

Modelo Computacional para a seleção de pessoal baseado na técnica TOPSIS

Hugo Leonardo Deiró de Souza
hugodeiro@gmail.com
Instituto Federal da Bahia

Marcelo Vera Cruz Diniz
marcelodiniz@ifba.edu.br
Instituto Federal da Bahia

RESUMO

Este trabalho propõe um modelo híbrido que auxilie na tomada de decisão para a seleção de bolsistas dos projetos de pesquisa. O modelo computacional proposto integra as técnicas TOPSIS e Fuzzy com o objetivo de tornar estes processos seletivos mais eficientes e confiáveis. O caso de uso no Instituto Federal de Educação, Ciência e Tecnologia do Estado da Bahia (IFBA), onde os processos seletivos para os programas de fomento a pesquisa são geridos pela Pró-reitoria de Pesquisa, Pós-graduação e Inovação (PRPGI) do próprio instituto, foi realizado para fins de validação. Foram utilizados os resultados da seleção de três edições do Programa Institucional de Bolsas de Iniciação Científica (PIBIC). Estes dados foram adequados ao contexto do sistema e reanalisados levando em conta os critérios da PRPGI e a opinião de dezessete professores que ponderaram estes critérios, permitindo visualizar diferentes cenários de seleção. O modelo foi implementado em um software batizado como Topsisfy que busca englobar todo o percurso do decisor no processo de seleção de bolsistas.

Palavras-chave

Suporte à Decisão, Seleção de Bolsistas, TOPSIS, MCDA.

Agradecimentos

Em primeiro lugar, a Deus, por toda a força que me concedeu para percorrer o árduo caminho da graduação.

Em segundo lugar, a minha família, em especial meus pais, que sempre fizeram tudo que puderam para me dar o suporte necessário para alcançar os meus objetivos.

Em terceiro lugar, a meu Orientador, Marcelo Diniz, que foi de fundamental importância para mim ao longo deste processo de Trabalho de Conclusão de Curso.

Agradeço aos dezessete professores que se dispuseram a ajudar no desenvolvimento deste trabalho.

Agradeço também ao Laboratório de Inovações e Práticas Interdisciplinares e seus integrantes. O meu querido grupo de pesquisa, que tanto ajudou no meu desenvolvimento pessoal e teve grande influência no meu TCC.

Agradeço também a Gabriela Costa, Ícaro Jerry, Caique Correia, Milena Montenegro, Thiago Circuncisão, Lucélia Alcântara, Geisa Andrade, Filipe Cruz e Eduardo Teles, que me ajudaram bastante na revisão do trabalho.

Por fim, mas não menos importante, agradeço a todos os professores e servidores do Departamento de Computação e do curso de Tecnologia em Análise e Desenvolvimento de Sistemas. Em especial aos professores, que foram essenciais

para mim ao longo dessa caminhada que foi a graduação. Mais uma vez, muito obrigado pelo compartilhamento de seus conhecimentos e experiências que, sem dúvidas, permitiram me lapidar não só como profissional, mas também como pessoa.

1 Introdução

O desenvolvimento tecnológico e o fomento a pesquisa são bases estruturantes para que uma sociedade possa progredir. Os frutos oriundos do capital intelectual gerados por essas duas ações sociais têm impactos diretos em diversas áreas da sociedade, dentre elas, destaca-se a educação[12].

As instituições de ensino buscam promover ações de fomento à pesquisa, como o Programa Institucional de Bolsas de Iniciação Científica (PIBIC)[6]. Não distante desta realidade, o Instituto Federal da Bahia (IFBA) também provê bolsas para este e outros programas. Estas ações são reguladas através de processos seletivos coordenados pela Pró-reitoria de Pesquisa, Pós-graduação e Inovação (PRPGI) do próprio instituto[26].

Porém, de acordo com os dados extraídos do Sistema Stela Experta[15], o perfil dos professores que participam de grupos de pesquisa no IFBA *campus* Salvador está distribuído em oito áreas do conhecimento (Ciências Agrárias, Biológicas, Exatas e da Terra, Humanas, Sociais, Saúde, Engenharias, Linguísticas e Artes) e dois grupos cuja área de conhecimento não é identificada (Outras e Não Informado), ilustrado na Figura 1.

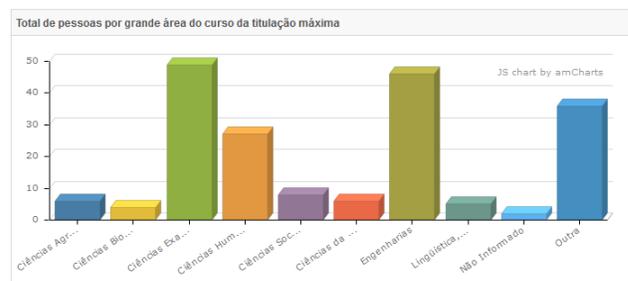


Figura 1: Total de Pessoas por grande área do curso da titulação máxima. Autor: [15].

A pluralidade criada pelas diferentes áreas de conhecimento dos professores pesquisadores do IFBA *campus* Salvador, evidencia as diferentes crenças e perspectivas acadêmicas e científicas desse corpo de pesquisadores. Esse fato permite questionar os critérios de avaliação utilizados pela

PRPGI, levando a seguinte pergunta: o processo de seleção do PIBIC é coerente com as opiniões e crenças dos professores da instituição?

Além disso, partindo da hipótese que os professores pesquisadores do IFBA têm diferentes perspectivas dos critérios de avaliação utilizados pela PRPGI, surge a segunda pergunta: o quão diferente seriam os resultados dos Editais 07/2013, 02/2014 e 01/2015, relativos ao processo seletivo do PIBIC, se fossem utilizadas diferentes perspectivas para definição do resultado final dos processos seletivos?

Esse trabalho faz parte de um projeto de pesquisa intitulado “Análise da Implantação dos cursos de Licenciatura do IFBA: Um estudo de caso da formação do professor de Física” e tem como principal objetivo validar a ferramenta Topsisfy, que será utilizada em outras investigações. Este projeto é mantido pelo Laboratório de Inovações e Práticas Interdisciplinares (LIPI), um dos grupos de pesquisa do Instituto.

Neste sentido, foi proposto um sistema que otimizasse este projeto pautado na técnica TOPSIS. Este *software* foi chamado de Topsisfy. A escolha desta técnica foi pautada na simplicidade e flexibilidade que a mesma propicia[58]. Para validar este sistema, foram utilizados os editais e resultados dos processos seletivos do PIBIC nos anos 2013-2014, 2014-2015 e 2015-2016 para coleta de dados e avaliação dos resultados baseado nos critérios ponderados pela própria PRPGI e por 17 professores da instituição que responderam um questionário no qual eles determinaram, com base em suas próprias análises, a importância de cada um dos critérios definidos pela Pró-reitoria.

