de métricas utilizados para avaliar serviços na nuvem.

2.3.1 Mean Time To fail (MTTF) e Mean Time To repair MTTR

MTTF ou tempo médio para falhar é o tempo médio que um determinado dispositivo ou serviço tem até falhar e MTTR ou tempo médio para reparo é o tempo médio que se leva para que determinado dispositivo ou serviço volte a funcionar.

2.3.2 COA

Capacity-Oriented Availability (COA) é um percentual que representa quanto de um serviço provido por um sistema é entregue, levando em consideração não apenas a disponibilidade do mesmo como também o impacto destas condições na entrega do serviço (ARAÚJO, 2019). O COA pode ser obtido por meio da seguinte equação;

$$COA = \frac{\sum_{S_i \in S} pc_i \times \pi_i}{N}, \quad (2.13)$$

onde pc representa a quantidade de recursos disponíveis em qualquer estado S , S é o conjunto de estados que podem ser alcançados e N é a capacidade máxima do sistema.

2.3.3 Confiabilidade

A confiabilidade é definida como a probabilidade de que um dispositivo irá desempenhar as suas funções pretendidas satisfatoriamente durante um período especificado de tempo, sob condições de operação específicas (DANTAS, 2018). Esta métrica foi obtida através de uma análise sobre um período de tempo de 24 horas. A confiabilidade de um sistema pode ser obtida utilizando a seguinte equação:

$$C(t) = e^{-\lambda \times t}, \quad (2.14)$$

em que t é o tempo para projeção e λ é a taxa de falhas do sistema.

2.3.4 Disponibilidade

Disponibilidade, é a probabilidade do sistema estar funcionando com o passar do tempo, mesmo que não esteja em sua carga máxima, pode-se obter o valor da disponibilidade utilizando a equação abaixo,

$$Disponibilidade = \frac{MTTF}{MTTF + MTTR}. \quad (2.15)$$

2.3.5 Downtime

Downtime é o período dado em minutos em que o sistema está inoperante em um período total considerado. Neste estudo será considerado o *downtime* durante o período de um mês (720 horas).

2.4 Distância Euclidiana

É utilizada para calcular a similaridade entre os resultados obtidos por cada configuração e o melhor valor obtido dentre o conjunto de soluções possíveis. Este cálculo foi aplicado para gerar um valor único para se comparar os resultados obtidos em cada configuração será calculada a distância euclidiana como descrito na equação abaixo:

$$score = \sqrt{(max(TS) - TS(i))^2 + (max(PRO) - PRO(i))^2}, \quad (2.16)$$

$TS(i)$ e $PRO(i)$ são respectivamente os valores obtido pela configuração i respectivamente no TOPSIS e PROMETHEE, $max(TS)$ e $max(PRO)$ são os maiores valores obtido por uma configuração dentre o grupo analisado.

3 Materiais e métodos

Este capítulo descreve as características metodológicas utilizadas neste trabalho. Também descreve a definição da pesquisa quanto aos objetivos, quanto à natureza da abordagem e quanto aos procedimentos técnicos, assim como as etapas da implementação da ferramenta proposta.

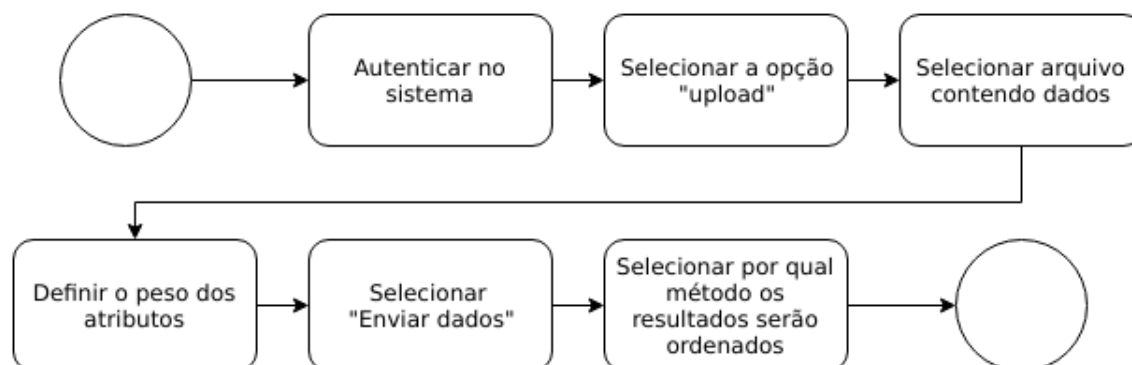
3.1 O *framework*

Para este trabalho foi confeccionada uma ferramenta para *web* que implementa os métodos MCDM PROMETHEE II e TOPSIS. Para este fim foi utilizada a linguagem de programação *Python* e o *microframework Flask* para o desenvolvimento do *backend* e *html*, *css* e *javascript* com o auxílio do *template engine jinja2* para o desenvolvimento do *frontend*. O desenvolvimento da aplicação foi dividida em 7 fases:

- Levantar os requisitos da ferramenta;
- Desenvolvimento do *backend* junto ao do *frontend* tendo em vista que o *flask* utiliza o *jinja2* para gerenciar os *templates* do projeto;
- Teste do *backend e frontend*(conta de usuário e renderização *templates*);
- Desenvolvimento dos métodos MCDM na linguagem *python*;
- Teste dos métodos desenvolvidos;
- Acoplamento dos métodos ao *backend*;
- Teste da ferramenta com os métodos inclusos;

Desenvolver utilizando *flask* permitiu que a ferramenta fosse criada sem muitas dificuldade por alguém sem conhecimento prévio sobre desenvolvimento *web* e graças a linguagem *python* a manipulação dos dados se torna mais fácil em relação a linguagens como Java. A seguir temos o fluxograma e um passo-a-passo explicando como se utilizar a ferramenta:

