

Seleção, pontuação e ranqueamento de provedores *serverless* utilizando método de Decisão Multicritério

Leandro Ribeiro Rittes

Universidade do Estado de Santa Catarina - UDESC
Joinville, Brasil
leandro.rittres1990@edu.udesc.br

Adriano Fiorese

Programa de Pós-Graduação em Computação Aplicada - PPGCAP
Universidade do Estado de Santa Catarina - UDESC
Joinville, Brasil
<https://orcid.org/0000-0003-1140-0002>

Abstract—As technology advances, it becomes increasingly challenging to identify the best approach or method to develop and distribute software that meets the ultimate goals of its creators and users, without becoming economically unfeasible and technically complex. Recognizing the relevance of opting for a third-party infrastructure solution (cloud computing) and the use of the *serverless* paradigm for such an approach, this study aims to propose a mechanism for selecting *serverless* platforms using a multicriteria method for decision making. The criteria used in modeling the solution were extracted from *serverless* service providers as well as from the analysis of *benchmarking* reports of *serverless* providers. Experiments regarding the accuracy and performance of the solution were carried out, together with the comparison of an implementation of the multicriteria method used available in a software library. As a result, it was identified that both implementations of the AHP algorithm obtained 100% accuracy in the results in a controlled environment. However, the algorithm implemented by the researchers presented a better performance in computation time in scenarios with more than 500 *serverless* providers.

Keywords—Serverless; Multi-criteria decision analysis; AHP.

Resumo—Conforme a tecnologia avança, torna-se cada vez mais desafiador identificar a melhor abordagem ou método para desenvolver e distribuir um software que atenda aos objetivos finais de seus idealizadores e usuários, sem se tornar economicamente inviável e tecnicamente complexo. Reconhecendo a relevância da opção por uma solução de infraestrutura de terceiros (nuvem computacional) e a utilização do paradigma *serverless* para tal abordagem, este estudo visa propor um mecanismo de seleção de plataformas *serverless* utilizando o método multicritério para a tomada de decisão AHP. Os critérios utilizados na modelagem da solução foram extraídos de provedores de serviço *serverless* bem como da análise de relatório de *benchmarking* de provedores *serverless*. Experimentos a respeito da acurácia e desempenho da solução foram realizados, juntamente com a comparação de uma implementação do método multicritério utilizado disponibilizada em uma biblioteca de software. Como resultado, foi identificado que ambas as implementações do algoritmo AHP obtiveram 100% de precisão nos resultados em ambiente controlado. No entanto, o algoritmo implementado pelos pesquisadores apresentou um

melhor desempenho em tempo de computação nos cenários com mais de 500 provedores *serverless*.

Palavras-chave—Serverless; Tomada de decisão multicritério; AHP.

I. INTRODUÇÃO

Conforme a tecnologia avança, torna-se cada vez mais desafiador identificar a melhor abordagem ou método para desenvolver um software que atenda aos objetivos finais pré-estabelecidos, sem se tornar economicamente inviável e tecnicamente complexo. A vasta área de desenvolvimento de software, incluindo front-end, back-end, DevOps, infraestrutura, entre outras, oferece centenas de tecnologias diferentes. A dificuldade na escolha da tecnologia ideal está relacionada, entre outras questões, às diversas características e critérios que podem afetar o desempenho do software em termos de custo, complexidade, manutenção, etc.

Especificamente na área de infraestrutura, existem diversas tecnologias disponíveis, algumas até redundantes. A escolha da tecnologia adequada pode ser influenciada pela disponibilidade de mão de obra especializada, que, em alguns momentos pelos quais passa o mercado de trabalho em tecnologia da informação, pode ser escassa e, portanto, economicamente inviável.

A terceirização dos serviços de infraestrutura, conhecida atualmente como computação em nuvem, representa uma estratégia eficaz para enfrentar diversos desafios computacionais, influenciando o desenvolvimento e a distribuição de software. Nesse modelo de computação em nuvem, o paradigma dominante de desenvolvimento de aplicações continua sendo o modelo de desenvolvimento para web, onde é possível disponibilizar a aplicação em servidores web (executando o protocolo *Hyper Text Transfer Protocol* (HTTP) e seus associados) em servidores virtualizados. Contudo, inclusive devido

à necessidade de controle de custos, novos paradigmas de desenvolvimento vêm surgindo, entre eles, o *serverless*.

O paradigma *serverless*, ou computação sem servidor, é um paradigma de desenvolvimento para a nuvem computacional que permite aos desenvolvedores criar e executar aplicações sem a necessidade de gerenciar servidores ou infraestrutura de *backend*. Esse novo paradigma de desenvolvimento permite que os desenvolvedores se concentrem exclusivamente na lógica de negócios e no código da aplicação, sem a necessidade de gestão (e custo de propriedade ou operação) dos servidores virtualizados na nuvem computacional. Nesse paradigma, a execução do programa de computador (muitas vezes chamado de função) ocorre sob demanda do usuário, o que caracteriza o pagamento pelo uso dos recursos do provedor de nuvem.