Este trabalho está dividido nas seguintes seções: 2) Referencial Bibliográfico: onde são explicadas as técnicas e metodologias utilizadas ao longo deste trabalho de conclusão de curso; 3) Trabalhos Correlatos: são apresentadas publicações correlacionadas com o tema deste trabalho; 4) Metodologia: Nesta seção é demonstrado o passo a passo para execução do TOPSIS; 5) Software Topsisfy: nesta seção são apresentados aspectos arquiteturais e tecnologias utilizadas para o desenvolvimento do sistema Topsisfy; 6) Caso de Uso: É apresentado um caso de aplicação da TOPSIS em processos seletivos através do sistema Topsisfy; 7) Resultados e Discussões: São apresentados os resultados obtidos na aplicação da TOPSIS e uma discussão sobre os mesmos; 8) Considerações Finais: São apresentadas as conclusões sobre o trabalho e tópicos para melhoria em trabalhos futuros.

2 Referencial Bibliográfico

2.1 Suporte à Decisão

O processo de decisão é uma área de estudos que considera as etapas e meios utilizados para deliberações acerca de um problema[50]. Existem quatro etapas, ilustradas na Figura 2, que descrevem a tomada de uma decisão: Prospeção (análise de um problema ou situação que requer uma solução); Concepção (criação de alternativas que solucionem o problema); Análise (julgamento das soluções propostas); e Decisão (escolha de uma proposta para atender necessidade)[13].



Figura 2: Representação Visual do Processo de Decisão. Fonte: Adaptado de [13].

As pessoas que estão envolvidas no processo são chamadas de atores e podem estar qualificadas de duas formas: passivo, quando não influencia diretamente na tomada de decisão; ou ativo (também chamado de decisor) quando é o responsável por decidir qual solução será aplicada para resolver um problema[13].

O processo decisor segue basicamente dois paradigmas, o racionalista e o construtivista. No primeiro, presume-se que todos os envolvidos possuem o mesmo objetivo, nível de conhecimento e acesso à informação[8]. Desta forma, quaisquer desvios podem ser considerados irracionalidades que são capazes de pôr em risco todo procedimento. Este paradigma é contraposto por Roy e Vanderpooten (1996), que consideram a subjetividade como um fator determinante para que o decisor realmente esteja integrado no processo de tomada de decisão[47].

No paradigma construtivista os aspectos subjetivos do decisor, como valores pessoais, objetivos, cultura e intuição, são essenciais para a tomada da decisão[8]. Esta incorporação da subjetividade traz consigo distinções relevantes e acentuadas quando contrastada ao paradigma racionalista, fazendo com que este paradigma tenha um caráter qualitativo muito maior do que a sua contraparte.

Analisando a Figura 3, observamos que ambos os modelos têm necessidades em comum, como o acesso à informação. Contudo, também existem aspectos específicos para os dois paradigmas. No caso do modelo racional, é necessário que os decisores tenham o mesmo objetivo e nível intelectual[34]. Já no paradigma construtivista, estas características racionalistas podem existir, mas não são mandatórias. Porém, os aspectos pessoais do decisor são importantíssimos, por influenciarem diretamente a decisão final[14].

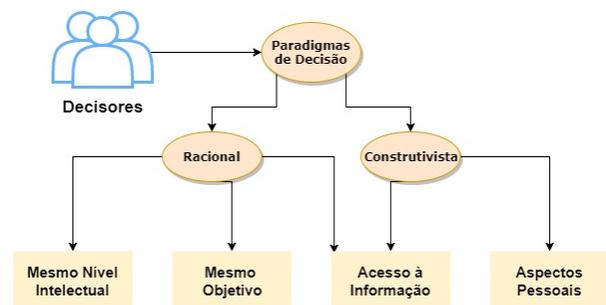


Figura 3: Representação Visual dos Paradigmas de Decisão. Fonte: Adaptado de [34] e [14].

Devido aos problemas relacionados à tomada de decisão, surgiram os Métodos Multicriteriais de Análise de Decisão (*Multi-criteria Decision Analysis* – MCDA). É importante ter em mente que, apesar do nome, a MCDA não se resume

apenas a uma coleção de teorias, técnicas e métodos. Ela é, na realidade, uma perspectiva específica para lidar com a resolução de problemas relacionados ao processo de decisão, onde devem ser considerados múltiplos critérios[20].

Existem diversas técnicas que estão contidas no escopo da MCDA, a exemplo de: *Multi-Attribute Utility Theory* (MAUT)[20], *Measuring Attractiveness by a Categorical Based Evaluation Technique* (MACBETH)[20], *Analytic Hierarchy Process* (AHP)[20] e a Tomada de Decisão Interativa Multicritério (TODIM)[58]. Contudo, dado o contexto deste trabalho, considerou-se que a *Technique for Order Preference by Similarity* (TOPSIS) seria a melhor opção. Esta decisão se deu pelo fato desta técnica viabilizar a utilização de múltiplos critérios que podem ser qualificados como mais ou menos importantes no processo tendo, assim, diferentes pesos e viabilizando de forma fácil a análise de cenários alternativos, bem como verificar as opções/soluções que ficam mais próximas do perfil ideal[49].

A TOPSIS opera a partir de um conjunto de opções que devem seguir critérios definidos no início do processo. Cada critério pode ter diferentes pesos na seleção. A partir daí, as respostas das opções são operadas de forma a compor a melhor e a pior situação possível dentro do escopo inserido. Estas situações são chamadas de soluções ideais positiva e negativa[13]. Após esta etapa, cada opção é ranqueada de acordo com a sua distância relativa à solução ideal positiva e negativa[58].

Apesar de sua alta confiabilidade, a TOPSIS está sujeita às possíveis imprecisões causadas por falhas humanas no momento de definir a importância de cada critério ou de verificar como cada opção atende aos critérios propostos. Estas falhas podem distanciar os resultados apresentados pelo sistema da realidade do problema. A fim de atenuar este risco, foram utilizados aspectos da Lógica Fuzzy para facilitar a avaliação das opções em cada critério, atribuindo-lhes escalas verbais como: insuficiente, baixo, abaixo da média, média, acima da média e alto ao invés de valores numéricos[3].

Apesar de ser mais utilizada na Engenharia devido a sua fácil compreensão e aplicabilidade, a técnica atraiu a atenção de pesquisadores de diversas áreas[58]. Existem diversos trabalhos que permitem demonstrar o potencial da TOPSIS, alguns destes destacados na na seção 3.

