

Seleção de um framework para desenvolvimento de ferramenta multicritério: aplicação do método AHP

Resumo: Este artigo explorou a aplicação do método *Analytic Hierarchy Process* (AHP) para selecionar o *framework* de desenvolvimento *backend* mais adequado para desenvolver ferramentas de decisão multicritério. Abordando a importância de uma seleção criteriosa em um cenário de desenvolvimento de *software*, o estudo utilizou uma revisão literária combinada com uma análise empírica para avaliar *frameworks* com base em critérios relacionados a desempenho, suporte da comunidade e facilidade de uso. A técnica de AHP foi empregada para decompor o problema de maneira hierárquica, facilitar comparações pareadas e sintetizar as preferências do decisor, resultando na identificação do *framework* que melhor equilibra as necessidades técnicas e estratégicas para aplicações de decisão multicritério. Neste estudo, foi consultado apenas um especialista, que será o próprio desenvolvedor da ferramenta multicritério. O estudo foi concluído com a escolha do *framework* Django, destacando sua adequação com base na análise dos critérios estabelecidos e sua popularidade robusta na comunidade de desenvolvedores.

Palavras-Chave: AHP; desenvolvimento web; multicritério.

Autores:

Marcos Alexandre Pinto de Castro Junior

im.alexandre07@gmail.com

Mestrando do PPGCC da Universidade do Estado do Rio de Janeiro (UERJ)

Igor Pinheiro de Araújo Costa

igorpin89@gmail.com

Gerente de Projetos no Centro de Análise de Sistemas Navais (CASNAV)

José Francisco Moreira Pessanha

pessanha@ime.uerj.br

Professor Doutor PPGCC da Universidade do Estado do Rio de Janeiro (UERJ)

Marcos dos Santos

marcosdossantos_doutorado_uff@yahoo.com.br

Professor Doutor da Escola Naval (EN)

1. Introdução

A seleção de um *framework backend* adequado é uma decisão crítica no desenvolvimento de *software*, particularmente para aplicações que exigem processamento complexo, como é o caso das aplicações de decisão multicritério (*Multiple Criteria Decision Method* – MCDM). MCDM é um subcampo de pesquisa operacional que lida com a tomada de decisões na presença de múltiplos critérios, muitas vezes conflitantes. Este tipo de aplicação requer um *backend* robusto, flexível e eficiente para gerenciar a complexidade inerente ao processamento de várias fontes de dados e critérios de decisão (Triantaphyllou, 2000). Assim, a escolha do *framework backend* torna-se não apenas uma decisão técnica, mas também estratégica, influenciando a eficiência do desenvolvimento, a performance da aplicação, e a facilidade de manutenção e escalabilidade (Chen; Okudan; Riley, 2010).

Para guiar esta escolha, foi aplicado o método *Analytic Hierarchy Process* (AHP), proposto por Saaty (1977). AHP é uma técnica consagrada de apoio à decisão que permite decompor um problema complexo em seus componentes, possibilitando avaliações pareadas para quantificar as preferências do decisor. A aplicabilidade de AHP em seleção de tecnologia foi demonstrada em diversos estudos, validando sua eficácia em ambientes de decisão complexos (Vaidya e Kumar, 2006).

Ao aplicar o AHP para selecionar um *framework backend*, diversos critérios devem ser considerados. Estes incluem, mas não se limitam a: performance do *framework*; sua comunidade e suporte; a facilidade de uso; a flexibilidade, a segurança; e a capacidade de integrar com outras ferramentas e sistemas (Rajaraman, 1994). A escolha do conjunto correto de critérios e a ponderação adequada de cada um são fundamentais para garantir que o *framework* selecionado atenda às necessidades específicas da aplicação de decisão multicritério.

Neste artigo, foi realizada uma revisão da literatura sobre o uso de MCDM e do método AHP, destacando estudos anteriores que aplicaram o AHP para selecionar ferramentas de *software*. Posteriormente, foi detalhado o método AHP, incluindo sua teoria fundamental e etapas de aplicação. Neste contexto, realizou-se a análise comparativa, o método AHP, combinando as preferências dos decisores com dados objetivos sobre os *frameworks*.

Neste sentido, o objetivo do presente trabalho foi selecionar o *framework web* mais indicado para desenvolver uma ferramenta de apoio à decisão multicritério, fornecendo ao especialista um método capaz de sintonizar tanto aspectos objetivos como subjetivos, quantificando suas preferências.

2. Fundamentação teórica

2.1. Métodos de Apoio Multicritério à Decisão

A tomada de decisão em ambientes complexos e multifacetados tem sido um desafio persistente para pesquisadores e profissionais em diversas áreas. O desenvolvimento de métodos de MCDM representa uma resposta a esse desafio, oferecendo uma estrutura para avaliar alternativas frente a múltiplos critérios, muitas vezes conflitantes. MCDM engloba uma família de métodos que facilitam a tomada de decisão quando os objetivos são muitos e não podem ser satisfeitos simultaneamente sem comprometimentos. Isso é especialmente relevante em cenários onde decisões estratégicas devem ser tomadas considerando uma ampla gama de impactos e interesses (Belton & Stewart, 2002).