Assim, ao optar pela contratação de uma empresa que oferece serviço *serverless*, as empresas contratantes (desenvolvedoras e/ou implantadoras de software, bem como demais disponibilizadores de software em nuvem) podem externalizar os custos e responsabilidades relacionadas à manutenção e operação dos recursos de computação (ex.: servidores, equipamentos de interconexão de rede, etc.), liberando assim tempo e recursos para focar no desenvolvimento de software. Essa estratégia não apenas economiza dinheiro, mas também promove maior eficiência e qualidade no produto final, graças à capacidade da equipe interna de se dedicar integralmente ao seu trabalho.

Reconhecendo a relevância de optar por uma solução de infraestrutura de terceiros e a utilização do paradigma *serverless*, este trabalho visa propor um mecanismo de seleção de plataformas (provedores) *serverless* por meio da utilização de um método multicritério para a tomada de decisão, considerando diferentes indicadores relacionados à disponibilização de software nesse paradigma, associados aos provedores em questão. Esta análise auxiliará os usuários do serviço *serverless* (clientes dos provedores) a comparar vários fornecedores mediante diversos critérios importantes, simplificando a adoção de decisões fundamentadas e informadas sobre a escolha do provedor que melhor atende às suas necessidades específicas.

A estrutura deste trabalho está organizada da seguinte maneira: a Seção II apresenta os conceitos necessários à compreensão deste trabalho. A Seção III destaca alguns trabalhos que atuam sobre a problemática envolvendo provedores *serverless*. Na sequência, a Seção IV apresenta e discute a proposta de seleção por meio de pontuação e ranqueamento, utilizando a técnica de decisão multicritério desenvolvida. A seguir, a Seção V discute os resultados de experimentos realizados para validação da proposta. Finalmente, a Seção VI apresenta as considerações finais do trabalho realizado.

II. REFERENCIAL TEÓRICO

Esta seção apresenta conceitos e técnicas considerados essenciais a compreensão do trabalho.

A. O paradigma *serverless*

O paradigma *serverless*, ou computação sem servidor, é um paradigma de desenvolvimento que permite aos desenvolvedores criar e executar aplicações sem a necessidade de gerenciar servidores ou infraestrutura de *backend*. Embora os servidores físicos e/ou virtuais ainda sejam utilizados, os mesmos residem na infraestrutura do provedor de serviços em nuvem computacional. Nesse sentido, a gestão de tais recursos independe do cliente do serviço *serverless*, permitindo que os mesmos se concentrem exclusivamente na lógica de negócios e no código da aplicação [1], [2]. Dessa forma, o provedor de nuvem assume a responsabilidade pelo provisionamento, manutenção e escala da infraestrutura, oferecendo uma experiência de desenvolvimento e implantação de software mais eficiente e econômica. De acordo com Vieira [3] uma das principais características de *serverless* é a escalabilidade.

O paradigma *serverless* é caracterizado pela execução sob demanda das aplicações, onde os recursos computacionais são escalados automaticamente conforme necessário, e os custos são baseados apenas nos recursos utilizados durante a execução do código [3]. Isso significa que não há cobrança por capacidade inativa, resultando em economia significativa para as empresas.

Atualmente, todos os principais provedores de serviços em nuvem oferecem soluções *serverless*, incluindo AWS Lambda da Amazon, Azure Functions da Microsoft, Google Cloud Functions do Google e IBM Cloud Code Engine da IBM. Essas plataformas permitem que os desenvolvedores aproveitem os benefícios da computação sem servidor, facilitando a criação de aplicações nativas de nuvem escaláveis e eficientes.

A solução *serverless* representa uma evolução significativa na forma como as aplicações são desenvolvidas e executadas, oferecendo uma abordagem eficiente e econômica para o gerenciamento de infraestrutura. Isso torna-se mais evidente pois, ao permitir que os desenvolvedores se concentrem na lógica de negócios e no código da aplicação, enquanto delegam a gestão da infraestrutura aos provedores de nuvem, o paradigma *serverless* está transformando a maneira como as empresas operam e inovam no mundo digital.

B. Tomada de Decisão Multicritério

A Análise Multicritério para Tomada de Decisão (AMTD) do inglês *Multi-criteria Decision Analysis* (MCDA), é uma abordagem para solução de problemas de tomada de decisão que vem ganhando destaque no campo da tomada de decisões complexas. Essa abordagem é particularmente útil em situações

onde para a solução do problema existem diversas alternativas e não basta considerar apenas um único critério para tomar a decisão de qual é a mais adequada alternativa, mas sim múltiplos aspectos que podem, inclusive, ser conflitantes entre si. A aplicação da MCDA abrange diversas áreas, desde engenharia e gestão até meio ambiente, políticas públicas e saúde, demonstrando sua versatilidade e utilidade prática.

As vantagens do uso da MCDA são evidentes. Ela permite a consideração de múltiplos critérios, mesmo aqueles qualitativos ou intangíveis, fornecendo um processo de decisão transparente e auditável. Isso facilita a comunicação entre os tomadores de decisão e promove a tomada de decisões coerentes e embasadas em critérios objetivos.