2.2 Desenvolvimento de Softwares

No âmbito do desenvolvimento de *softwares* existem diversos conceitos importantes, dentre os quais destacam-se banco de dados, *front-end* e *back-end*.

2.2.1 Desenvolvimento Front-end

A parte visual é um dos aspectos mais importantes de um sistema computacional. Isso acontece porque, para o usuário final, a camada de visão é o próprio sistema. Ele não precisa saber que há toda uma infraestrutura para que o *software* possa funcionar. Independente do quão bem feito e/ou estruturado seja um sistema, para o usuário final o que realmente vale é a interface, pois é o que ele pode ver, ouvir ou tocar[19].

O estudo das interfaces de usuário está incluso como uma subárea do campo de estudo Interação Humano-Computador (IHC), que estuda como planejar as ferramentas computacionais de forma a trazer satisfação para o usuário[19].

As interfaces evoluíram bastante com o passar do tempo. Originalmente, a interação com o usuário era feita através da *Command Line Interface* (CLI), o que implica em diversas questões como: requer certo domínio da tecnologia por demandar comandos específicos para executar suas atividades; É altamente propensa a erro humano; e também não é nada intuitiva. Com o passar do tempo, surgiram as *Graphical User Interfaces* (GUI), que revolucionaram a usabilidade dos sistemas computacionais, pois, o que era feito através de diversos códigos em linha de comando se reduziu ao clicar de um botão[19].

Com a expansão da *World Wide Web*, foi necessário repensar os conceitos oriundos das GUIs, pois, agora seria necessário proporcionar ao usuário uma boa experiência de navegação, além de outros atributos como consistência, integração dos conteúdos, segurança e confiabilidade[19]. Baseado nestes conceitos, surgiu uma nova vertente para as interfaces gráficas, as Web GUIs.

A interação do usuário com o sistema deve ser a mais simples possível[2]. Baseado nisso, o autor Jacob Nielsen, tido como o pai da usabilidade, propôs dez heurísticas que podem ajudar no desenvolvimento de uma boa interface.

A primeira das heurísticas trata sobre a visibilidade do estado em que o usuário está no sistema, que é responsabilidade de informar o que está acontecendo em tempo real para o usuário. Por exemplo, em plataformas de *streamming* de vídeo, convém ter o título do vídeo, independente dele estar sendo reproduzido ou não[2].

A segunda das heurísticas discute a correspondência entre o sistema e o mundo real, ela discute sobre manter uma relação o mais próxima possível com o mundo real para considerar sons e sinais visuais que o usuário normalmente encontra fora do sistema. Por exemplo, utilizar a cor vermelha para elementos negativos[2];

A terceira heurística aborda a liberdade de controle fácil para o usuário, onde busca permitir ao usuário fazer o que ele quiser dentro do sistema, com exceção, claro, das ações que vão contra as regras de negócio ou outras funcionalidades[2].

A quarta heurística fala sobre a consistência ao longo do sistema. Discute como é importante manter a consistência e padrão visual (texto, cor, ícones, sons, entre outros) ao longo do *software*. Por exemplo, um botão de submeter um formulário deverá ter sempre o mesmo tamanho, cor e texto, para evitar erros do usuário[2].

A quinta heurística abrange a prevenção de erros. Neste ponto, busca-se criar interfaces que evitem ao máximo a ocorrência de erros. É muito melhor não permitir ao usuário errar do que apresentar uma mensagem de erro para o mesmo[2].

A sexta heurística visa fortalecer o reconhecimento ao invés de memorização. O usuário não tem obrigação de decorar o caminho que fez até chegar em uma determinada tela do sistema. É interessante utilizar algum recurso que apresente para o mesmo o percurso que ele fez até aquela tela, por exemplo[2].

A sétima heurística prima pela flexibilidade e eficiência de uso. Neste ponto, é discutido sobre como é importante deixar uma experiência boa com o sistema desde o usuário mais leigo até o mais avançado. Um exemplo prático desta heurística é a utilização de atalhos. Algumas atividades podem ser feitas de uma forma mais longa, porém intuitiva, enquanto estas mesmas atividades poderão também ser feitas utilizando atalhos. Por exemplo, para fechar uma tela,

o usuário poderia ir até o botão vermelho com um “X” no canto superior direito ou esquerdo da tela, ou simplesmente utilizar o atalho “Alt + F4”[2].

A oitava heurística fala sobre a estética e design minimalista. Neste aspecto, indica-se que *layout* e conteúdo devam estar disponíveis da forma mais simples possível para não deixar o usuário com nenhuma dúvida[2].

A nona heurística busca ajudar os usuários a reconhecerem, diagnosticarem e recuperarem-se de erros. Apesar das heurísticas de Nielsen proporem fazer o tratamento de erro para evitar apresentar mensagens para o usuário, existem momentos em que isso não é possível. Desta forma, é necessário que as mensagens de erro sejam claras, próximas ao conteúdo ou ação que causou o erro[2].

Por fim, a décima heurística fala sobre ajuda e documentação para o usuário. O ideal é não precisar utilizar esta heurística. Contudo, a depender do porte do sistema, pode ser necessário. Nestes casos, a documentação deve estar visível para o usuário, preferencialmente próxima do elemento que tenha necessidade de uma explicação mais detalhada[2].

Estes conceitos foram agrupados pela indústria no termo *front-end*. Desta forma, tem-se que o objetivo do desenvolvimento *front-end* é criar interfaces *web* que sejam claras, fáceis e rápidas para o usuário. É necessário haver uma preocupação com a informação, colocando-a sempre em um contexto adequado para propiciar uma boa experiência de uso para seus usuários [21].

Os desenvolvedores *front-end* são capacitados a desenvolver aplicações focando na interação com o usuário. Estes desenvolvedores buscam sempre aprimorar a usabilidade e experiência do usuário. São estes profissionais que ligam o usuário ao *back-end* (que será explicado na próxima subseção), sempre coletando dados alimentados por um usuário e levando-os para o tratamento no outro lado da aplicação [24].

Durante muito tempo, a área de *front-end* se preocupou basicamente em entregar conteúdo para o usuário, mas não com a forma que este conteúdo era recebido pelo mesmo. Devido a isso, surgiu uma nova área de estudo no âmbito da IHC, a *User Experience* (UX), que busca propiciar ao usuário uma experiência completa, de forma que ele se sinta confortável ao utilizar a ferramenta e que o faça de forma intuitiva [57].