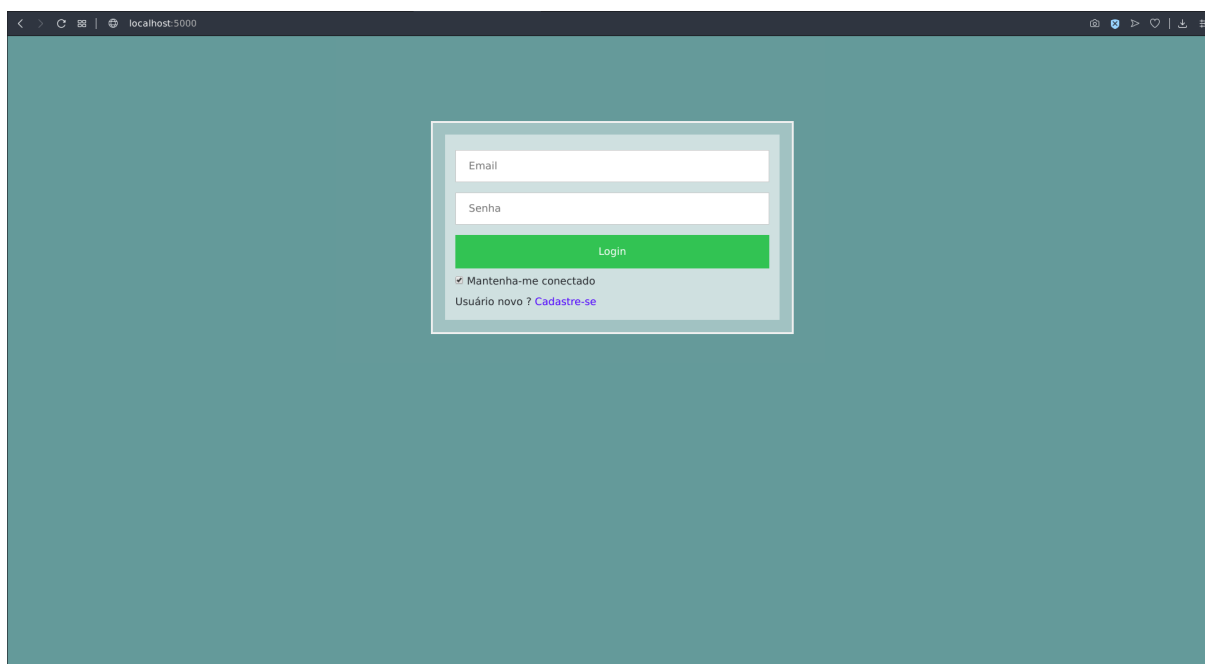
Figura 2 – Fluxograma da ferramenta



Fonte: Elaborado pelo autor

- Fazer login no sistema utilizando *e-mail* e senha e pressionar o botão "logar";

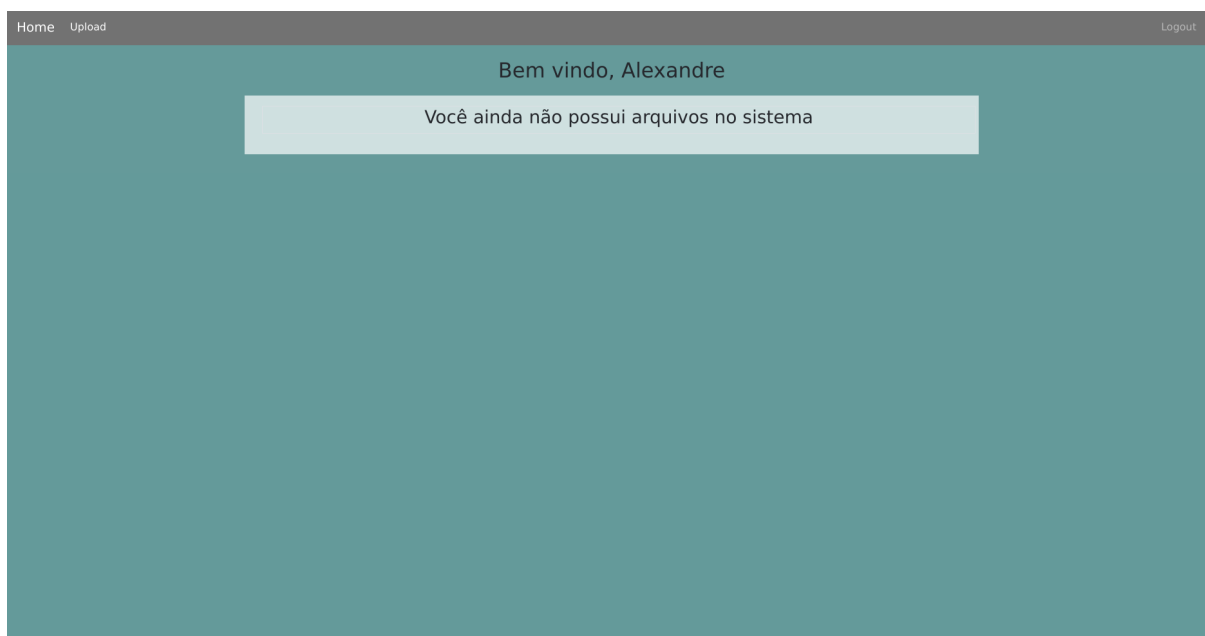
Figura 3 – Tela de login



Fonte: Elaborado pelo autor

- Pressionar o botão "upload" no canto superior esquerdo da tela;

Figura 4 – Tela principal do sistema



Fonte: Elaborado pelo autor

- Nesta tela você deve selecionar o arquivo que deseja enviar, os formatos suportados são xls, xlsx e csv. O arquivo deve estar no seguinte formato: cada linha representa uma configuração de IaaS e as colunas devem conter respectivamente as métricas COA, confiabilidade, disponibilidade, *Downtime* e custo. Caso o arquivo possua cabeçalho ou as configurações possuam nomes deve-se marcar as respectivas *checkbox*;
- Definir os pesos de cada um dos 5 atributos, que devem variar de 0,1 até 1 com passos de 0,1;
- Pressionar o botão "Enviar arquivo"

Figura 5 – Tela de *upload* do arquivo

Selecione um arquivo(xls, xlsx ou csv)

Escolher arquivo Nenhum arquivo selecionado

Contém cabeçalho ?