No entanto, é importante reconhecer que o uso da MCDA também apresenta desafios. A aplicação pode ser complexa, exigindo conhecimento técnico especializado e tempo para coleta e análise de dados. A definição dos pesos para os diferentes critérios pode ser influenciada por valores pessoais ou visões de mundo dos decisores, introduzindo um elemento de subjetividade no processo. Além disso, nem todos os critérios podem ser facilmente quantificados, o que pode complicar a comparação entre as alternativas. E, finalmente, o processo de decisão multicritério pode ser demorado, especialmente quando envolve um grande número de alternativas e/ou critérios [4].

Apesar desses desafios, o valor da MCDA reside em sua capacidade de fornecer uma estrutura organizada para avaliar e comparar alternativas de forma transparente e sistemática. Ao permitir a consideração de múltiplos critérios, promover a transparência e encorajar a participação das partes interessadas, o MCDA facilita a tomada de decisões mais informadas, equilibradas e sustentáveis.

No âmbito da MCDA, a diversidade de abordagens disponíveis é vasta, cada uma adequada a diferentes tipos de problemas. Portanto, a seleção cuidadosa do método multicritério mais apropriado para a modelagem e solução do problema torna-se crucial. Este artigo concentra-se especificamente na tarefa de selecionar, por meio de pontuação e ranqueamento, provedores *serverless* a partir de critérios específicos. Com base nessa necessidade, o método escolhido foi o *Analytic Hierarchy Process* (AHP) [5].

1) *Analytic Hierarchy Process* (AHP): O *Analytic Hierarchy Process* (AHP), concebido por Thomas L. Saaty [5], surge como uma metodologia para auxiliar na tomada de decisões complexas. Este método, que se destaca pela sua capacidade de decompor problemas intrincados em componentes gerenciáveis através de uma hierarquia bem definida, oferece uma abordagem sistemática para avaliar e comparar múltiplos critérios e alternativas. Ao empregar comparações pareadas, o AHP não só facilita a determinação da importância relativa de cada critério

e alternativa, mas também calcula um peso ou prioridade para cada um deles, auxiliando para a escolha mais adequada.

Como citado em [6], o método AHP é amplamente reconhecido por sua simplicidade e facilidade de uso, o que contribui significativamente para sua grande difusão. Uma de suas principais forças é a notável flexibilidade, permitindo integração com outras técnicas como *Quality Function Deployment* (QFD) [7] e matriz *Strengths, Weaknesses, Opportunities, and Threats* (SWOT) [8]. A extensa pesquisa acadêmica e a vasta aplicação prática em diversas áreas conferem ao AHP alta confiabilidade e aceitação. Um aspecto particularmente valioso como citado em [6], [9]–[11] é a capacidade de medir a consistência interna dos julgamentos dos especialistas no problema sendo modelado, proporcionando maior robustez às decisões. Além disso, o método promove uma interação produtiva entre analistas e decisores, facilitando um entendimento comum do problema em questão. Esta característica é especialmente útil em ambientes de tomada de decisão colaborativa. Por fim, a síntese dos resultados oferecida pelo AHP permite uma comparação clara das prioridades e da importância relativa de cada fator considerado, fornecendo *insights* valiosos para o processo decisório.

Apesar de suas muitas vantagens, o AHP também apresenta algumas limitações importantes. Uma das principais é o limite máximo aconselhado de elementos de comparação, pois à medida que o número de comparações aumenta, o processo pode se tornar tedioso e menos confiável. A necessidade de estruturar o problema em uma hierarquia com atributos totalmente independentes entre si nem sempre é viável em situações reais complexas. Ainda, a subjetividade inerente às comparações pareadas, a potencial complexidade na gestão de problemas de grande escala e a sensibilidade dos resultados às mudanças nos pesos atribuídos aos critérios são aspectos que devem ser cuidadosamente considerados. Adicionalmente, a implementação efetiva do AHP pode demandar um certo grau de conhecimento especializado, destacando a importância de uma compreensão profunda dos princípios subjacentes ao método [10], [12]–[14].

Em suma, o *Analytic Hierarchy Process* representa uma abordagem robusta e estruturada para a tomada de decisões em ambientes caracterizados pela presença de múltiplos critérios. Ao proporcionar uma base quantitativa sólida, facilitando a análise comparativa e promovendo a participação inclusiva, o AHP promove decisões mais informadas baseadas na modelagem hierárquica entre objetivo, critérios disponíveis e alternativas de solução de um problema.

III. TRABALHOS RELACIONADOS

Diversos estudos na literatura abordam a correlação de provedores *cloud* e *serverless*, fornecendo *insights* valiosos para

a seleção, pontuação e ranqueamento desses serviços.

O trabalho de [15] propõe uma técnica de pontuação lógica e matemática para classificar e escolher o mais apropriado entre vários candidatos provedores de computação em nuvem para o usuário. Este método é baseado na análise de vários critérios que compreendem valores de indicadores de desempenho exigidos pelo usuário e associados a cada provedor de computação em nuvem que é capaz de atender às necessidades do usuário. Se baseando nesse estudo é possível compreender como obter e utilizar os indicadores de desempenho e como são utilizados em métodos já conhecidos. Porém, o fato do usuário necessitar popular o banco de dados com os valores dos PI's e ter de fazer ajustes finos em algumas constantes do método faz com que o resultado final possa não corresponder com o melhor aproveitamento do método. Dessa forma o método proposto neste trabalho tenta minimizar esses possíveis erros disponibilizando ao usuário do método apenas a necessidade de além do valor do critério desejado, pontuar os pesos dos mesmos.