2.2.2 Desenvolvimento Back-end

Back-end é um termo cunhado na indústria que refere-se a camada do sistema em que são implementadas as regras de negócio apresentadas através dos requisitos da aplicação. É neste âmbito do sistema que boa parte das diretrizes de segurança, performance e integração com outros sistemas (como Receita Federal, por exemplo) são seguidas[18]. Os profissionais desta área são responsáveis por desenvolver como o sistema irá operar os dados trazidos através do *front-end*, bem como retornar estas informações para o usuário[18]. Contudo, esta área não demanda apenas o desenvolvimento do sistema, mas também toda uma infraestrutura de recursos computacionais a serem utilizados pela aplicação.

É um ramo crítico, pois implica diretamente no negócio e na performance da aplicação. As regras de negócio definem o *back-end*. Entretanto, independente das diferenças entre diferentes aplicações, um código desta camada de negócios deverá primar sempre por manter os padrões baseados nas boas práticas de mercado e em manter os códigos o mais

limpo, bem organizados e compreensíveis possível[18].

Conforme supracitado, o *back-end* não se resume apenas a códigos da lógica de negócio. Existe toda uma infraestrutura física para dar suporte ao sistema como servidores, armazenamento, redes e afins. Por muito tempo, o dimensionamento destes recursos foi um problema, contudo, com o advento e pluralização da *Cloud Computing*, isso mudou[45].

De acordo com o *National Institute of Standards and Technology* (NIST), “*Cloud Computing* é um novo modelo para viabilizar o acesso a um conjunto de recursos de forma simples, flexível e sem grande impacto financeiro para o cliente”. Com esta definição, é possível identificar algumas características que tornam este modelo bastante atrativo para gestores de negócio, como: investimento de acordo com a demanda; preço operacional menor; alta escalabilidade; facilidade de acesso; e redução de riscos e gastos com manutenção[45].

A nuvem é estruturada em quatro camadas, conforme ilustra a Figura 4. Estas camadas são descritas por [45] como: Hardware: responsável por gerenciar todos os recursos físicos da *cloud*, a exemplo de roteadores, *switches*, sistemas elétricos e de refrigeração; Infraestrutura: também conhecida como camada de virtualização. Esta camada disponibiliza o acesso aos recursos computacionais através do particionamento dos recursos físicos; Plataforma: construída sobre a camada de infraestrutura, é esta camada que contém os sistemas operacionais e *frameworks* das aplicações; e Aplicação: Nível mais alto da hierarquia. É nesta camada em que são disponibilizadas as aplicações de *cloud*.

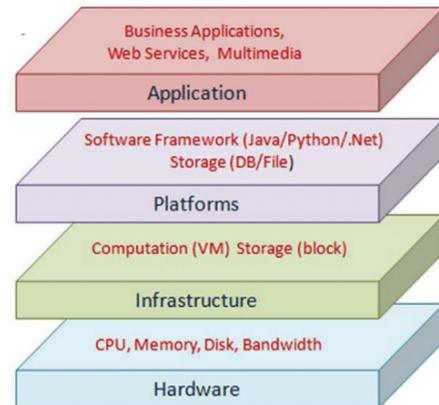


Figura 4: Arquitetura de Cloud Computing. Fonte: [45]

Com a pluralização da *Cloud Computing* surgiram novos modelos de negócio que abrangem desde a infraestrutura até o *software* em si.

O primeiro destes modelos é o *Infrastructure as a Service* (IaaS), que refere-se ao provisionamento da infraestrutura e recursos necessários para utilização do cliente. Normalmente, a IaaS disponibiliza o acesso a estes serviços através de máquinas virtuais[45].

O segundo modelo que veio com a *Cloud Computing* foi o *Platform as a Service* (PaaS), que se refere a camada de recursos utilizados na infraestrutura, o que inclui Sistema Operacional, *softwares* e *frameworks*, por exemplo[45].

O último destes modelos é o *Software as a Service* (SaaS), que se refere ao provisionamento de aplicações de *cloud* através da internet[45].

Estes modelos de negócios se relacionam de forma que acaba havendo uma camada de infraestrutura, que abarca os modelos IaaS e PaaS que interagem diretamente com a camada de provisão de serviços, a camada de SaaS. Estes serviços estão disponíveis para o usuário final através de uma interface *web*, conforme ilustrado na Figura 5.

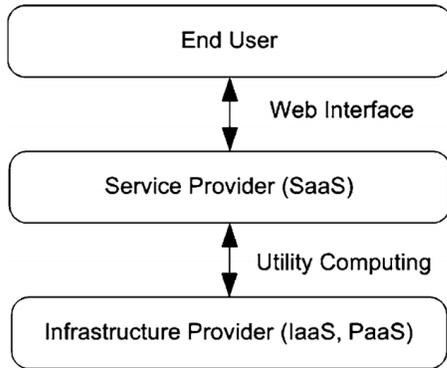


Figura 5: Modelos de Negócio da Cloud. Fonte: [45]

2.2.3 Banco de Dados

O conceito de Banco de Dados surgiu na IBM, entre as décadas de 1960 e 1970 enquanto a empresa estudava funções desempenhadas em escritórios e possíveis formas de automatizá-las. Ao longo deste estudo, foi detectado que o processo de armazenamento, indexação e organização de informações era muito custoso, pois era necessário empregar um grande número de pessoas para esta tarefa. Desta forma, a empresa se propôs a investir em pesquisas que tornassem estas atividades mais baratas através da mecanização das mesmas[51].

Diversas pesquisas foram realizadas, contudo, a mais bem sucedida delas surgiu em 1970 quando um pesquisador da IBM chamado Ted Codd publicou o primeiro artigo sobre Bancos de Dados Relacionais. Este trabalho tratava sobre o uso de cálculo e álgebra relacional para permitir o controle de uma grande quantidade de informações[51].

Após a publicação deste trabalho, Codd, apoiado pela IBM, criou um grupo de pesquisa conhecido como *System R* (Sistema R), que tinha como objetivo criar um sistema de banco de dados que eventualmente iria se tornar um produto. A medida que o tempo foi passando e o projeto foi ganhando maturidade, o Sistema R se tornou um cliente de banco de dados conhecido como DB2[51]. Para utilizar o sistema, o grupo desenvolveu uma linguagem bastante similar ao inglês, a *Structured Query Language* (SQL), que acabou se tornando um padrão nas indústrias de bancos de dados relacionais. O SQL é atualmente padronizado pela *International Organization for Standardization* (ISO)[51] através da ISO/IEC 9075[16].

Até os bancos de dados serem amplamente utilizados, as empresas costumavam armazenar seus dados em arquivos, físicos ou digitais. Esta abordagem era muito difícil de manter, pois havia uma alta propensão a falhas, inconsistências e alta redundância ocasionadas pelo uso destes arquivos[46]. Estes problemas foram solucionados quando houve a implementação dos bancos de dados devido a quatro características acronimizadas como ACID (Atomicidade, Consistência, Integridade e Durabilidade)[46], descritas a seguir.