As alternativas possuem nome ?

Peso dos atributos

Disp.	COA	Conf.	Downtime	Custo
0,1	0,1	0,1	0,1	0,1

Enviar dados

Fonte: Elaborado pelo autor

- Na tela de resultados pode ordená-los usando PROMETHEE, TOPSIS ou pela distância euclidiana(usada como *score* final);

Figura 6 – Tela de resultados

#	PROMETHEE	TOPSIS	Média
Alternativa 9	-0.0281593836	0.939695658	0.4557681372
Alternativa 0	-0.0429190496	0.939695658	0.4483835042
Alternativa 10	-0.0281593836	0.8754551327	0.4236476745
Alternativa 11	-0.0281593836	0.8754551327	0.4236476745
Alternativa 1	-0.0429190496	0.8754551327	0.4162696415
Alternativa 2	-0.0429190496	0.8754551327	0.4162696415
Alternativa 26	0.0143594336	0.7573756939	0.3856725638
Alternativa 21	0.0143594336	0.7010348159	0.3577021247
Alternativa 22	0.0143594336	0.6342375518	0.3245034927
Alternativa 23	0.0143594336	0.6342375518	0.3245034927
Alternativa 24	0.0143594336	0.6342375518	0.3245034927
Alternativa 25	0.0143594336	0.6212815925	0.3178255131
Alternativa 15	0.0297701531	0.3757780896	0.2027731214
Alternativa 16	0.0143594336	0.3869523057	0.2006698696
Alternativa 13	0.0143594336	0.3757780896	0.1956737617
Alternativa 14	0.0143594336	0.3757780896	0.1956737617
Alternativa 7	-0.0281593836	0.3869523057	0.179395461
Alternativa 4	-0.0281593836	0.3757780896	0.1736093531
Alternativa 5	-0.0281593836	0.3757780896	0.1736093531
Alternativa 6	-0.0281593836	0.3757780896	0.1736093531
Alternativa 12	0.0143594336	0.3064913856	0.1604394096
Alternativa 3	-0.0281593836	0.3064913856	0.139165001
Alternativa 17	0.0143594336	0.2505846699	0.1324757513
Alternativa 8	-0.0281593836	0.2505846699	0.1112123427

Fonte: Elaborado pelo autor

O *framework* pode ser acessado através do *github* através da url: https://github.com/Alexandregouveia/Tcc_project.

4 Resultados e Discussão

4.1 Introdução

Este capítulo tem como objetivo demonstrar a aplicabilidade dos métodos de tomada de decisão multi-critério por meio da ferramenta proposta. Com base no estudo de caso criado por (ARAÚJO, 2019), que consiste em 36 configurações de IaaS, a partir deste estudo de caso foram criados 4 casos para esta avaliação:

- **caso 1** Nesta situação o ranqueamento considerou as métricas COA e custo como as mais prioritárias (recebendo peso 1) e as demais métricas receberam peso 0.1;
- **caso 2** Nesta situação o ranqueamento considerou as métricas confiabilidade e custo como as mais prioritárias (recebendo peso 1) e as demais métricas receberam peso 0.1;
- **caso 3** Nesta situação o ranqueamento considerou as métricas disponibilidade e custo como as mais prioritárias (recebendo peso 1) e as demais métricas receberam peso 0.1;
- **caso 4** Nesta situação o ranqueamento considerou as métricas *downtime* e custo como as mais prioritárias (recebendo peso 1) e as demais métricas receberam peso 0.1;

Estes casos descritos acima servem para analisar como os resultados obtidos se comportam. A seguir temos a Tabela 1 contendo os fatores e seus respectivos níveis utilizados para a construção das configurações utilizadas no estudo de caso:

Tabela 1 – Fatores e seus respectivos níveis

Fatores	Níveis
Nós	1 e 2
VMs por nó	1, 2 e 3
Tipo de serviço	<i>Gold, Silver e Bronze</i>
Tipo de redundância	<i>N/R, Hot, Cold e Warm</i>

Fonte: (ARAÚJO, 2019)

- Quantidade de nó
- Numero de máquinas virtuais por nó

- Tipo de serviço - Representa os níveis de manutenção, esse fator reflete a capacidade do time de manutenção suportar diferentes níveis de qualidade de manutenção. É considerada uma redução de 10% no MTTR usando um serviço *Silver* comparado com um serviço *Gold*, e uma diferença de 20% no MTTR usando um serviço *Bronze* em relação ao um serviço *Gold*.
- Tipo de redundância - Refere-se a provisão de recursos de redundância. É dividido em três níveis: *Hot redundancy* quando o sistema não deve ser desligado, mesmo por um breve período. O sistema utiliza uma unidade de redundância capaz de suportar a mesma carga de trabalho que a unidade principal. *Cold redundancy* troca para a unidade de redundância após a falha da unidade principal. Para que a troca ocorra é agendado um horário para que se substitua a unidade principal pela unidade de redundância. *Warm redundancy* o tempo para substituição da unidade principal pela unidade de redundância é reduzido. N/R significa que a unidade não possui qualquer tipo de redundância.

Fazendo uso das informações contidas na Tabela 2 foram criados modelos com redes de Petri estocásticas e diagrama de blocos de confiabilidade, em seguida o autor apresenta os dados analisados a partir dos modelos. Foi gerada a Tabela 3, que foi utilizada como entrada para o *framework*. De posse dos resultados, adotamos a nossa proposta para realizar o ranqueamento consideramos as métricas: COA, confiabilidade, disponibilidade, *Downtime* e custo.