Outro trabalho relevante na área de tomada de decisão em provedores de nuvem é o [16]. Naquele trabalho são utilizados modelos de regressão com a ferramenta H2O, para pontuar e ranquear provedores a partir da qualificação do usuário. O trabalho conclui que é possível pontuar e ranquear provedores de nuvem porém há um custo no tempo de resposta do modelo, pois o tempo de resposta é proporcional a quantidade de dados inseridos.

O trabalho de [17] realiza uma análise entre dois métodos de tomada de decisão; o TOPSIS e o FUZZY-TOPSIS, utilizando-os para a seleção de fornecedores para uma empresa de cabos de transmissão. Para a construção dos algoritmos foi utilizada a ferramenta MATLAB. O trabalho apresenta formas de comparar MCDMs em casos do mundo real, e também como minimizar erros quando há dados faltando, bem como minimizar a complexidade do algoritmo sem perder a confiabilidade do resultado.

IV. METODOLOGIA E DESENVOLVIMENTO

Este trabalho realiza uma pesquisa de natureza documental e quantitativa, que envolve uma análise aprofundada das informações disponíveis nos sites web oficiais dos principais provedores *serverless* e em relatórios de *benchmarking*. Em uma primeira etapa, o estudo busca identificar critérios/características associados ao serviço *serverless* que auxiliem na modelagem do problema de seleção do mais adequado provedor de serviço *serverless* tendo em vista as necessidades de seus clientes. Em uma segunda etapa, tendo já elencados os critérios, a metodologia executada realiza a implementação do método de seleção proposto, utilizando o MCDM AHP. Associado a este aspecto, o trabalho também

executa a comparação entre a implementação do método AHP desenvolvida, em termos de acurácia e tempo de execução (métricas de avaliação), com a implementação disponibilizada pela biblioteca PyDecision, como mecanismo de validação da implementação realizada.

A. Critérios para a Seleção

A seleção dos provedores *serverless*, neste trabalho, é realizada por meio do método AHP que se baseia em um conjunto de critérios ou Indicadores de Desempenho (PIs). Estes indicadores são fundamentais, pois servem como parâmetros para as avaliações quantitativas e qualitativas, permitindo a comparação e a pontuação das diferentes alternativas.

Podemos classificar os critérios ou PI's de acordo com a sua função e utilidade [15]. Ou seja, o quão útil o PI se apresenta de acordo com o objetivo final de uma análise. Nesse sentido, os tipos classificam os PI's de acordo com 3 possibilidades:

- **HB (Higher is Better):** usuários e gestores de sistemas preferem valores maiores em certos indicadores. Por exemplo, taxa de transferência do sistema, quantidade de recurso (dinheiro, memória, material, etc), disponibilidade do sistema, etc,
- **LB (Lower is Better):** usuários e gestores de sistemas preferem valores menores em certos indicadores. Por exemplo, tempo de resposta, custo, latência, etc.
- **NB (Nominal is Best):** usuários e gestores de sistemas preferem valores específicos, nem alto nem baixo. Por exemplo, carregamento para utilização do sistema, uma utilização alto do sistema pode gerar um tempo de resposta alto, em contra partida uma utilização baixa quer dizer que estão usando pouco o sistema.

Os valores dos PI's são coletados e armazenados com os tipos pois o método AHP os utiliza para construir a matriz de julgamento, tendo em vista o valor do PI oferecido/disponibilizado pelo provedor *serverless* e o requisitado pelo cliente.

Nesse sentido, para o caso do critério *Execution Time*, representando o intervalo de tempo mínimo de execução de uma função *serverless* para o qual é realizada a cobrança do cliente, pode-se deduzir, sob o ponto de vista da escolha do mais adequado provedor para o cliente, que seu tipo seja HB. Ou seja, quanto maior o valor de *Execution Time* disponibilizado pelo provedor, mais bem avaliado ele será pelo cliente.

Sendo assim, a definição dos critérios a serem utilizados para a seleção do mais adequado provedor *serverless* foi conduzida em duas etapas. Primeiramente, acessou-se diretamente os sites oficiais dos principais provedores *serverless* (AWS Lambda, Google Cloud Functions e Microsoft Azure), para observar as características que pudessem ser tomadas como critérios para escolha do provedor por um cliente, oferecidas pelos mesmos. Da mesma forma, nesse acesso, obtiveram-se

os valores mais recentes para tais características (PI's). Em seguida, complementou-se a análise com dados de *benchmark* de provedores *serverless* provenientes do relatório de Rifai [18], que apresenta uma avaliação detalhada de diversos aspectos relevantes para a escolha de um provedor *serverless*. Assim, com as características e valores extraídos nos sites oficiais e no relatório, os seguintes critérios foram utilizados para executar a seleção por meio do método AHP.