A primeira das propriedades ACID é a Atomicidade, que consiste da propriedade que torna as transações de bancos de dados atômicas. Desta forma, quaisquer alterações deverão acontecer em sua totalidade para surtir efeitos. Em caso de erros, a base de dados deve ser recuperada para o seu estado anterior[46].

A segunda propriedade é a Consistência, que consiste da necessidade das transações manterem a consistência do banco de dados, levando-o sempre de um estado consistente para outro[46].

A terceira propriedade ACID é o Isolamento, que determina que uma transação deve ser executada isolada das demais. Ou seja, ela não pode sofrer interferência de quaisquer outras transações concorrentes[46].

A quarta propriedade é a Durabilidade, que assegura que as mudanças aplicadas no banco de dados por uma transação devem ser efetivadas e persistidas no mesmo. Em casos de falhas, as alterações não devem ser perdidas[46].

As bases de dados são gerenciadas por um Sistema de Gerenciamento de Banco de Dados (SGBD), que é um mecanismo que abstrai grande parte das especificidades da gestão de dados[51]. Existem diversas vantagens resultantes da utilização de um SGBD, dentre as quais se destacam: controle de redundância, restrição de acesso não autorizado, processamento eficiente de consultas, garantia de backup e restauração, fornecimento de múltiplas interfaces para utilização e as restrições de integridade. Todas estas características serão melhor explicadas a seguir[46].

O controle de redundância determina que o SGBD deve ter a capacidade de controlar redundâncias indesejadas que possam gerar inconsistências nos dados[46].

A restrição de acesso não autorizado é útil quando se considera que vários usuários pode utilizar um mesmo banco de dados. Nestes casos, é provável que boa parte destes não deva acessar e/ou alterar os dados disponíveis no banco de dados. Desta forma, O SGBD tem por função garantir a segurança através de um subsistema de autorização no qual seja possível estabelecer restrições para os usuários[46].

Para realizar o processamento eficiente de consultas, os SGBDs normalmente contam com um módulo para realizar o armazenamento temporário dos resultados de consultas, que ficam na memória principal durante um período de tempo para serem utilizadas no caso de haver outra consulta similar[46].

Para haver garantia de backup e restauração, o SGBD deve prover facilidades para restauração de falhas de *hardware* ou *software*. Em casos de falhas durante uma transação, o SGBD deve restaurar a base para o seu estado anterior[46].

Os SGBDs também fornecem múltiplas interfaces para os usuários utilizarem. Estas interfaces/clientes podem ser *Command Line Interfaces* (CLIs), *Graphical User Interfaces* (GUIs) ou Web GUIs[46].

Por padrão, o banco de dados tem certas restrições de integridade que devem complementar os dados na forma com que o banco foi estruturado durante sua modelagem/criação[46].

A modelagem do banco de dados conta com alguns entes como as Entidades e os Relacionamentos, por exemplo. Uma Entidade representa algum ser ou objeto do mundo real, seja ele concreto ou abstrato. Cada entidade pode possuir Atributos que determinam as características genéricas da mesma. Os relacionamentos, por sua vez, definem a forma com que as Entidades se relacionam entre si. Um relacio-

namento pode acontecer de três formas: Um para Um, Um para Muitos e Muitos para Muitos[46].

A modelagem de um banco de dados pode acontecer de duas formas: Lógica, quando são definidas as entidades e os relacionamentos entre as mesmas; e Física, também chamada de Diagrama Entidade-Relacionamento (DER), que, além de definir entidades e seus relacionamentos, também cuida de questões como tipos de dados dos atributos das entidades, cardinalidade de relacionamento e controle da relação de entidades através de chaves para identificação [46].

Utilizando o exemplo dos alunos que participam de um curso, o Diagrama Lógico apresentado na Figura 6 considera duas Entidades. A primeira, Estudante, conta com dois atributos: Nome e E-mail. Já a segunda entidade, Curso, conta com dois atributos: Nome e Carga Horária. Um estudante deve participar de no mínimo um e também de no máximo um curso para existir. Já o curso, pode ter a participação de nenhum ou vários estudantes.

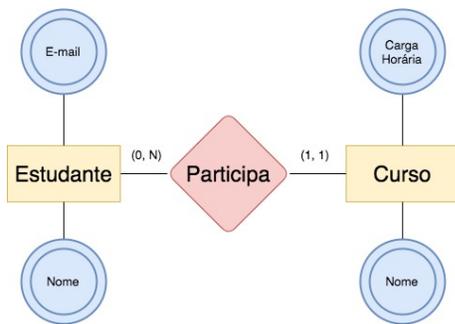


Figura 6: Exemplo de Diagrama Lógico. Fonte: Autor.

O Diagrama Entidade Relacionamento (DER) apresentado na Figura 7, apresenta o exemplo de estudantes que participam de um curso de forma mais formalizada. Agora, os atributos precisam ser tipados (como foi o caso de “INT”, que representa números inteiros e “VARCHAR”, que representa uma sequência de caracteres, um texto). Além disso, surgiram duas novas palavras, “Primary Key” e “Foreign Key”, que acompanham os novos atributos “id” e “curso_id”. A Primary Key representa um campo de chave primária, responsável por especificar um elemento no banco de dados. A Foreign Key indica um registro referenciado, que, no caso do exemplo, é o curso do qual o estudante participa.



Figura 7: Exemplo de Diagrama Entidade Relacionamento. Fonte: Autor.

Os dados de um banco são armazenados em tabelas que representam as entidades. Cada tabela pode ter linhas, também chamadas de registros ou tuplas, que indicam uma instância específica daquela entidade. As tabelas podem ser referenciadas através de chaves primárias que, por sua vez, podem ser estrangeiras em outros registros que referenciem a tupla que contém a chave. As chaves estrangeiras podem

apontar para linhas de uma mesma tabela ou de tabelas diferentes[46], conforme ilustra a Figura 8.

Tabela Estudante			
id	nome	email	curso_id
1	Maria	maria@email.com	1
2	Pedro	pedro@email.com	1
3	João	joao@email.com	2

Tabela Curso		
ID	nome	carga_horaria
1	Banco de Dados	90
2	Topsis	30

Figura 8: Exemplo de Tabelas do Banco de Dados. Fonte: Autor.