Tabela 2 – Dados da configurações de IaaS

Conf.	Nós	#VM	TS	TR	Conf.	Ns	#VM	TS	TR
1	1	1	Gold	N/R	34	2	2	Bronze	Hot
5	1	1	Silver	N/R	35	2	2	Bronze	Cold
9	1	1	Bronze	N/R	36	2	2	Bronze	Warm
14	1	2	Gold	Hot	38	2	4	Gold	Hot
15	1	2	Gold	Cold	39	2	4	Gold	Cold
16	1	2	Gold	Warm	40	2	4	Gold	Warm
18	1	2	Silver	Hot	42	2	4	Silver	Hot
19	1	2	Silver	Cold	43	2	4	Silver	Cold
20	1	2	Silver	Warm	44	2	4	Silver	Warm
22	1	2	Bronze	Hot	46	2	4	Bronze	Hot
23	1	2	Bronze	Cold	47	2	4	Bronze	Cold
24	1	2	Bronze	Warm	48	2	4	Bronze	Warm
26	2	2	Gold	Hot	50	3	3	Gold	Hot
27	2	2	Gold	Cold	54	3	3	Silver	Hot

28	2	2	Gold	Warm	58	3	3	Bronze	Hot
30	2	2	Silver	Hot	62	3	6	Gold	Hot
31	2	2	Silver	Cold	66	3	6	Silver	Hot
32	2	2	Silver	Warm	70	3	6	Bronze	Hot

Tabela 3 – Base de dados gerada

#	COA(%)	Conf.(%)	Disp. (%)	Downtime(Min)	Custo(\$)
1	99.76543387	89.4469247967	99.5509574705	196.68062793	306.315
5	99.73985565	89.4469247967	99.5048180576	216.88969077	301.106
9	99.71429055	89.4469247967	99.4587108348	237.08465436	297.616
14	99.87285352	90.2149434700	99.5706543903	188.05337705	306.292
15	99.77596818	90.4897545045	99.5616066096	192.01630500	306.302
16	99.77936634	90.4925359715	99.5650001940	190.52991502	306.299
18	99.85761355	90.2149434700	99.5280378157	206.71943671	301.090
19	99.75435300	90.4897545045	99.5194114138	210.49780078	301.096
20	99.75832953	90.4925359715	99.5233822554	208.75857214	301.093
22	99.84238241	90.2149434700	99.4854486046	225.37351118	297.605
23	99.73294430	90.4897545045	99.4774397512	228.88138896	297.608
24	99.73739454	90.4925359715	99.4818832542	226.93513466	297.607
26	99.97929139	94.7970536818	99.7846109835	94.34038921	348.837
27	99.92582999	94.8881075626	99.7201853569	122.55881369	338.826
28	99.94998262	94.7885068480	99.7486208259	110.10407827	331.844
30	99.97417865	97.970536818	99.7638083387	103.45194767	348.939
31	99.91964902	94.8881075626	99.6976733870	132.41905652	349.264
32	99.94459634	94.7885068480	99.7272945753	119.44497603	349.119
34	99.96904946	94.7970536818	99.7430013430	112.56541175	338.735
35	99.91362024	94.8881075626	99.6754125418	142.16930671	339.006
36	99.93925348	94.7885068480	99.7060521418	128.74916191	338.883
38	99.97639721	94.8827293630	99.7846995558	94.30159457	331.721
39	99.93639415	94.9415323833	99.7282517140	119.02574928	331.913
40	99.96410148	94.8717737486	99.7534014438	108.01016760	331.828
42	99.97128226	94.8827293630	99.7639238067	103.40137265	612.144
43	99.93419387	94.9415323833	99.7061245286	128.71745648	612.275
44	99.96330957	94.8717737486	99.7322176473	117.28867048	612.216
46	99.96615039	94.8827293630	99.7431470289	112.50160136	601.884
47	99.93234323	94.9415323833	99.6841904563	138.32458013	601.965
48	99.96265691	92.7019871643	99.7110986022	126.53881225	601.928

50	99.98020828	95.1170633317	99.7851593857	94.10018905	594.981
54	99.97527976	95.1170633317	99.7644824363	103.15669289	594.997
58	99.97035136	95.1170633317	99.7438139525	112.20948879	595.012
62	99.98020440	95.1242439537	99.7851596846	94.10005812	612.095
66	99.97527366	95.1242439537	99.7644828673	103.15650413	601.855
70	99.97034257	95.1242439537	99.7438147914	112.20912137	595.012

Fonte: (ARAUJO, 2019)

Utilizando as informações contidas na Tabela 3 foram criados 4 casos para se verificar a aplicabilidade do *framework*. Para cada um dos cenários foi atribuído peso 1 para o custo junto com um dos outros atributos, e os demais atributos receberam peso 0,1 para reduzir a influência dos mesmos sobre o resultado, assim espera-se que a ferramenta selecione as configurações em que os valores dos atributos com peso 1 sejam maiores. As subseções a seguir contém um *ranking* com os 10 melhores resultados e um gráfico de dispersão onde as configurações foram divididas em 2 grupos: um chamado de *Best 10* contendo os 10 melhores resultados e um contendo os demais resultados (*Others*).