- **Tempo de computação:** se refere ao Tempo de execução¹ (segundos) multiplicado pelo Consumo de recurso convertido em GBs, o resultado é o valor mensal gratuito por mês que o usuário pode utilizar. Contabilizado em GB/segundo. Tipo: HB.
- **1GB/segundo adicional:** valor adicional para 1GB/segundo que o usuário necessitar no tempo de computação que passar da quantidade gratuita disponibilizada. Contabilizado em R\$. Tipo: LB.
- **Arredondamento da Duração:** valor inteiro mais próximo da duração de uma função. Ex: função durou 101ms, valor padrão de cobrança do provedor é 100ms, logo a duração será contabilizada como tendo durado 200ms. Contabilizado em milissegundo. Tipo: LB.
- **Requisição de graça/mês:** se refere a quantidade de requisições http gratuitas mensais que o provedor oferece ao seu cliente. Tipo: HB.
- **Requisição adicional:** se refere ao valor adicional para cada 1 milhão de requisições que ultrapassar a quantidade de requisições gratuitas por mês. Contabilizado em R\$. Tipo: LB.
- **Scalability:** se refere à capacidade do provedor de se ajustar automaticamente para lidar com um aumento ou diminuição na demanda de execução da função. Contabilizado em binário. Tipo: NB
- **Concurrency:** se refere à capacidade de executar múltiplas instâncias de uma função simultaneamente. Contabilizado em binário. Tipo: NB
- **Cold Start:** se refere ao tempo adicional para responder a uma solicitação de acesso à função em uma instância inativa². Contabilizado em segundos. Tipo: LB.
- **Memory:** se refere a quantidade de memória que o provedor oferece aos usuários para sua aplicação. Contabilizado em MB. Tipo: HB.
- **Execution Time:** se refere ao tempo máximo que um provedor ficará executando uma função. Contabilizado em minuto. Tipo: HB.

¹Tempo de execução é o resultado da quantidade de requisições multiplicado pela duração da execução (segundos)

²Instância inativa é uma instância que executou uma função e foi desalocada.

B. Parâmetros do Cliente

O cliente do provedor de serviço *serverless* apresenta suas necessidades ao método proposto por meio da parametrização dos critérios elencados para a seleção do provedor, na forma de uma requisição. Assim, a requisição do cliente representa a formalização das suas preferências e/ou necessidades em relação aos critérios de decisão, e por consequência em relação a escolha do provedor *serverless*. Através dela, o método AHP recebe as informações necessárias para construir as matrizes de decisão internas ao método, e assim realizar os cálculos e gerar um resultado de pontuação para cada provedor, de forma que reflita a hierarquia de preferência do usuário entre os mesmos.

A Tabela IV-B apresenta um exemplo de requisição de cliente ao método de seleção proposto. Nela o cliente indica os valores desejados para cada critério/PI envolvido na seleção, bem como o peso relativo de cada critério de acordo com sua necessidade.

TABELA I
FORMATO DE UMA REQUISIÇÃO

Critérios ou PI's	Valor	Peso
Tempo de computação	400.000 GB/s	1
1GB/segundo adicional	0,25 R\$	5
Arredondamento da Duração	1 ms	1
Requisição	1000	9
Requisição adicional	0,5 R\$	7
Scalabilidade	1	9
Concorrência	1	9
Cold Start	1s	9
Memória	512 MB	5
Tempo de Execução	30 min	9

Assim, os pesos atribuídos a cada critério refletem sua importância relativa na necessidade do usuário. Tais pesos são expressos utilizando a escala de Saaty [19] que designa valores de 1 a 9 para referir-se a importância de determinado critério ou valor em relação a um outro, em uma comparação pareada. Assim, o valor 1 indica a menor importância ao critério específico em relação aos demais, enquanto o valor 9 indica a maior. A Tabela IV-B apresenta a escala de Saaty.

C. Algoritmo AHP

Para o desenvolvimento da abordagem de seleção de provedores *serverless* o método AHP foi desenvolvido utilizando a linguagem de programação Python. Paralelamente, experimentos também foram conduzidos utilizando-se a implementação do método AHP disponibilizada pela biblioteca PyDecision. Essa estratégia teve por objetivo avaliar a confiabilidade da implementação desenvolvida, assumindo-se que a

TABELA II
ESCALA DE SAATY

Peso	Importância
1	Igual importância
3	Fraca importância
5	Forte importância
7	Muito forte importância
9	Importância absoluta
2,4,6,8	Valores intermediários

implementação da referida biblioteca é confiável, e portanto um *baseline* para os experimentos.

De forma a, originalmente, avaliar a acurácia das diferentes implementações, foram criados 5 cenários hipotéticos, cada um composto por 1 requisição de cliente e por 5 provedores *serverless* com valores predefinidos e classificação conhecida. Dessa forma, sabe-se o ranqueamento do mais ao menos adequado provedor para cada requisição e podemos avaliar os resultados da pontuação atribuída a cada provedor e conseqüentemente seu ranqueamento para cada implementação do método AHP. Essa abordagem permitiu comparar a acurácia das implementações, gerando como resultados que ambas as implementações são acuradas.

Cada cenário foi construído com o objetivo de simular uma situação real de seleção de provedor *serverless*. Em cada cenário foram atribuídos 3 PI's com "Importância absoluta" (valor 9 na escala de Saaty) e todo o restante com "Igual importância" (valor 1 na escala de Saaty).