3 Trabalhos Correlatos

O artigo “Multi-criteria Group Decision Making Using a Modified TOPSIS Procedure”, propõe uma modificação na técnica, com o objetivo de levar em consideração as seleções em que haja múltiplos tomadores de decisão. É discutida a aplicação do método TOPSIS, em conjunto com a Lógica Fuzzy, para facilitar a ponderação das opções em relação aos critérios, a partir da especificação de valores linguísticos como “Alto”, “Médio” e “Baixo” ao invés de valores numéricos[49]. É um trabalho que se propõe a discutir as vantagens da utilização Fuzzy-TOPSIS, que é uma técnica diretamente aplicada neste trabalho.

O artigo “Multi Criteria Group Decision Making Approach for Smart Phone Selection Using Intuitionistic Fuzzy TOPSIS” discute a aplicação da Fuzzy-TOPSIS no contexto de seleção de smartphones a fim de reduzir as incertezas que incorrem ao indivíduo que está ponderando sobre a compra de um novo dispositivo[3]. Este trabalho apresenta uma utilização bastante diferente da mesma técnica.

O artigo “A-TOPSIS - An Approach Based on TOPSIS for Ranking Evolutionary Algorithms” propõe um método para solucionar ranqueamento e comparação de algoritmos utilizando TOPSIS. O trabalho se baseia no fenômeno de que algoritmos aplicados à Computação Evolucionária são reexecutados diversas vezes a fim de calcular desvios de padrão e outros fenômenos que possam vir a ocorrer. A partir disso, propõe-se uma variação da técnica na qual as alternativas são algoritmos a serem avaliados e os critérios os pontos de referência que os algoritmos consideram. Esta técnica foi nomeada A-TOPSIS [30].

A tese “Modelagem e Simulação de Apoio à tomada de Decisão na Reativação de Campos Maduros Terrestres e Economicamente Marginais de Petróleo e Gás” utiliza a TOPSIS e outras técnicas como Árvores de Decisão, Opções Reais e Lógica Fuzzy para propor um modelo híbrido que auxilie na tomada de decisão na reativação de campos maduros de petróleo e gás. São observadas as incertezas acerca da produção e questões econômico-financeiras relativas ao tema[58].

O artigo “A multi-criteria decision analysis approach for importance identification and ranking of network components” utiliza a TOPSIS para identificar a importância dos componentes de uma rede a partir da sua vulnerabilidade, criando um ranking de acordo com a sua importância dentro da infraestrutura de redes[1].

O artigo “Decision Support System Best Employee Assessments with Technique for Order of Preference by Similarity to Ideal Solution” realiza uma avaliação na qual verifica, dentro de uma avaliação de 10 funcionários, os melhores. Foram

considerados aspectos de performance de trabalho, atitude e personalidade, trabalho em equipe e capacidade intelectual[29].

O artigo “Análise de decisão multicritério para alocação de recursos e avaliação de tecnologias em saúde: tão longe e tão perto?” apresentou os aspectos metodológicos das MCDAs como definições, abordagens, aplicações e etapas de implementação. Para tal, foi realizado um estudo de caso no Instituto de Câncer do Estado de São Paulo (ICESP) com a intenção de compreender as perspectivas dos tomadores de decisão na elaboração de uma recomendação que visasse incorporar a tecnologia no Sistema Único de Saúde (SUS)[4].

O artigo “Análise da produção científica brasileira sobre seleção de fornecedores apoiada em métodos multicritério” busca verificar o que foi produzido cientificamente até então sobre a seleção de fornecedores por métodos MCDA. Esta pesquisa, de natureza exploratória, analisou artigos publicados entre os períodos de 2006 e 2015 nas bases de dados Scielo e Google Acadêmico. Foi detectada uma tendência de evolução nas publicações a respeito do tema, sobretudo no que tange ao estudo de caso como estratégia de pesquisa[31].

O artigo “Perspectivas de Gestão de Risco: *Disclosure* de informações sob a Lógica *Fuzzy*” visou apresentar uma metodologia baseada na utilização da lógica *Fuzzy* que modelasse e quantificasse as incertezas e subjetividades decorrentes da divulgação de informações de risco divulgadas nas demonstrações contábeis. Para tal, utilizou-se a TOPSIS como uma ferramenta para gerar um *ranking* das divulgações de risco entre as 47 empresas analisadas para este trabalho[9].

O artigo “Estruturação do processo decisório para reforma de edificações escolares públicas do ensino fundamental utilizando o método multicritério de apoio à decisão construtivista (MCDA-C)” visou, por meio de um estudo de caso exploratório, de natureza aplicada e abordagem quantitativa e qualitativa, desenvolver um modelo MCDA que auxilie na priorização de intervenções de reforma em edificações escolares públicas do ensino fundamental levando em conta os aspectos legais para tal tipo de edificação[5].

A monografia “TOPSIS versus AHP: Métodos Multicritério aplicados em um estudo de caso” apresenta como técnicas MCDA, em especial a TOPSIS e a AHP, que podem ser utilizadas no cotidiano para auxiliar na tomada de uma decisão. Para implementar as técnicas, foi utilizado um estudo de caso real chamado “Projeto preparatório para o gerenciamento dos recursos hídricos de Paraíba do Sul”. Ao longo do trabalho foram identificados os prós e contras das utilizações das técnicas na análise e ponderação de critérios, bem como na classificação e seleção dos projetos de investimento[32].

O artigo “*Strategic Planning Methodology for Information Security*” propõe o desenvolvimento de uma metodologia de planejamento de segurança da informação que é adaptada da PETIC e agrega requisitos das normas NBR ISO/IEC 27001:2013, NBR ISO/IEC 27002:2013 e NBR ISO/IEC 27005:2011. Esta metodologia aborda aspectos da PETIC *Decision Maker* (PDM) que tem como objetivo analisar a tomada de decisão em um planejamento estratégico de TIC através da TOPSIS para priorização de projetos relacionados a este âmbito. Esta adaptação é focada em minimizar custos e maximizar o ROI (*Return over Investment*)[33].

A partir destes trabalhos é possível constatar que a flexibilidade da TOPSIS viabiliza sua utilização em diversas áreas, com várias possibilidades de aplicações.

4 Metodologia

Após a realização do levantamento bibliográfico de trabalhos relevantes que utilizam os termos “TOPSIS” e “Suporte a Decisão” e de coletar a base de dados sobre o PIBIC que foi cedida pela PRPGI foi possível verificar a viabilidade da aplicação da técnica TOPSIS para responder as perguntas da pesquisa.