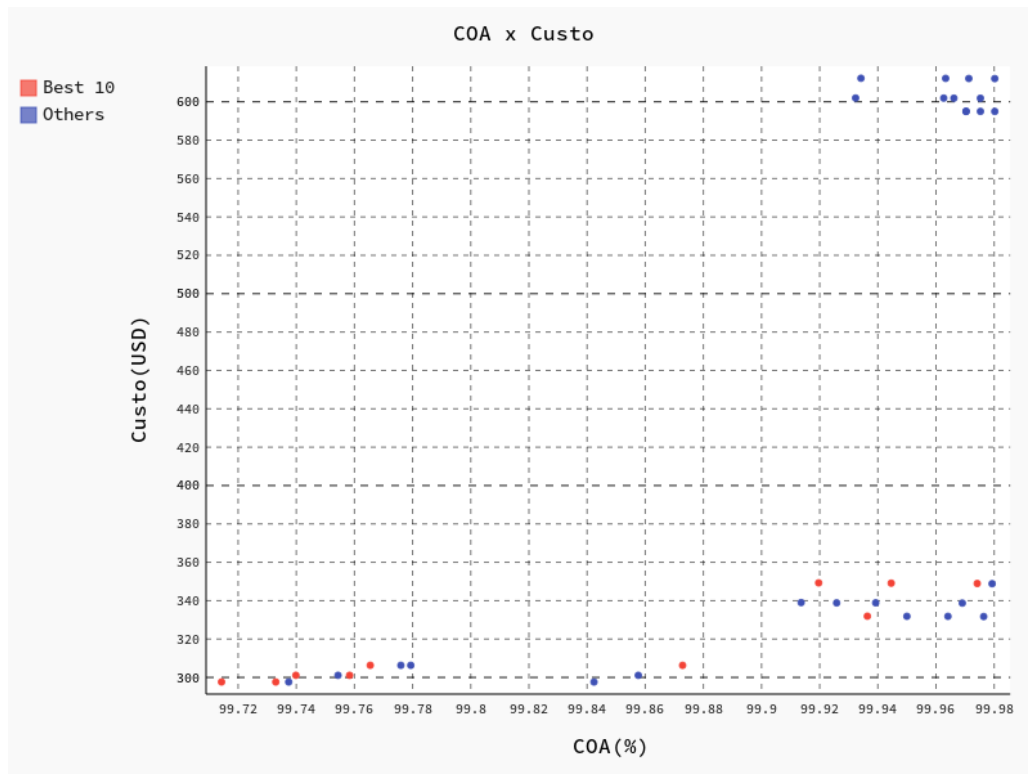
4.2 Caso 1

Tabela 4 – Ranqueamento dos resultados para COA e custo com peso 1.

Conf.	PROMETHEE	TOPSIS	Euclidiana
32	0,094364791636616	0,973938702817144	0,192732851710697
30	0,141991196591606	0,945441182550435	0,289448706053845
39	0,138005849595094	0,929776705842424	0,337128520628535
1	0,093905258016209	0,931281652390181	0,344262875231184
14	0,106006237218504	0,929318241183733	0,345948532281012
23	0,102631764924805	0,91364334590497	0,38630743654758
31	0,069981572087519	0,913629146584846	0,39606583112914
5	0,095649867375668	0,879543460592302	0,460248562670547
20	0,10442404634463	0,874502877405434	0,467831567974398
9	0,094690054565421	0,871169799178789	0,476125276487771

Fonte: Elaborada pelo autor.

Figura 7 – Scatter plot COA x Custo



Fonte: Elaborado pelo autor.

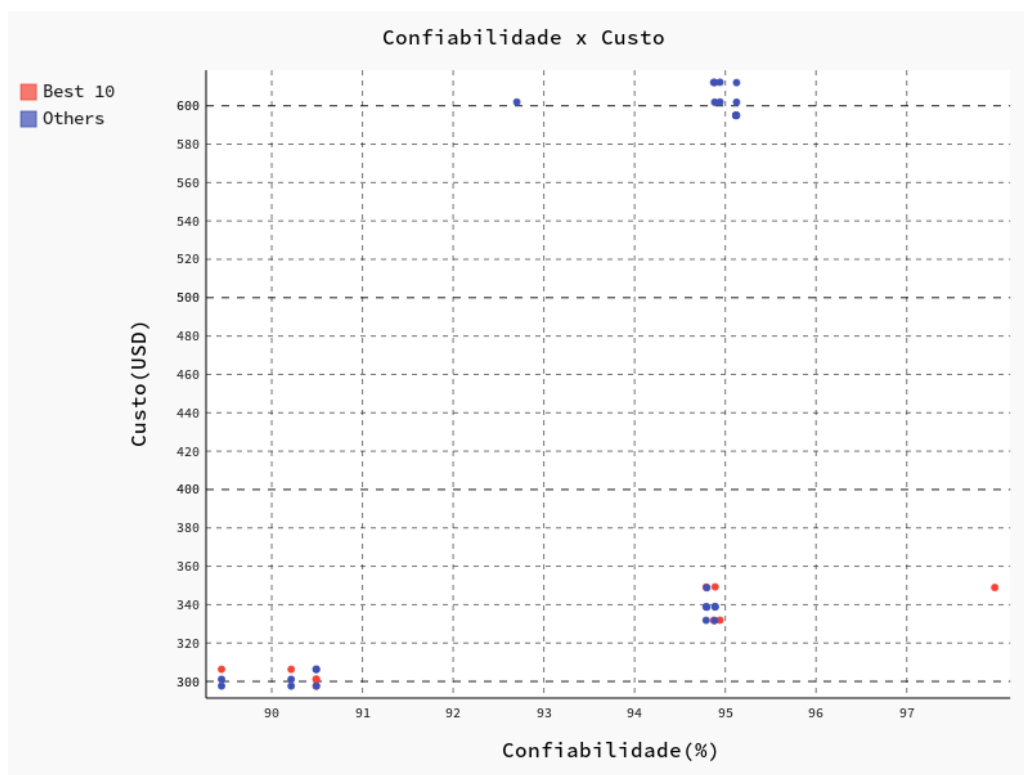
4.3 Caso 2

Tabela 5 – Ranqueamento dos resultados para confiabilidade e custo com peso 1.

Conf.	PROMETHEE	TOPSIS	Euclidiana
32	0,138912175963931	0,973938702817144	0,210765140333694
30	0,276897096907776	0,945441182550435	0,282983520995091
39	0,190043126125183	0,929776705842424	0,342049108068795
31	0,116270367298094	0,913629146584846	0,406228299842455
14	0,027063407074925	0,929318241183733	0,41559978584669
1	-0,004119949366418	0,931281652390181	0,430840767610531
19	0,034863758803698	0,913811804299005	0,444372395513879
23	0,034904934877673	0,91364334590497	0,444696239903959
40	0,21458914182392	0,849285644636143	0,506492722453982
20	0,036197874230139	0,874502877405434	0,516836082355305

Fonte: Elaborada pelo autor.

Figura 8 – Scatter plot Confiabilidade x Custo



Fonte: Elaborado pelo autor.