Uma contribuição fundamental na área de MCDM foi a introdução do método AHP. Este método permite aos decisores decompor um problema de decisão em seus elementos constituintes, organizá-los de forma hierárquica, comparar elementos em cada nível da hierarquia de maneira pareada, e, por fim, derivar prioridades de decisão. O método é notável por sua capacidade de lidar tanto com dados quantitativos quanto qualitativos, facilitando a inclusão de julgamentos subjetivos e preferências no processo de decisão (Saaty, 1980).

A evolução da análise multicritério também tem sido marcada pela crescente incorporação de tecnologias de informação e sistemas de apoio à decisão. *Softwares* especializados e plataformas de decisão baseadas em *web* têm facilitado a aplicação de métodos MCDM em uma variedade de contextos, desde gestão ambiental até planejamento urbano e seleção de tecnologia (Mardani et al., 2015).

No entanto, a implementação efetiva de MCDM requer não apenas a seleção de métodos apropriados, mas também uma compreensão profunda dos problemas de decisão e dos contextos nos quais essas decisões ocorrem. Questões como a identificação correta de critérios, a ponderação adequada de diferentes fatores, e o gerenciamento de incertezas e preferências dos stakeholders são cruciais para o sucesso da tomada de decisão multicritério (Zavadskas et al., 2012).

2.2. Método Analytical Hierarchy Process (AHP)

O AHP é uma técnica influente de apoio à decisão multicritério que facilita a tomada de decisão complexa, ao decompor um problema em seus componentes fundamentais, organizá-los em uma estrutura hierárquica, e realizar comparações pareadas para quantificar as preferências. Esta metodologia é particularmente valiosa em situações onde decisões devem ser tomadas considerando uma série de critérios qualitativos e quantitativos, permitindo ao(s) decisor(es) estruturar problemas de forma lógica e coerente e realizar avaliações consistentes com base em julgamentos subjetivos e dados objetivos (Saaty, 1980).

O algoritmo do AHP começa com a definição do problema e a identificação dos critérios e alternativas relevantes, que são então estruturados em uma hierarquia de níveis que vão do

objetivo geral nas camadas superiores aos critérios e subcritérios nas camadas intermediárias, chegando às alternativas na base. A essência do método envolve realizar comparações pareadas entre os elementos de cada nível da hierarquia, com respeito a cada elemento no nível imediatamente superior, utilizando uma escala de importância que varia de 1 (igual importância) a 9 (extrema importância). Isso facilita a conversão de percepções subjetivas em um conjunto de números que podem ser usados para calcular pesos relativos para cada critério e alternativa, fornecendo uma base quantitativa para comparação (Saaty, 1977).

O cálculo dos pesos e a agregação das preferências em AHP são realizados através do método de autovalores, o qual determina um vetor de prioridades para os elementos comparados, refletindo sua importância relativa em atingir o elemento pai na hierarquia. O vetor de prioridades é derivado da matriz de comparação pareada, e sua consistência é avaliada para garantir que as avaliações sejam lógicas e coerentes. A consistência das comparações pareadas garante a validade dos resultados do AHP. Portanto, utiliza-se um índice de consistência para ajudar os usuários a avaliar e, se necessário, ajustar suas comparações a fim de melhorar a consistência (Saaty, 1980).

A flexibilidade e a generalidade do AHP tornam-no aplicável a uma ampla gama de problemas de decisão, desde escolhas pessoais até decisões estratégicas corporativas e políticas públicas. Seu poder de sintetizar dados qualitativos e quantitativos, juntamente com a habilidade de incorporar o julgamento e as preferências dos decisores, fornece uma fundamentação sólida para tomar decisões bem-informadas. Além disso, ao explicitar os critérios e as razões para as escolhas feitas, o AHP ajuda a aumentar a transparência e a compreensão do processo de decisão, facilitando a comunicação e o consenso entre diferentes partes interessadas (Vaidya e Kumar, 2006).

O método *Analytic Hierarchy Process* (AHP) tem sido amplamente aplicado na seleção de tecnologias em diversos campos, demonstrando sua eficácia e adaptabilidade. Por exemplo, Zolfani e Sapauskas (2013) empregaram o AHP na escolha de soluções de *software* para gestão de projetos, destacando como o método pode auxiliar na comparação objetiva entre opções complexas, incorporando critérios de eficiência, usabilidade e custo-benefício.

Segundo Kłos e Trebiina (2014), a seleção de um *Enterprise Resource Planning* (ERP) deve considerar os aspectos relacionados a custo total do ciclo de vida, suporte técnico e tempo de implementação. Neste estudo, foram considerados apenas opções de *software* de código aberto, não sendo considerados os aspectos relacionados a custos (tanto de implementação quanto de manutenção). Já o suporte técnico está intimamente ligado com a popularidade de um framework *open source* (Tulachan et al., 2018), sendo, portanto, utilizada como critério na avaliação. O tempo de implementação também pode variar de acordo com as necessidades de desenvolvimento de funcionalidades e, por isto, foi substituída pela facilidade de implementação percebida pelo especialista.

Face ao exposto, o presente estudo se mostrou viável, tendo em vista que existem exemplos na literatura aplicando o método AHP na seleção de *software* para situações específicas, como no caso de *software* ERP. Portanto, pode-se encontrar ampla validação para a aplicação do método utilizado, bem como trabalhos aplicando-o na área de tecnologia.