A aplicação do método TOPSIS utiliza uma matriz (D), representada pela Equação 1, que é composta por uma lista (a) relativa às opções viáveis. Cada item listado está agrupado sobre um conjunto de critérios (c). Estes itens atendem os critérios de diferentes maneiras, de forma que seu valor (x) representa o desempenho desta opção sobre aquele determinado critério[41].

$$D = [x_{ij}]_{axc} \quad (1)$$

Existe um vetor de pesos (Equação 2) predefinido pelo tomador de decisão, de forma que W_1 equivale a C_1 . Um fator importante é que o somatório dos valores propostos no vetor de pesos deverá equivaler a 1 (ou 100%, a depender da interpretação), vide Equação 3.

$$W = (w_1, w_2, \dots, w_n) \quad (2)$$

$$\sum_{j=1}^n w_j = 1 \quad (3)$$

Após a inserção destes valores, é necessário normalizar a Matriz de Opções, conforme a Equação 1. Este processo é necessário devido à utilização de escalas verbais, de forma que seus valores verbais precisam ser convertidos para valores numéricos. A matriz (R), ilustrada na Equação 4, conta com linhas (m) e colunas (n), que são respectivamente representadas pelas opções de decisão e os critérios estabelecidos no início da seleção[58].

$$R = [r_{ij}]_{m \times n} \quad (4)$$

Cada item desta Matriz Normalizada é dado pela razão entre o valor original do item sobre o somatório dos quadrados de cada opção do critério (Equação 5)[41].

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}} \quad (5)$$

Uma vez concluído o processo de normalização, os valores da matriz R (Equação 4) precisarão ser ponderados de acordo com o vetor de pesos (Equação 2), que relativizam a importância de cada critério. Após a ponderação, é gerada uma matriz (P), que é oriunda deste da operação representada na Equação 6 e está representada na Equação 7[41].

$$p_{ij} = w_j \cdot r_{ij} \quad (6)$$

$$P = [p_{ij}]_{m \times n} \quad (7)$$

Uma vez criada a matriz P (Equação 7), inicia-se o processo de cálculo do TOPSIS. A primeira etapa consiste em identificar quais são os melhores e piores casos possíveis dentro do escopo de opções informado. Estes valores são conhecidos como: solução ideal positiva (Equação 8), para os critérios de benefício; e solução ideal negativa (Equação 9) para critérios de custo.

$$A^+ = (p_1^+, p_2^+, \dots, p_n^+) \quad (8)$$

$$A^- = (p_1^-, p_2^-, \dots, p_n^-) \quad (9)$$

$$p_j^+ = \begin{cases} \max_i(p_{ij}), & \text{se o critério é de benefício} \\ \min_i(p_{ij}), & \text{se o critério é de custo} \end{cases} \quad (10)$$

$$p_j^- = \begin{cases} \min_i(p_{ij}), & \text{se o critério é de benefício} \\ \max_i(p_{ij}), & \text{se o critério é de custo} \end{cases}$$

Outra alternativa para o cálculo das soluções ideais é utilizar valores binários, de forma que os benefícios equivalham a 0 e os custos a 1. Independente da alternativa escolhida, é mandatório que haja o conceito de custo e benefício para o TOPSIS funcionar corretamente[41].

Após aferir as soluções ideais, é necessário calcular a Distância Eclideana dos valores de cada critério das opções propostas. Convém ressaltar que algumas literaturas poderão representar este passo como distâncias relativas ou distâncias até a solução ideal. Com esta operação, é retornado o *ranking* das alternativas que melhor se aplicam à conjuntura proposta[41].

Após aferir as soluções ideais, faz-se necessário calcular as distâncias relativas às soluções ideais positiva (Equação 11) e negativa (Equação 12) para cada opção proposta ao projeto. Convém ressaltar que, em algumas literaturas, a nomenclatura desta etapa pode ser representada como “Distâncias Relativas” ou “Distâncias até a Solução Ideal”.

$$d_i^+ = \sqrt{\sum_{j=1}^n (d_{ij}^+)^2} \quad (11)$$

$$d_i^- = \sqrt{\sum_{j=1}^n (d_{ij}^-)^2} \quad (12)$$

$$d_i^+ = p_j^+ - p_{ij} \quad (13)$$

$$d_i^- = p_j^- - p_{ij} \quad (14)$$

Por fim, utilizando-se das distâncias calculadas, a TOPSIS irá calcular a proximidade relativa (Equação 15) de cada elemento à solução ideal e, a partir disso, irá realizar o ranqueamento das opções de forma a priorizar as que melhor se aplicam ao escopo apresentado.

$$\xi_i = \frac{d_i^-}{d_i^+ + d_i^-} \quad (15)$$

Em resumo, o TOPSIS é uma técnica que realiza o ranqueamento das opções de um projeto a fim de obter as alternativas que melhor se apliquem ao contexto apresentado[58]. Para tal, é necessário que estas opções atendam aos critérios preestabelecidos pelo mantenedor da seleção. Nesta perspectiva, foi verificado que o TOPSIS pode ser utilizado em seleções como o PIBIC. Desta forma, propôs-se a aplicação da técnica no contexto de seleção de bolsistas de forma que estes processos possam se tornar mais rápidos e flexíveis.

Convém ressaltar que a TOPSIS permite considerar tanto os aspectos objetivos quanto os subjetivos. Contudo, eventualmente podem haver imprecisões derivadas de uma má ponderação dos critérios ou na avaliação dos projetos. Desta forma, foram utilizadas escalas verbais ao invés de valores numéricos para atenuar a possibilidade de erros durante a avaliação dos projetos. Em outras palavras, ao invés de determinar como um projeto atende a um determinado critério a partir de um valor numérico, como uma nota, por exemplo, o usuário irá qualificá-lo em níveis como “insuficiente”, “na média”, “acima da média”, “excelente”. Neste caso, o sistema se encarrega de interpretar e converter estes valores durante a aplicação da TOPSIS[49].

5 Software Topsisfy

5.1 Arquitetura do Topsisfy

O Topsisfy é um *software* estruturado em três camadas, o *client-layer*, *Restfull API* e *Database*, que segue o estilo arquitetural *Client-server*, onde os clientes de uma aplicação solicitam informações e/ou processamento de um servidor que pode, ou não, consumir dados de um banco de dados conforme ilustra a Figura 9.

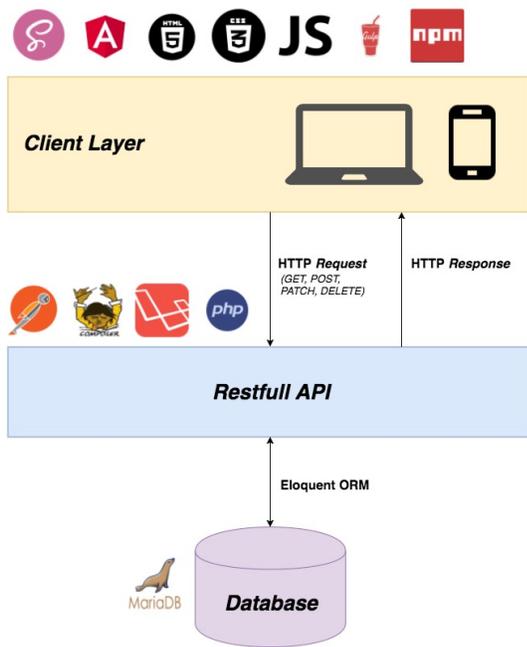


Figura 9: Visão estrutural do software Topsisfy. Fonte: Autor.