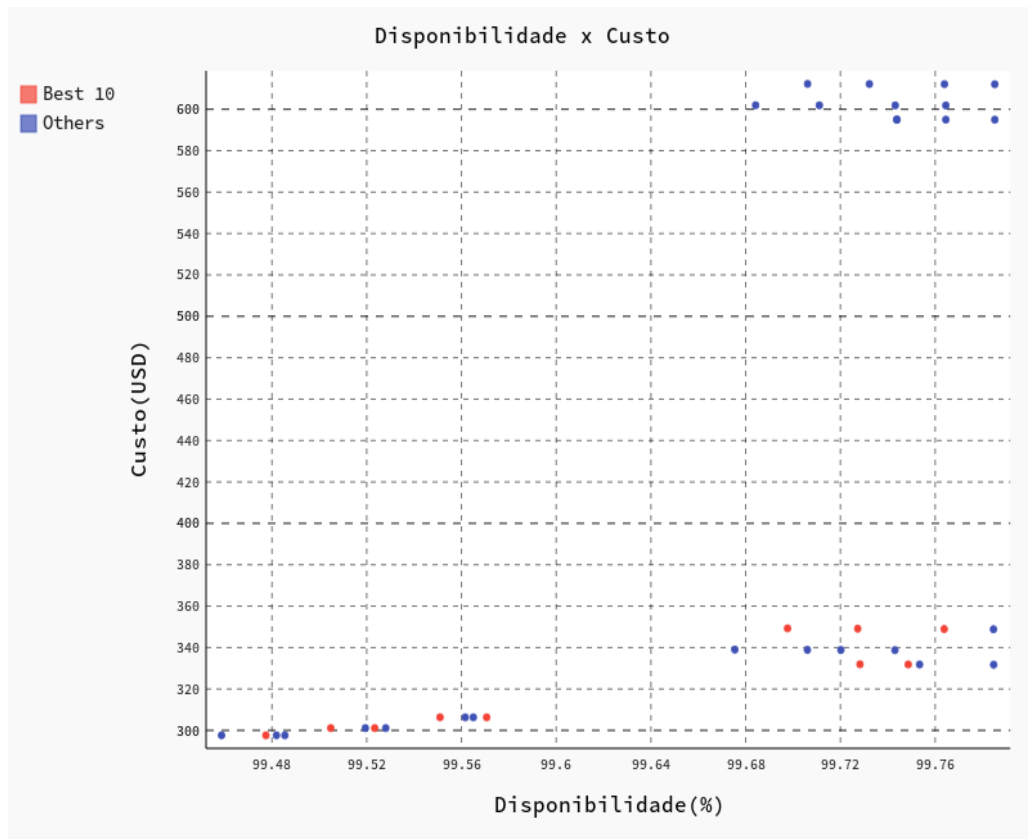
4.4 Caso 3

Tabela 6 – Ranqueamento dos resultados para disponibilidade e custo com peso 1.

Conf.	PROMETHEE	TOPSIS	Euclidiana
32	0,094776291590096	0,973938702817144	0,193142923240555
30	0,142610573985885	0,945441182550435	0,289558262685984
39	0,138680359071042	0,929776705842424	0,337218045026094
1	0,094389458107193	0,931281652390181	0,344470455305204
14	0,104231912481586	0,929318241183733	0,346775020506632
28	0,158582501513245	0,919430193531196	0,361450349696793
23	0,102061504227447	0,91364334590497	0,386754155270866
31	0,070257582251833	0,913629146584847	0,39635596020108
5	0,095604034481832	0,879543460592302	0,460525799567935
20	0,104385859347124	0,874502877405434	0,468080486978083

Fonte: Elaborada pelo autor

Figura 9 – Scatter plot Disponibilidade x Custo.



Fonte: Elaborado pelo autor.

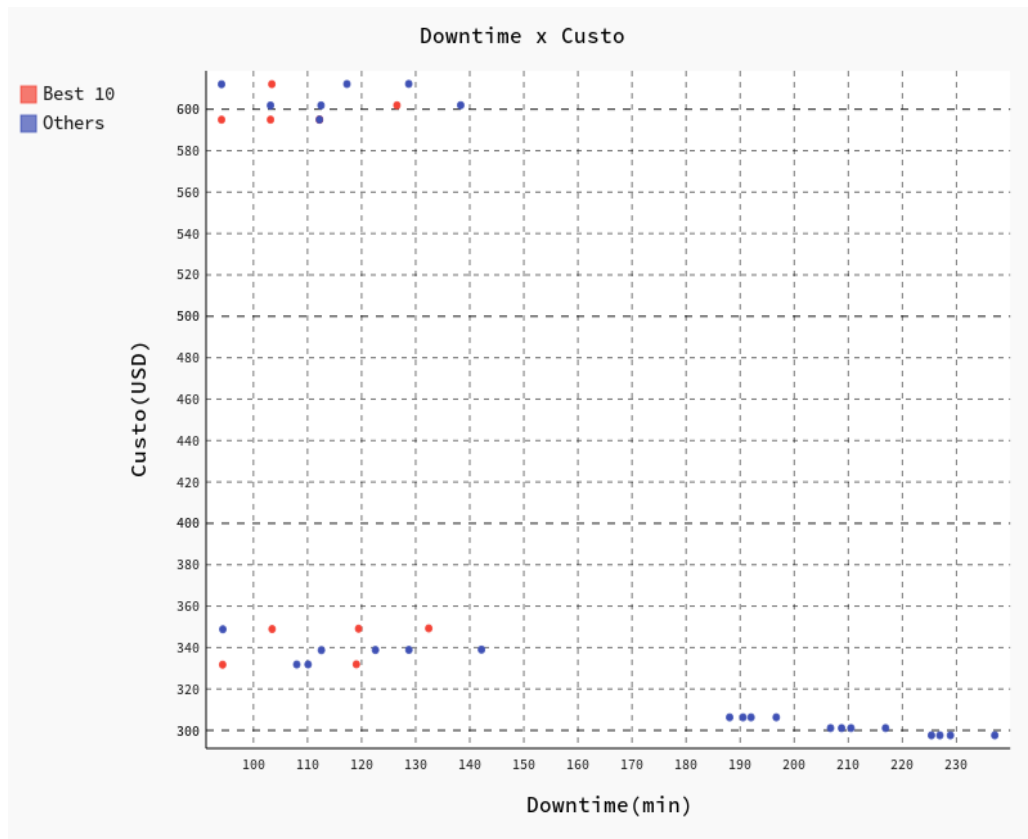
4.5 Caso 4

Tabela 7 – Ranqueamento dos resultados para *Downtime* e custo com peso 1.