3. Método de pesquisa

3.1. Enquadramento metodológico

De acordo com a classificação metodológica proposta por Gil (2008) e Vergara (2016), a presente pesquisa pode ser caracterizada como aplicada, uma vez que busca gerar conhecimento voltado à solução prática de um problema específico: a seleção de um *framework web* para o desenvolvimento de sistemas. Quanto aos objetivos, a investigação é exploratória e descritiva, pois procura ampliar a compreensão sobre os critérios relevantes na escolha de *frameworks* e, ao mesmo tempo, retratar e organizar tais critérios de forma sistemática.

No que se refere à abordagem do problema, a pesquisa adota um método misto, conjugando aspectos qualitativos e quantitativos. O caráter qualitativo está presente na definição dos critérios e subcritérios fundamentados na literatura e na experiência do especialista consultado, enquanto o caráter quantitativo se manifesta na aplicação da metodologia *Analytic Hierarchy Process* (AHP), que permite atribuir pesos numéricos às preferências do decisor. Já do ponto de vista dos procedimentos técnicos, o estudo configura-se como um estudo de caso, pois aplica o AHP em um contexto específico de escolha de *framework web*, utilizando julgamentos de um especialista da área (Gil, 2008; Vergara, 2016).

3.2 Avaliação de Julgamentos

De acordo com Saaty (1991), o método AHP possui três princípios fundamentais: a decomposição do problema, a comparação de elementos e a síntese de prioridades. A decomposição do problema é aplicada para criar uma hierarquia de elementos em níveis independentes dos posteriores.

A metodologia avança aplicando o princípio de comparação, que envolve a criação de matrizes baseadas em comparações paritárias entre elementos, levando em conta o nível hierárquico imediatamente acima. Este processo assegura uma análise sistemática das preferências. Posteriormente, a etapa de síntese de prioridades é implementada, que percorre toda a estrutura hierárquica, atribuindo pesos específicos às matrizes. Este procedimento é feito de maneira cuidadosa, respeitando e refletindo as relações de importância estabelecidas nos níveis superiores da hierarquia, garantindo assim uma avaliação coerente e alinhada com os critérios definidos.

Neste contexto, partindo do objetivo, que é atribuir uma hierarquia aos *frameworks*, foram selecionados 3 critérios (primeiro nível): a popularidade (representada pelo número de pessoas contribuindo para o projeto), a performance do *framework* e a facilidade de implementação (percebida pelo especialista). Segundo Saaty (2008), o AHP consiste em quantificar, por meio de comparação, o julgamento dos indivíduos que participam da decisão, visando atribuir escalas de prioridade. Desta forma, foi criada uma matriz quadrada contendo os critérios tanto nas linhas quanto nas colunas. Ao final das comparações, esta matriz, denominada Matriz de

Julgamentos (MJ) deverá conter a prioridade relativa entre os critérios, de acordo com a avaliação de cada especialista (Vargas, 1990).

De acordo com Meniz e Özkan (2023), o AHP permite considerar medidas objetivas e de preferências pessoais entre critérios, sendo capaz de auxiliar uma análise que vislumbra aspectos tanto qualitativos quanto quantitativos. Desta forma é possível auxiliar a tomada de decisão de maneira mais abrangente, respeitando critérios de diversas naturezas.

Tabela 1. Escala de Saaty para comparação de elementos.

Intensidade da relação	Descrição	Definição
1	Mesma importância	Os elementos possuem relevância equivalente para o problema.
3	Importância pequena de uma sobre a outra	Um elemento possui julgamento levemente superior ao outro
5	Importância grande ou essencial	Um elemento possui julgamento fortemente superior ao outro
7	Importância grande ou demonstrada	Um elemento é muito fortemente superior ao outro, possuindo dominância experimentada na prática
9	Importância absoluta	Maior grau de dominância de um elemento em relação ao outro
2, 4, 6, 8	Valores Intermediários	Utilizados para manter relação entre duas definições

Fonte: Saaty (1991)

A tabela acima foi apresentada ao especialista, que atribuiu intensidades a cada par de elementos. A partir das comparações, foi construída a MJ. Os elementos da diagonal foram preenchidos com o valor 1, significando que um critério possui a mesma importância que ele mesmo. Já os elementos acima da diagonal receberam os respectivos valores de comparação. Por definição, aos elementos localizados abaixo da diagonal, foram atribuídos os valores inversos de acordo com a fórmula $a_{ij} = p, a_{ji} = \frac{1}{p}$.

De acordo com Abdullah et al. (2023), o próximo passo é normalizar MJ, obtendo, assim a Matriz de Julgamentos Normalizada (MJN). A MJN é obtida por meio da eq. (1)

$$MJN_{ij} = \frac{a_{ij}}{\sum a_{ij}} \quad (1)$$

Na eq. (1), MJN_{ij} corresponde ao valor de cada elemento da MJN, sendo a_{ij} a representação dos valores dos elementos da MJ.

O Vetor de Prioridade (VP) representa a prioridade de cada critério considerando os demais, sendo utilizado para obter a Matriz de Julgamentos Ponderada (MJP) a partir da MJ. Assim, utilizou-se a eq. (2) para atribuir aos elementos da MJP os respectivos valores. (Ávila et al., 2012; VARGAS, 1990):

$$MJP_{ij} = VP_i * a_{ij} \quad (2)$$

Onde:

MJP_{ij} : valor de cada elemento da MJP; e

VP_i : é o valor relativo da prioridade de cada critério.