Cada cenário consiste em uma única requisição e cinco provedores fictícios, cujos valores foram atribuídos de forma a gerar um ranking predefinido. Para a escolha do provedor ranqueado com a melhor pontuação foi atribuído a ele os 3 PI's com valor de importância máxima (9) e todos os valores esperados pela requisição.

O provedor configurado para ser a segunda melhor opção disponibiliza o valor esperado de 2 dos 3 PI's com máxima importância (9 na escala de Saaty), e o valor do terceiro critério é um valor abaixo ou acima do esperado, dependendo do tipo (i.e., HB, LB, NB), de acordo com a requisição do cliente, e o restante dos critérios recebem valores dentro de uma faixa pré-definida de valores mínimos e máximos, para todos os provedores, de importância 1.

Na seqüência, o provedor configurado para ser ranqueado como terceiro mais adequado disponibiliza o melhor valor de 1 dos 3 PI's mais importantes, e os ranqueados como quarto e quinto não disponibilizam nenhum dos 3 melhores e mais importantes critérios, sendo ranqueados pela pontuação gerada pelo AHP e em caso de empate, são ranqueados por ordem

alfabética.

A Tabela III apresenta o conjunto de dados relativos aos valores dos critérios e importâncias associadas aos fictícios provedores *serverless* para o Cenário 1 dos experimentos executados.

V. EXPERIMENTOS E RESULTADOS

Com o intuito de avaliar, o método de seleção proposto, incluindo o tempo de execução e a capacidade de escalabilidade das diferentes implementações do método AHP, experimentos foram desenvolvidos. Tais experimentos contam com um *range* de valores para os critérios obtidos de provedores e do relatório de *benchmark* provenientes de [18], como já mencionado. Esses valores para os critérios formam uma base de dados de onde são extraídos os limites para geração automatizada de valores para os critérios nos diversos cenários simulados. Nesse sentido, foi desenvolvido gerador de requisições e provedores fictícios. Esse gerador utiliza os dados disponíveis no banco de dados como base, selecionando aleatoriamente valores para os critérios. Com essa abordagem, foi possível simular cenários com um número variável de provedores, desde 5 até 1000, para cada uma das 100 requisições geradas aleatoriamente. Essa metodologia permitiu testar a capacidade dos algoritmos de lidar com um grande volume de dados e de realizar a seleção, pontuação e ranqueamento dos provedores de forma eficiente.

A Figura 1 ilustra o acréscimo de tempo de execução, de ambas implementações do método AHP, para o ranqueamento dos provedores, em função do crescimento do número de provedores.

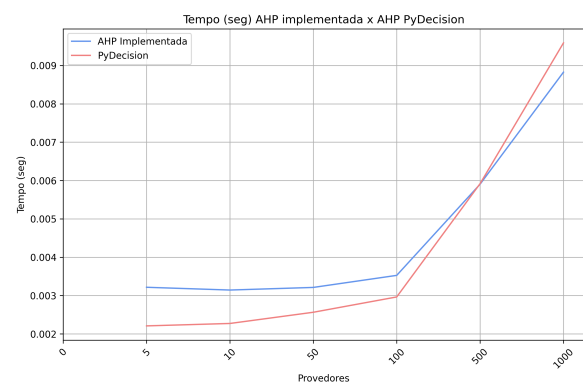


Fig. 1. Gráfico linha de tempo de execução x quantidade de provedores

Os resultados da análise de escalabilidade indicam que, para conjuntos de dados com mais de 500 provedores, o algoritmo AHP implementado nesse estudo demonstra superioridade em termos de tempo de computação. No entanto, para conjuntos de dados menores (menor quantidade de provedores), a biblioteca

TABELA III
PROVEDORES COM SEUS RESPECTIVOS VALORES DE ACORDO COM CENÁRIO 1

PI	Provedor									
	Provedor 1		Provedor 2		Provedor 3		Provedor 4		Provedor 5	
	Valor	Peso	Valor	Peso	Valor	Peso	Valor	Peso	Valor	Peso
Tempo de computação	400.000 GB/s	1	400.000 GB/s	1	400.000 GB/s	1	400.000 GB/s	1	400.000 GB/s	1
1GB/segundo adicional	0,25 R\$	1	0,25 R\$	1	0,25 R\$	1	0,25 R\$	1	0,25 R\$	1
Arredondamento da Duração	1 ms	1	1 ms	1	1 ms	1	1 ms	1	1 ms	1
Requisição	200000	9	200000	9	200000	9	100000	1	100000	1
Requisição adicional	0,5 R\$	1	0,5 R\$	1	0,5 R\$	1	0,5 R\$	1	0,5 R\$	1
Scalabilidade	1	1	1	1	1	1	1	1	1	1
Concorrência	1	1	1	1	1	1	1	1	1	1
Cold Start	1ms	9	2ms	1	2ms	1	2ms	1	2ms	1
Memória	512 MB	1	512 MB	1	512 MB	1	512 MB	1	512 MB	1
Tempo de Execução	30 min	9	30 min	9	5 min	1	5 min	1	5 min	1