Conf.	PROMETHEE	TOPSIS	Euclidiana
32	0,100689433800129	0,973938702817144	0,273182182860125
30	0,224313165574578	0,945441182550435	0,29956786369214
39	0,117389780266513	0,929776705842424	0,389316267356972
58	0,041107039850512	0,929803381745073	0,434333194845719
38	0,322604730839834	0,87752024778228	0,45154946301515
42	0,107187956105236	0,901967135985003	0,454750893629921
54	0,113927099977788	0,894224144374831	0,466742151574551
31	0,024274388142083	0,913629146584846	0,477725317208484
50	0,200791642886085	0,861700601155214	0,496235030185006
48	-0,056905160445055	0,931836872431504	0,499646378529141

Fonte: Elaborada pelo autor.

Figura 10 – Scatter plot Downtime x Custo



Fonte: Elaborado pelo autor.

4.6 Discussão

Analisando os resultados exibidos nas Figuras 7, 8, 9 e 10 percebe-se que nem todas as configurações contidas no grupo *Best 10* são melhores do que algumas contidas no grupo *Others*. Dentre os fatores que colaboram para este tipo comportamento ocorra são: o fato de que para algumas configurações possuem uma variação pequena no valor como ocorre com o atributo COA em que a variância é de aproximadamente de 0,0083 e isto faz com que os atributos com menor peso influenciem os resultados mesmo não constando nos gráficos.

Na maioria dos casos (1,2 e 3) as melhores configurações estão entre as que possuem custo mensal inferior a \$360 dólares porém isto não ocorreu no caso 4 (*downtime* x *Custo*), onde das 10 configurações melhor avaliadas temos 5 configurações com custo superior a \$590 dólares. No entanto, ao analisando a base de dados percebemos que as configurações com melhor *downtime* estão entre as mais caras, dentre as 12 configurações com menor *downtime*, 5 possuem um custo acima de \$590 dólares e as configurações com custo mais baixo possuem valores muito elevados para o *downtime*.

Outro fator a ser levado em consideração é a diferença entre escolher as melhores

configurações usando o PROMETHEE II ou o TOPSIS, isto se deve a maneira como cada método trabalha, enquanto o primeiro ele realiza uma comparação por pares entre as configurações o TOPSIS calcula a distância euclidiana da configuração a ser avaliada com uma solução ideal (os melhores valores dentre as possíveis soluções) e uma não-ideal (os piores valores dentre as possíveis soluções).

5 Conclusão

O propósito deste trabalho foi apresentar uma solução baseada na implementação de uma ferramenta considerando métodos multicritério baseado em preferências. Este trabalho adotou as técnicas PROMETHEE II e TOPSIS, para auxiliar na escolha de serviços na nuvem. Como foi mostrado ao longo deste trabalho, os métodos MCDM desempenham um importante papel no que diz respeito a auxiliarem as pessoas e empresas a escolherem soluções que melhor atendem suas necessidades.

A contribuição deste trabalho se deve a confecção de uma ferramenta que auxilie os tomadores de decisão a escolher uma solução de serviços na nuvem desde que ele tenha em mão os dados necessários (COA, confiabilidade, disponibilidade, *downtime* e custo).

Devido a limitações de escopo e de tempo seria interessante a realização de uma comparação entre resultados obtidos pela ferramenta apresentada neste trabalho com outras. Por fim, uma limitação da ferramenta se deve ao fato da implementação do PROMETHEE II não possuir funções de preferência ou deixar o usuário utilizar suas próprias funções.

Trabalhos futuros

A seguir são apresentados alguns pontos relevantes que podem dar continuidade a ferramenta apresentada neste trabalho:

- Adicionar mais opções de métodos MCDM como por exemplo o AHP e o ELECTRE deixando a critério do usuário escolher a escolha;
- Possibilitar que o tomador de decisão possa escolher quais atributos serão utilizados no ranqueamento;
- Adicionar uma opção para que o usuário possa utilizar suas funções de preferência no PROMETHEE II;
- Permitir que o tomador de decisão utilize dominância de pareto no lugar da distância Euclidiana para gerar o resultado final;
- Criar um banco de dados com os resultados para comparações posteriores;

Referências

- ABEDI S. ALI TORABI, G.-H. N. M. H. G.-R. E. M. Promethee ii: A knowledge-driven method for copper exploration. 2011. Citado na página 16.
- ALABOOL, H. et al. Cloud service evaluation method-based multi-criteria decision-making: A systematic literature review. *Journal of Systems and Software*, 2018. Citado na página 11.
- ARAUJO, J. Tomada de decisão multicritério em infraestruturas como serviço em nuvem: Uma abordagem baseada em modelos de dependabilidade, performabilidade e custo. 2019. Citado 3 vezes nas páginas 19, 25 e 28.
- BARABAS, J. *IaaS, PaaS and SaaS – IBM Cloud service models*. IBM, 2016. último acesso em 01/06/2018. Disponível em: <<https://www.ibm.com/cloud/learn/iaas-paas-saas>>. Citado na página 15.
- BOUGRINE, A. Topsis based multi-criteria reconfiguration of manufacturing systems considering operational. 2017. Citado na página 12.
- COLUMBUS, L. *2017 State Of Cloud Adoption And Security*. 2017. último acesso em 01/06/2018. Disponível em: <<https://www.forbes.com/sites/louiscolumbus/2017/04/23/2017-state-of-cloud-adoption-and-security/\52d31a8c1848>>. Citado na página 10.
- DANTAS, J. R. Planejamento de infraestrutura de nuvens computacionais para serviço de vod streaming considerando desempenho, disponibilidade e custo. 2018. Citado na página 19.
- GARG, S. K.; VERSTEEG, S.; BUYYA, R. A framework for ranking of cloud computing services. 2013. Citado na página 11.
- GRGUREVIĆ, G. K. I. Multi-criteria decision-making in cloud service selection and adoption. 2017. Citado na página 13.
- HAJKOWICZ, S.; HIGGINS, A. A comparison of multiple criteria analysis techniques for water resource management. 2008. Citado na página 12.
- KITTUR, J. Using the promethee and topsis multi-criteria decision making methods to evaluate optimal generation. In: *2015 International Conference on Power and Advanced Control Engineering (ICPACE)*. [S.l.: s.n.], 2015. p. 80–85. Citado na página 10.
- KROENKE ANDRÉIA DANI, D. d. S. N. H. A. Aplicação do método ahp no processo de seleção de fornecedores em uma empresa de santa catarina. 2016. Citado 2 vezes nas páginas 11 e 12.
- MICROSOFT. 2017. Último acesso em:07/06/2018. Disponível em: <<https://azure.microsoft.com/en-us/overview/cloud-computing-dictionary/>>. Citado na página 10.