Na etapa subsequente, o valor de $\lambda_{m\acute{a}x}$ foi calculado com base na equação (3). Com esse valor disponível, foi possível calcular o Índice de Consistência (IC) por meio da equação (4) e, adicionalmente, determinar a Razão de Consistência (RC) empregando a equação (5):

$$\lambda_{m\acute{a}x} = \sum \frac{MJP_i}{VP_i} \quad (3)$$

$$IC = \frac{(\lambda_{m\acute{a}x} - n)}{(n-1)} \quad (4)$$

$$RC = \frac{IC}{IR} \quad (5)$$

Neste contexto, $\lambda_{m\acute{a}x}$, o autovalor máximo, serve como indicador para aferir a consistência das preferências expressas na matriz de julgamentos. Este conceito é essencial para a análise, visto que um autovalor elevado sugere uma consistência maior nas comparações parietais efetuadas pelos julgadores (Saaty & Vargas, 2012).

Para a avaliação da confiabilidade das avaliações, a Razão de Consistência (RC) é empregada. Ela permite identificar se as comparações entre os critérios são consistentes ou se apresentam discrepâncias consideráveis. Uma RC reduzida indica que as comparações entre os critérios mantêm uma consistência lógica, enquanto uma RC elevada sinaliza a necessidade de revisar as avaliações devido à existência de inconsistências (Saaty & Vargas, 2012).

O Índice de Consistência Randômico (IR), por outro lado, oferece um parâmetro de comparação para normalizar o IC e calcular a RC. Esse índice é determinado por uma tabela que estabelece valores médios de consistência esperada para matrizes de avaliação de distintos tamanhos. Assim, é possível comparar o grau de consistência do julgamento realizado com aquele que

seria esperado ao acaso, o que é fundamental para assegurar a validade das avaliações (Saty, 1980; Forman & Gass, 2001).

Segundo Dolan (2008), o método AHP é flexível em relação à exigência de consistência perfeita nos julgamentos feitos por seus usuários, não sendo obrigatória a Razão de Consistência (RC) estritamente igual a zero. Essa tolerância permite uma certa margem de variação nas percepções e avaliações humanas, reconhecendo que a subjetividade e a complexidade das preferências podem impedir uma consistência absoluta.

Neste estudo, a validade dos julgamentos foi verificada com base nos valores de RC obtidos. Uma RC superior a 0,1 seria interpretada como um indicativo de que os julgamentos estariam enviesados. Essa condição sugeriria que as avaliações feitas pelo especialista na comparação par a par dos critérios e opções não alcançaram um nível de consistência aceitável, o que poderia levar à necessidade de uma revisão detalhada e possível reavaliação dos julgamentos efetuados. Por outro lado, um valor de RC inferior a 0,1 foi considerado como um sinal de que os julgamentos realizados pelos decisores para cada critério foram suficientemente coerentes Saaty (1980). Tal resultado demonstra que as preferências e comparações entre as opções seguiram uma lógica consistente, alinhando-se com a metodologia proposta pelo AHP.

3.3 Coleta de Dados

A metodologia empregada na fase descrita anteriormente concentrou-se exclusivamente na validação dos julgamentos aplicados aos critérios em análise. Essa etapa do processo envolveu uma verificação da coerência dos julgamentos pareados relativos a cada critério, garantindo assim a confiabilidade dos resultados obtidos. A medida utilizada para avaliar a consistência desses julgamentos foi o valor da Razão de Consistência (RC). Um valor de RC superior a 0,10 indicaria uma inconsistência significativa nos julgamentos, necessitando, portanto, de um retorno à etapa de avaliação pareada dos critérios para uma revisão e ajuste dos julgamentos realizados.

Segue-se então a próxima etapa, que adota uma abordagem quantitativa para a hierarquização das alternativas em consideração. Esta fase do processo foi fundamentada na análise de dados quantitativos e julgamento do especialista de acordo com a natureza de cada critério: popularidade, desempenho e facilidade de implementação do código. Utilizando esses dados, foi possível estabelecer uma classificação das alternativas, permitindo uma comparação direta com base em métricas objetivas.

Quando confrontados com a necessidade de comparar diferentes elementos dentro de um grupo, a natureza humana instintivamente busca ordená-los em uma hierarquia. O método de Análise Hierárquica de Processos (AHP) é projetado para capitalizar essa tendência, organizando as variáveis em uma estrutura hierárquica. Segundo Al-Saggaf et al. (2020), esta estrutura é dividida em três níveis principais: no topo, encontra-se o objetivo que se deseja alcançar; no nível intermediário, situam-se os critérios, que são as características utilizadas para avaliar as opções; e, na base, estão as alternativas, que são as opções disponíveis para escolha. Esta organização hierárquica facilita a representação visual, como ilustrado na Figura 1.