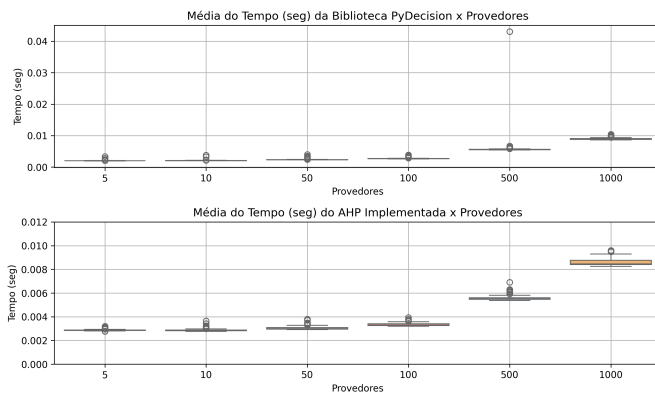


Fig. 2. Gráfico boxplot de tempo médio x quantidade de provedores

PyDecision, devido às suas otimizações, pode apresentar um desempenho mais eficiente. Essa observação sugere que a escolha do algoritmo ideal depende do tamanho do conjunto de dados a ser analisado. A Figura 2 apresenta os boxplots dos tempos de execução para a biblioteca e para o algoritmo implementado. Os *boxplots* permitem visualizar a distribuição dos tempos de execução, incluindo a mediana (linha central da caixa), os quartis (bordas da caixa) e os *outliers*, indicando pouca variabilidade mesmo com o aumento do número de provedores, para ambas as implementações.

As Figuras 3 e 4 apresentam os resultados de acurácia (dispersão e variabilidade, respectivamente) para a classificação de todos os provedores em um cenário com cinco provedores, cujas classificações (ranqueamento) corretas foram definidas a

priori. Ambas as figuras evidenciam a consistência do algoritmo implementado e da biblioteca PyDecision, apresentando 100% de acurácia em todos os cenários.

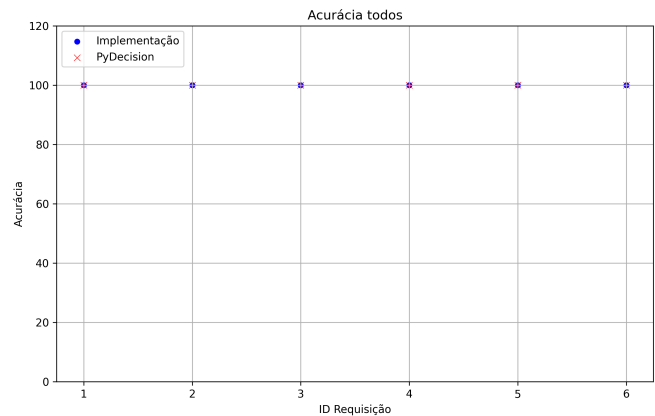


Fig. 3. Gráfico de dispersão de acurácia x requisição do cenário de teste

VI. CONSIDERAÇÕES FINAIS

Esse trabalho propôs um método para a seleção de provedores *serverless*, por meio de ranqueamento, baseado na análise multicritério para tomada de decisão, utilizando o método AHP. Na modelagem da solução do problema de seleção por ranqueamento, múltiplos critérios foram utilizados. Tais critérios modelam a relação de oferta do serviço *serverless* e a necessidade de seus clientes, em termos da implantação de aplicação de software como função nesses ambientes. O método que recebe como entrada os dados dos provedores *serverless* e

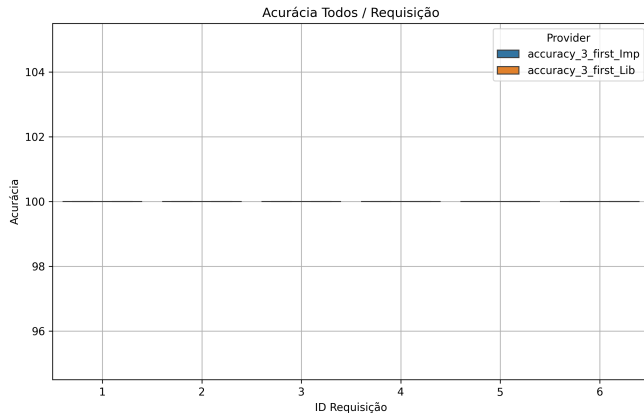


Fig. 4. Gráfico boxplot de acurácia x requisição do cenário de teste

uma requisição de cliente, baseada nos critérios utilizados e na importância dos mesmos para a sua necessidade do requerente.

Além da modelagem do problema, este trabalho realizou a implementação do método AHP, utilizando a linguagem de programação Python. Também foi utilizada uma implementação do método AHP disponibilizada pela biblioteca PyDecision que foi utilizada para a comparação de resultados com a implementação executada. Experimentos foram executados de forma a metrificar a acurácia e desempenho (tempo de execução) das implementações.

Os resultados desses experimentos demonstram que o algoritmo AHP implementado por esse trabalho apresentou um desempenho superior em termos de tempo de execução quando comparado à biblioteca PyDecision, especialmente em cenários com 500 ou mais provedores *serverless*. Ambos os algoritmos alcançaram 100% de acurácia na classificação dos melhores provedores em um ambiente controlado. Esses resultados indicam que ambos os métodos são promissores para a seleção de provedores *serverless* em cenários onde os critérios são conhecidos a priori.