RAO, R. V. *Decision Making in the Manufacturing Environment: Using Graph Theory and Fuzzy Multiple Attribute Decision Making Methods*. [S.l.: s.n.], 2007. Citado na página 17.

SABAEI, D.; ERKOYUNCU, J.; ROY, R. A review of multi-criteria decision making methods for enhanced maintenance delivery. 2015. Citado na página 11.

SANKET, T. *Cloud Analytics with Google Cloud Platform: An end-to-end guide to processing and analyzing big data using Google Cloud Platform*. [S.l.]: Packt Publishing, 2018. Citado na página 10.

WHAIDUZZAMAN ABDULLAH GANI, N. B. A. M. S. M. N. H. M.; HAQUE, I. T. Cloud service selection using multicriteria decision analysis. 2014. Citado 2 vezes nas páginas 11 e 12.

Apêndices

APÊNDICE A – Implementação do TOPSIS

```

def TOPSIS(array, weights, names=False):

    #1 Normaliza os dados
    norm = normalize(array)

    #2 Aplica os pesos
    for i in range (array.shape[1]):
        norm[:, i] = np.multiply(norm[:, i], weights[i])

    #3 Encontra o maximo e o minimo
    Vp = [np.amax(norm[:, col]) for col in range (norm.shape[1])]
    Vm = [np.amin(norm[:, col]) for col in range (norm.shape[1])]

    #4 Calcular medida de separacao (positiva)

    #4.1 lista com o quadrado da distancia entre Vij e Vj+
    Sp = [(norm[row, col] - Vp[col])**2 for col in
range(norm.shape[1]) for row in range (norm.shape[0])]

    #4.2 Transforma a lista anterior em numpy array
    Sp = np.array(Sp)
    Sp = Sp.reshape(norm.shape)

    #4.3 Somatorio das alternativas
    Sp = Sp.sum(axis= 1)

    #4.4 Raiz quadrada dos somatorios
    Sp = [np.sqrt(Sp[row]) for row in range (Sp.shape[0])]

```

```
#5 Calcular medida de separacao (negativa)

#5.1 lista com o quadrado da distancia entre Vij e Vj+
Sn = [(norm[row, col] - Vm[col])**2 for col in
range(norm.shape[1]) for row in range (norm.shape[0])]

#5.2 Transforma a lista anterior em numpy array
Sn = np.array(Sn)
Sn = Sn.reshape(norm.shape)

#5.3 Somatorio das alternativas
Sn = Sn.sum(axis= 1)

#5.4 Raiz quadrada dos somatorios
Sn = [np.sqrt(Sn[row]) for row in range (Sn.shape[0])]

#6 Calcular a aproximidade relativa
p = np.array([(Sp[i]/(Sn[i] + Sp[i] ) for i in range
(norm.shape[0]))])
n = np.array([(Sn[i]/(Sn[i] + Sp[i] ) for i in range
(norm.shape[0]))])

c = [math.sqrt(math.pow(n[i],2) + math.pow(p[i],2))
for i in range (norm.shape[0])]

#7 Organizar e ordenar os resultados

if (not(names)):
    names = [ 'Alternativa ' + str(rows) for rows in range
(norm.shape[0])]

df = pd.DataFrame(pd.DataFrame(names))
df["TOPSIS"] = pd.DataFrame(c)

df = df.sort_values(by=["TOPSIS"], ascending=False)

return df
```

Listing A.1 – Código do método TOPSIS

APÊNDICE B – Implementação do PROMETHEE II

```

def PROMETHEE_II(array, weights, names=False):

    #Confronta as alternativas
    array = normalize(array)
    row=[]
    for a in range (array.shape[0]):
        for b in range (array.shape[0]):
            parcial =[]
            for k in range (array.shape[1]):
                parcial.append(array[a,k]-array[b,k])
            row.append(parcial)
    valors = addWeights(np.asarray(row), weights)

    pi=[]
    for row in range(valors.shape[0]):
        pi.append(sum(valors[row])/sum(weights))

    pi = np.asarray(pi)
    rang= int(math.sqrt(pi.shape[0]))
    pi = np.reshape(pi, (rang, rang))

    #Calculo de sobre classificacao positiva
    pos=[]
    for i in range (pi.shape[0]):
        pos.append(sum(pi[i,:])/(pi.shape[0]-1))

    #Calculo de sobre classificacao negativa
    neg=[]
    for i in range (pi.shape[0]):
        neg.append(sum(pi[:,i])/(pi.shape[0]-1))

```

```
#Calcula o fluxo final
final =[]
for row in range(len(pos)):
    final.append(pos[row] - neg[row])

final = np.asarray(final)

# caso nao seja passado um nome para as alternativas
# eles serao gerados aqui
if (not(names)):
    names=['Alternativa '+str(rows) for rows in range
           (final.shape[0])]

df = pd.DataFrame(pd.DataFrame(names))
df["PROMETHEE"] = pd.DataFrame(final)

#7.2 Ordena o dataframe
df = df.sort_values(by=["PROMETHEE"], ascending=False)

return df
```

Listing B.1 – Código do método PROMETHEE II