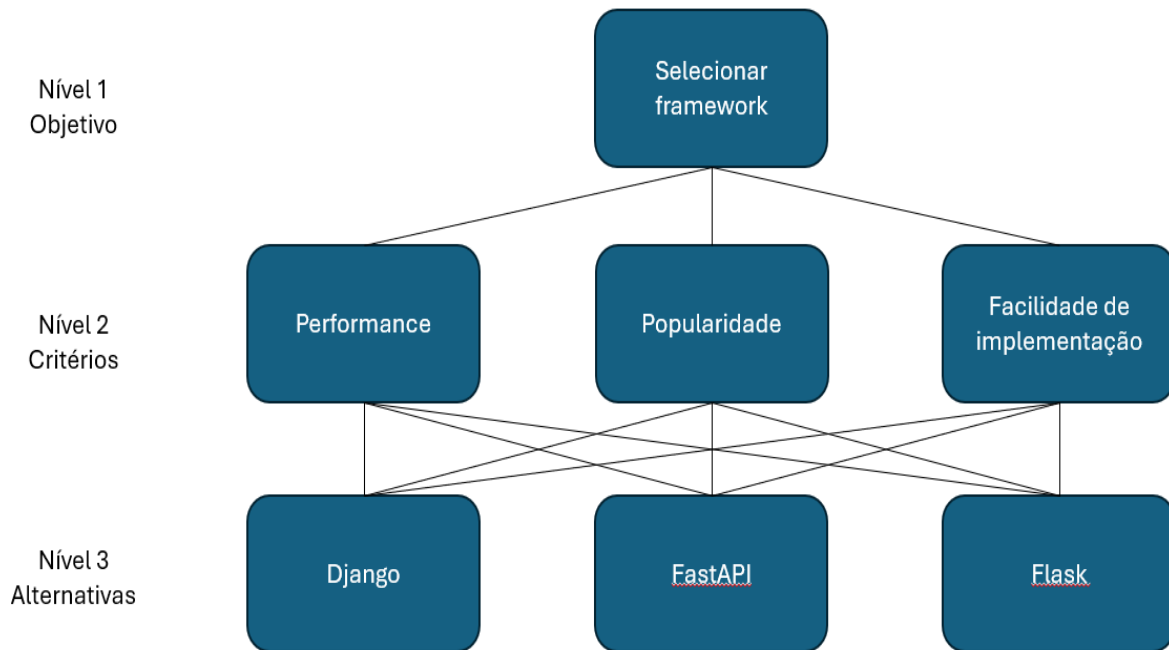


Figura 1. Representação gráfica da análise hierárquica de atributos.
 Fonte: Dados originais da pesquisa.

A Figura 1 representa a análise hierárquica do problema. A partir do objetivo da análise (selecionar *framework web*), os *frameworks* "Django", "Flask" e "FastApi" representam as alternativas, que foram avaliadas de acordo com os seguintes critérios "Popularidade", "Facilidade de implementação" e "Desempenho"

Como o critério "facilidade de implementação" possui caráter subjetivo, foi considerada a impressão subjetiva do especialista, atribuiu valores às comparações paritárias de acordo com a subseção anterior.

O objetivo do processo de normalização foi o de uniformizar a escala dos diferentes critérios utilizados na Análise Multicritério (MCA), culminando na criação da Matriz de Critérios e Alternativas Normalizada (MCAN). Essa normalização foi crucial para assegurar que, independentemente de um critério ser considerado como monotônico de lucro ou monotônico de custo, uma equação apropriada fosse aplicada para sua avaliação. Neste estudo, a quantidade de colaboradores do projeto foi considerada critério monotônico de lucro, pois quanto maior o valor, maior o suporte ao *framework* (Tulachan et al., 2018). Com isto, sua normalização será realizada de acordo com a eq. (6).

$$MCAN_{ij} = \frac{MCA_{ij}}{\sum MCA_j} \quad (6)$$

Onde $MCAN_{ij}$: são os valores normalizados dos atributos da Matriz; MCA_{ij} são os valores dos critérios da MCA e $\sum MCA_j$ são os valores dos somatórios das colunas da MCA.

A partir da normalização, obteve-se a MCAN, que, em combinação com o Vetor de Prioridades (VP), deu início ao procedimento de agregação. Neste processo, cada elemento da matriz normalizada foi multiplicado pelo valor correspondente no vetor de prioridades, conforme descrito na eq. (7), levando à formação da Matriz Agregada (MA).

$$MA_{ij} = MCAN_{ij} * VP_i \quad (7)$$

Finalmente, a determinação da prioridade entre as diferentes alternativas foi alcançada somando-se os valores de cada linha da Matriz Agregada (MA), com a alternativa que apresentou o maior total sendo identificada como a opção mais vantajosa entre os *frameworks*.

3.3 Avaliação de popularidade

A popularidade de um *framework web* de código aberto exerce um impacto significativo sobre o suporte e a manutenção do mesmo, principalmente devido ao número de contribuidores ativos que participam do seu desenvolvimento. Conforme destacado por Mockus et al. (2000), projetos de *software* de código aberto com uma base ampla de usuários tendem a atrair um maior número de desenvolvedores dispostos a contribuir, o que pode resultar em uma evolução mais rápida do *software*, correções de bugs mais ágeis e uma maior inovação. Isso se deve ao fato de que uma comunidade grande e ativa é capaz de oferecer suporte variado, desde a solução de problemas específicos até a implementação de novas funcionalidades.

Ademais, Raymond (1999) argumenta que "com olhos suficientes, todos os bugs são superficiais", sugerindo que o envolvimento de uma comunidade extensa no processo de revisão e contribuição pode levar a um *software* mais robusto e confiável. Assim, a popularidade de um *framework* não só facilita a obtenção de suporte por parte dos usuários, mas também potencializa a qualidade e a segurança do *software* por meio de revisões constantes e contribuições de uma comunidade diversificada de desenvolvedores.