No entanto, são necessárias pesquisas adicionais para avaliar a robustez desses métodos em cenários mais complexos e com maior variabilidade nos dados. Nesse sentido, propõem-se como trabalho futuros aumentar o número de casos de testes, gerando com isso uma confiança ainda maior em cenários hipotéticos com mais de 5 provedores, bem como a realização de comparações entre outros MCDM para obter uma maior confiabilidade nos resultados obtidos.

AGRADECIMENTOS

Esse trabalho teve o suporte financeiro da UDESC para sua divulgação e realização por meio da bolsa de Iniciação Científica PROBIC/UDESC.

REFERÊNCIAS

- [1] P. Castro, V. Ishakian, V. Muthusamy, and A. Slominski, "The rise of serverless computing," *Communications of the ACM*, vol. 62, no. 12, pp. 44–54, 2019.
- [2] J. Wen, Z. Chen, X. Jin, and X. Liu, "Rise of the planet of serverless computing: A systematic review," *ACM Transactions on Software Engineering and Methodology*, vol. 32, no. 5, pp. 1–61, 2023.
- [3] A. Vieira, G. Pantuza, J. Freire, L. Duarte, R. Pacífico, M. Vieira, L. Vieira, and J. Nacif, "Computação serverless: Conceitos, aplicações e desafios," in *Minicursos do XXXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, 1st ed., D. Gomes, I. Moraes, and M. Campista, Eds. SBC, Dec. 2020, pp. 190–236.
- [4] A. Ishizaka and P. Nemery, *Multi-criteria decision analysis: methods and software*. Chichester, West Sussex, United Kingdom: Wiley, 2013.
- [5] T. L. Saaty, "How to make a decision: The analytic hierarchy process," *European Journal of Operational Research*, vol. 48, pp. 9–26, September 1990.
- [6] G. A. Norris and H. E. Marshall, "Multiattribute decision analysis method for evaluating buildings and building systems," National Institute of Standards and Technology, Gaithersburg, MD, USA, Tech. Rep. NIST IR 5663, January 1995.
- [7] Y. Akao, *Quality Function Deployment: Integrating Customer Requirements into Product Design*. New York: Productivity Press, Oct. 2024.
- [8] R. W. Puyt, F. B. Lie, and C. P. M. Wilderom, "The origins of SWOT analysis," *Long Range Planning*, vol. 56, no. 3, Jun. 2023.
- [9] M. J. Liberatore and R. L. Nydick, "Group decision making in higher education using the analytic hierarchy process," *Research in Higher Education*, vol. 38, pp. 593–614, 1997.
- [10] V. P. Salomon, J. A. MONTEVECHI, and E. O. Pamplona, "Justificativas para aplicação do método de análise hierárquica," *Encontro Nacional de Engenharia de Produção*, vol. 19, 1999.
- [11] T. O. Boucher, O. Gogus, and E. M. Wicks, "A comparison between two multiattribute decision methodologies used in capital investment decision analysis," *The Engineering Economist*, vol. 42, no. 3, pp. 179–202, 1997.
- [12] H. Morita, T. Shimizu, and F. LAURINDO, "Modelos para estruturar e avaliar alternativas de decisão em tecnologia da informação," *ENEGEP- Encontro nacional de engenharia de produção*, vol. 19, 1999.
- [13] E. G. Gomes, F. R. Leta, R. B. V. Pessolani *et al.*, "Conceitos básicos do apoio multicritério à decisão e sua aplicação no projeto aerodesign," *Engvista*, 2003.
- [14] F. R. Guglielmetti, F. A. S. Marins, and V. A. P. Salomon, "Comparação teórica entre métodos de auxílio à tomada de decisão por múltiplos critérios," *Encontro Nacional de Engenharia de Produção*, vol. 23, 2003.
- [15] L. B. De Moraes and A. Fiorese, "A scoring method based on criteria matching for cloud computing provider ranking and selection," in *Enterprise Information Systems*, S. Hammoudi, M. Śmialek, O. Camp, and J. Filipe, Eds. Cham: Springer International Publishing, 2018, vol. 321, pp. 339–365, series Title: Lecture Notes in Business Information Processing.
- [16] K. Hopfer and A. Fiorese, "Seleção de provedores de nuvem através do auxílio de aprendizado de máquina automatizado," in *Anais da XXI Escola Regional de Alto Desempenho da Região Sul*. Porto Alegre, RS, Brasil: SBC, 2021, pp. 29–32.
- [17] F. R. Lima Junior and L. C. R. Carpinetti, "Uma comparação entre os métodos TOPSIS e Fuzzy-TOPSIS no apoio à tomada de decisão multicritério para seleção de fornecedores," *Gestão & Produção*, vol. 22, no. 1, pp. 17–34, Mar. 2015.
- [18] M. Rifai, "Serverless showdown: Aws lambda vs azure functions vs google cloud functions," <https://www.pluralsight.com/resources/blog/cloud/serverless-showdown-aws-lambda-vs-azure-functions-vs-google-cloud-functions>, 2023, accessed: 2024-12-08.
- [19] T. L. Saaty, "Decision making – the analytic hierarchy and network processes (ahp/anp)," *Journal of Systems Science and Systems Engineering*, vol. 13, pp. 1–35, March 2004.