Neste estudo, a análise da popularidade dos *frameworks web* foi fundamentada na quantidade de contribuidores que os respectivos repositórios detêm na plataforma *GitHub*. Essa métrica foi adotada por refletir não apenas o interesse da comunidade de desenvolvimento nos projetos, mas também por servir como um indicador de sua adoção e uso contínuos.

A escolha dessa abordagem se justifica pela relevância do *GitHub* como um dos principais ecossistemas de desenvolvimento de *software* livre, onde a interação e o engajamento da comunidade podem fornecer *insights* significativos sobre as tendências tecnológicas atuais (GitHub INC., 2024).

Dessa forma, os dados coletados oferecem uma perspectiva quantitativa sobre a popularidade e preferência entre os desenvolvedores, permitindo uma análise mais aprofundada sobre como esses *frameworks* são percebidos e utilizados no desenvolvimento *web* contemporâneo. A seguir, a tabela 2 representa os dados coletados da plataforma *GitHub*, demonstrando o quantitativo de contribuidores cada *framework* possui:

Tabela 2. Quantidade de colaboradores.

Flask	Django	FastApi
715	2508	643

Fonte: Github INC. (2024)

Na Tabela 2, pode-se observar que o *framework* Django possui um número de colaboradores maior que as outras alternativas, o que contribui para um melhor suporte do *software* pela comunidade. Esta relação se deve a uma quantidade superior de pessoas analisando e atualizando o projeto, garantindo correções mais tempestivas e reação mais rápida a possíveis vulnerabilidades.

3.4 Avaliação de desempenho

O desempenho da aplicação como um todo se deve, principalmente à implementação dos cálculos relacionados ao MCDM aplicado. Contudo, conforme a ferramenta ganha espaço na academia e no mercado, deve-se atentar para a experiência do usuário. Com isto, o desempenho e a capacidade de resposta do próprio *framework* também ganham importância.

O critério “desempenho”, constituído pela capacidade de respostas em formato JSON (*JavaScript Object Notation*) foi considerado um critério monotônico de lucro, pois quanto maior a capacidade de tráfego do *framework*, melhor o desempenho geral da aplicação. A seguir, a Tabela 3 representa os dados de um *benchmarking* realizado pela plataforma *TechEmpower* (TechEmpower, 2024), demonstrando a capacidade de respostas trabalhadas por segundo por cada *software*.

Tabela 3. Quantidade de respostas em formato JSON.

Flask	Django	FastAPI
1.561	73.479	154.061

Fonte: TechEmpower (2024)

Com isto, verifica-se que o “FastAPI” possui uma performance muito superior às demais alternativas. Para que um sistema tenha um bom desempenho em ambientes com demandas crescentes, é necessário possuir capacidade de lidar com um número significativo de requisições, respondendo no mínimo tempo possível. Neste caso, o FastAPI possui mais do que o dobro da capacidade de serializar respostas JSON do que os outros frameworks.

3.5 Avaliação da facilidade de implantação

A facilidade de implementação de um *framework web* é frequentemente considerada um critério subjetivo, variando significativamente de acordo com a experiência prévia, o conhecimento técnico e as preferências pessoais dos desenvolvedores. Segundo estudo realizado por Garousi et al. (2019), a percepção de facilidade ou dificuldade na adoção de um determinado *framework* pode ser profundamente influenciada pela familiaridade do desenvolvedor com as tecnologias subjacentes e com os paradigmas de programação empregados pelo *framework*. Esta

variabilidade na percepção destaca a importância de considerar a experiência individual e as preferências do especialista ao avaliar *frameworks web*, uma vez que o que é considerado intuitivo por um desenvolvedor pode ser percebido como complexo por outro, influenciando diretamente a curva de aprendizado e a produtividade do desenvolvimento.

Adicionalmente, a escolha de um *framework* não se baseia apenas em critérios objetivos como desempenho, escalabilidade ou segurança, mas também em aspectos subjetivos que incluem a facilidade de implementação e a compatibilidade com as habilidades existentes da equipe de desenvolvimento (Kabbedijk e Jansen, 2011). A preferência do especialista, portanto, desempenha um papel importante na seleção de tecnologias, com a escolha de um *framework* muitas vezes refletindo um equilíbrio entre as capacidades técnicas oferecidas e a facilidade percebida de adoção e uso contínuo. Esta perspectiva sublinha a relevância de avaliações detalhadas e a consulta a especialistas durante o processo de seleção, assegurando que as decisões tomadas se alinhem tanto com os objetivos técnicos do projeto quanto com as competências e preferências da equipe de desenvolvimento.

Desta forma, para quantificar as preferências do desenvolvedor, foi utilizada a escala de Saaty, representada na Tabela 1. O resultado aferido foi demonstrado na Tabela a seguir:

Tabela 4. Quantidade de respostas em formato JSON.

	Django	FastAPI	Flask
Django	1	7	5
Fastapi	1/7	1	1/3
Flask	1/5	3	1

Fonte: Dados originais da pesquisa.

Para obter a Matriz de Julgamentos Normalizada a partir dos valores fornecidos, precisamos seguir estes passos:

- Construir a Matriz de Julgamentos com os valores fornecidos.
- Calcular a soma de cada coluna da matriz.
- Dividir cada elemento da matriz pela soma da sua coluna correspondente para normalizar.

A tabela fornecida representa a comparação par-a-par entre Django, FastAPI e Flask. A partir dela, pode-se obter a matriz de julgamentos, aplicando a eq. (1).

Tabela 5. Matriz de Julgamentos Normalizada

	Django	FastAPI	Flask
Django	0.745	0.636	0.789
FastAPI	0.106	0.091	0.053
Flask	0.149	0.273	0.158

Fonte: Dados originais da pesquisa.

A partir da Matriz de Julgamentos Normalizada, o Vetor de Prioridades foi calculado mediante a média dos valores presentes em cada linha da matriz. Este procedimento visa representar o peso ou a importância de cada opção analisada (Django, FastAPI e Flask) dentro do contexto da análise de decisão empregando o método AHP. Os resultados obtidos são os seguintes:

Tabela 6. Vetor de Prioridades

Vetor de Prioridades	
Django	0.724
FastAPI	0.083
Flask	0.193

Fonte: Dados originais da pesquisa.

4. Resultados e Discussão

4.1 Avaliação da facilidade de implantação

De acordo com o descrito anteriormente, este estudo utilizou os dados de 3 frameworks de desenvolvimento web para análise, bem como as impressões do especialista com relação à comparação de critérios e facilidade de implementação. A tabela a seguir demonstra o resultado da comparação entre os critérios, de acordo com os valores apresentados na Tabela 1.

Tabela 7. Avaliação dos critérios

	performance	popularidade	facilidade
performance	1	1/5	1/7
popularidade	5	1	1/3
facilidade	7	3	1
Σ	1.343	6.333	11

Fonte: Dados originais da pesquisa.

A tabela demonstra que a diagonal foi preenchida com valor 1, indicando a equivalência de um critério em relação a ele mesmo. Acima das diagonais foram atribuídos os valores resultantes das comparações paritárias. Já na parte inferior a diagonal, a tabela foi preenchida com os valores inversos dos valores presentes acima.

A MJN foi obtida a partir da matriz de julgamentos, aplicando-se a equação 1.

Tabela 8. Matriz de Julgamentos Normalizada

	performance	popularidade	facilidade
performance	0.076923	0.047619	0.096774
popularidade	0.384615	0.238095	0.225806
facilidade	0.538462	0.714286	0.677419

Fonte: Dados originais da pesquisa.

A partir da MJN, foi obtido o VP, calculando-se a média aritmética de cada linha, resultando na tabela 9.

Tabela 9. Vetor de Prioridades dos Critérios

Vetor de Prioridades	
performance	0.073772
popularidade	0.282839
facilidade	0.643389

Fonte: Dados originais da pesquisa.

4.2 Avaliação com Dados das Alternativas

Após confirmar a coerência dos julgamentos relativos aos pesos atribuídos a cada critério, os dados da MCA foram normalizados, utilizando-se a eq. 7 e resultando na tabela 10.

Tabela 10. Dados Para Análise Multicritério

Alternativa	Performance	Popularidade	Facilidade
Django	0,321	0,649	0,724
Flask	0,007	0,185	0,083
fastAPI	0,672	0,166	0,193

Fonte: Dados originais da pesquisa.

Finalmente, ao aplicar a equação (8), a MA foi calculada, conforme apresentado na Tabela 10. A análise dos resultados revelou que o *framework* Django se destacou como a melhor opção para o desenvolvimento da ferramenta. Isso ocorreu porque, ao somar os critérios agregados, Django apresentou a maior pontuação total entre todas as alternativas consideradas. Essa conclusão destaca a utilidade da metodologia empregada na comparação e escolha do *framework* mais adequado baseado em uma avaliação sistemática e objetiva dos critérios relevantes.

Tabela 11. Matriz de agregação

Alternativa	Pontuação obtida
Django	0.6723
Flask	0.1771
FastAPI	0.1506

Fonte: Dados originais da pesquisa.

Da tabela 11, pode-se interpretar a superioridade do Django em relação aos outros *frameworks*. De maneira intuitiva, pode-se depreender que, pelo fato de o *software* ser superior em dois dos três critérios, provavelmente sua prioridade será mais elevada em relação aos demais. Este

resultado comprova, de maneira objetiva, as preferências do especialista em relação as alternativas.

6. Conclusões

Este estudo detalhou a aplicação do método AHP na seleção de um framework backend adequado para o desenvolvimento de ferramentas de decisão multicritério. Através de uma análise sistemática e metodológica, foram considerados critérios essenciais como desempenho, popularidade e facilidade de implementação, os quais foram avaliados por meio de comparações pareadas e ponderações quantitativas.

A escolha de Django como o framework mais adequado destaca sua robustez e adaptabilidade às necessidades complexas inerentes às aplicações multicritério. Este resultado não apenas confirma a eficácia do AHP como um método rigoroso de apoio à decisão, mas também fornece uma base sólida para futuras implementações e pesquisas na área de desenvolvimento de software, reforçando a importância de uma escolha metodológica criteriosa para enfrentar os desafios da tomada de decisão em ambientes tecnológicos complexos.

Este estudo se limitou a analisar *frameworks* baseados na linguagem *Python*. Portanto, como sugestão de estudos futuros, sugere-se analisar o emprego de diversas linguagens, comparando tanto as linguagens entre si quanto opções de *frameworks* baseados em linguagens diferentes. Esta análise pode adicionar mais um nível à hierarquia do problema, adicionando pesos às linguagens e avaliando as alternativas tanto em relação aos critérios utilizados neste estudo quanto em relação às linguagens de implementação.

Referências

ABDULLAH, Ade Gafar et al. Multi-criteria decision making for nuclear power plant selection using fuzzy AHP: Evidence from Indonesia. *Energy and AI*, v. 14, p. 100263, out. 2023.

AL-SAGGAF, Abdullah; NASIR, Hassan; HEGAZY, Tarek. An Analytical Hierarchy Process-based system to evaluate the life-cycle performance of buildings at early design stage. *Journal of Building Engineering*, v. 31, p. 101364, set. 2020.

ÁVILA, Paulo et al. Supplier's Selection Model based on an Empirical Study. *Procedia Technology*, v. 5, p. 625–634, 2012.

BELTON, Valerie; STEWART, Theodor. *Multiple criteria decision analysis: an integrated approach*. [S.l.]: Springer Science & Business Media, 2002.

CHEN, Ying; OKUDAN, Gül E.; RILEY, David R. Sustainable performance criteria for construction method selection in concrete buildings. *Automation in Construction*, v. 19, n. 2, p. 235–244, mar. 2010.

DOLAN, James G. Shared decision-making – transferring research into practice: The Analytic Hierarchy Process (AHP). *Patient Education and Counseling*, v. 73, n. 3, p. 418–425, dez. 2008.

FORMAN, Ernest H.; GASS, Saul I. The Analytic Hierarchy Process—An Exposition. *Operations Research*, v. 49, n. 4, p. 469–486, ago. 2001.

GAROUSI, Vahid; FELDERER, Michael; MÄNTYLÄ, Mika V. Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. *Information and Software Technology*, v. 106, p. 101–121, fev. 2019.

GIL, Antonio Carlos. *Métodos e técnicas de pesquisa social*. 6. ed. São Paulo: Atlas, 2008.

KABBEDIJK, Jaap; JANSEN, Slinger. Variability in Multi-tenant Environments: Architectural Design Patterns from Industry. In: [S.l.: S.n.]. p. 151–160.

KŁOS, Sławomir; TREBIINA, Peter. Using the AHP Method to Select an ERP System for an SME Manufacturing Company. *Management and Production Engineering Review*, v. 5, n. 3, p. 14–22, 29 set. 2014.

MARDANI, Abbas et al. Multiple criteria decision-making techniques and their applications – a review of the literature from 2000 to 2014. *Economic Research-Ekonomska Istraživanja*, v. 28, n. 1, p. 516–571, 11 jan. 2015.

MENİZ, Büşra; ÖZKAN, E. Mehmet. Vaccine selection for COVID-19 by AHP and novel VIKOR hybrid approach with interval type-2 fuzzy sets. *Engineering Applications of Artificial Intelligence*, v. 119, p. 105812, mar. 2023.

MOCKUS, Audris; FIELDING, Roy T.; HERBSLEB, James. A case study of open source software development. In: New York, New York, USA: ACM Press, 2000.

RAJARAMAN, V. *Computer programming in C*. [S.l.]: PHI Learning Pvt. Ltd., 1994.

RAYMOND, Eric. The cathedral and the bazaar. *Knowledge, Technology & Policy*, v. 12, n. 3, p. 23–49, 1999.

SAATY, T. L. *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*. [S.l.]: McGraw-Hill International Book Company, 1980.

SAATY, Thomas L. A scaling method for priorities in hierarchical structures. *Journal of Mathematical Psychology*, v. 15, n. 3, p. 234–281, jun. 1977.

SAATY, Thomas L. Some Mathematical Concepts of the Analytic Hierarchy Process. *Behaviormetrika*, v. 18, n. 29, p. 1–9, 15 jan. 1991.

SAATY, Thomas L. Decision making with the analytic hierarchy process. *International Journal of Services Sciences*, v. 1, n. 1, p. 83, 2008.

SAATY, Thomas L.; VARGAS, Luis G. *Models, Methods, Concepts & Applications of the Analytic Hierarchy Process*. Boston, MA: Springer US, 2012. v. 175

TRANTAPHYLLOU, Evangelos. *Multi-criteria decision making methods*. [S.l.]: Springer, 2000.

TULACHAN, Aakash et al. Development as a journey: factors supporting the adoption and continued use of software frameworks. *Journal of Software Engineering Research and Development*, v. 6, n. 1, p. 1–31, 2018.

VAIDYA, Omkarprasad S.; KUMAR, Sushil. Analytic hierarchy process: An overview of applications. *European Journal of Operational Research*, v. 169, n. 1, p. 1–29, fev. 2006.

VARGAS, Luis G. An overview of the analytic hierarchy process and its applications. *European Journal of Operational Research*, v. 48, n. 1, p. 2–8, set. 1990.

VERGARA, Sylvia Constant. *Projetos e relatórios de pesquisa em administração*. 16. ed. São Paulo: Atlas, 2016.

ZAVADSKAS, E. K.; TURSKIS, Z.; ANTUCHEVICIENE, J. Optimization of Weighted Aggregated Sum Product Assessment. *Electronics and Electrical Engineering*, v. 122, n. 6, 11 jun. 2012.