A primeira camada, *Client Layer*, contém todo *front-end* necessário para que os usuários possam interagir com a aplicação. Esta camada foi desenvolvida utilizando HTML, CSS, JavaScript, AngularJS, Gulp, Sass, NPM e o *template* de projeto *Weskit*, que serão explicados na Subseção 5.2. Algumas destas tecnologias, no caso o NPM, Gulp e Sass, foram utilizados somente durante o período de desenvolvimento.

A segunda camada, *Restfull API*, que fica do lado do servidor, contém as regras de negócio do sistema. É uma API que provê serviços para Login de usuários, recuperação de senha, cadastro, edição e deleção de registros de projetos, processos seletivos e afins, além, é claro, do algoritmo da TOPSIS. Nesta camada foram utilizadas as tecnologias PHP, Laravel, Eloquent ORM, Postman e Composer. Assim como o *front-end*, houveram tecnologias que só são utilizadas para o âmbito de desenvolvimento, como o Composer e o Postman. Estas tecnologias serão melhor discutidas na Subseção 5.3.

A terceira camada, *Database*, é onde está estruturado o banco de dados conforme especificado no Diagrama Entidade e Relacionamento disponível no Apêndice 10.2 e onde ficarão armazenadas os dados da aplicação. O sistema de gerenciamento de banco de dados utilizado foi o MariaDB, que será melhor apresentado na Subseção 5.4.

A comunicação entre o *Client Layer* e a *Restfull API* acontece através de *HTTP Request* com os verbos *GET*, *POST*, *PATCH* e *DELETE*. Estas *requests* recebem uma *HTTP Response* para ser tratada no *front-end*.

Já a *Restfull API* e o *Database* se comunicam através do *framework* Eloquent ORM, que abstrai toda parte de comunicação com banco de dados.

5.2 Front-end do Topsisfy

O desenvolvimento *front-end*, em tradução livre, significa "Desenvolvimento da parte dianteira". Neste sentido, refere-se a criação de interfaces que propiciam a interação do usuá-

rio com o *software*. É a construção da *Graphical User Interface*.

Para a criação do sistema Topsisfy, foram utilizadas as seguintes tecnologias de front-end:

Hyper-text Markup Language (HTML) é uma linguagem de marcação inicialmente projetada para descrever documentos científicos. Atualmente, é uma das tecnologias-chave da Internet, por se tratar de uma linguagem que estrutura a apresentação de conteúdo na *World Wide Web* (WWW) [10]. O HTML foi utilizado no sistema para determinar a estrutura das páginas.

Cascading Style Sheet (CSS) é uma linguagem de folhas de estilo que formata e estiliza a informação estruturada através do HTML[10]. É importante porque, além de propiciar uma melhor experiência de utilização ao usuário, separa o modo de apresentação do conteúdo do documento[35]. O CSS foi utilizado para estilizar as páginas estruturadas via HTML.

JavaScript é uma linguagem de programação dinâmica e de alto nível que é adequada para o paradigma de computação Funcional e também contém aspectos da Orientação a Objetos[36]. Está presente em diversos navegadores e propicia melhorias na comunicação entre o usuário e aplicações *web*[36]. A JavaScript é atualmente mantida e evoluída pela Ecma International[52] através do *Technical Comittee 39*[28], responsável pela especificação ECMA262[52], que define a linguagem.

AngularJS é um *framework* criado em 2009 por Misko Hevery, um engenheiro da Google[27]. Esta ferramenta possui funcionalidades interessantes para o desenvolvimento, como o *two-way data binding*, que é a atualização da *view* e do *model* em tempo real, a injeção de dependências, códigos facilmente testáveis, entre outros[10]. Atualmente, o Angular é mantido através de um repositório público no Github[53]. Foi utilizado no sistema para interagir com o *back-end*, fazer o roteamento de páginas e a interação com o usuário.

Syntactically Awesome Style Sheets (Sass) é uma *gem*[48], uma biblioteca da linguagem de programação Ruby, que foi utilizada para fazer o pré-processamento CSS. O Sass permite a utilização de diversos recursos como a modularização do CSS, criação de estruturas de controle (Laços de repetição e decisão), funções e diversas outras funcionalidades[23]. O Sass foi utilizado durante o desenvolvimento do *software* para gerar o CSS que iria para o ambiente de produção. Sua utilização se deu devido a características como constantes, inserções de arquivos, funções e outras mais, que tornam o desenvolvimento do CSS mais simples e reutilizável[23].

Gulp é uma ferramenta de *build* automatizado para JavaScript que conta com diversos plugins para realizar as mais diversas funcionalidades, desde minificação de código-fonte até a concatenação, configuração de um servidor local e várias outras possibilidades[22]. O gulp foi utilizado no sistema para fazer tarefas como compilação do Sass, minificação do HTML, CSS e JavaScript desenvolvido. Foi uma tecnologia utilizada somente durante o desenvolvimento.

Node Package Manager (NPM) é uma ferramenta para gerenciamento de pacotes e dependências para a linguagem JavaScript[37]. Foi utilizado durante a etapa de desenvolvimento para ter um controle maior das bibliotecas e plugins utilizados.

Web Starter Kit (Weskit) é um *template* de projeto que já traz diversas configurações para o GulpJS e já define uma estrutura apropriada para páginas Web[11].

5.3 Back-end do Topsisfy

O desenvolvimento do *Back-end* do sistema Topsisfy consistiu do desenvolvimento das regras de negócio da aplicação, a implementação do algoritmo da TOPSIS e a intermediação das informações persistidas no banco de dados para o *Front-end*.

Dentre as tecnologias utilizadas, estavam: PHP, Laravel, Eloquent, Postman e Composer. Todas elas sendo melhor explicadas a seguir.

O *PHP Hypertext Preprocessor* (PHP) é uma linguagem de programação *open-source* focada para o desenvolvimento web[55]. Esta linguagem foi utilizada no sistema devido sua flexibilidade para aplicações *web*.

O Laravel é um framework para a linguagem de programação PHP que permite a criação de aplicações web completas. Pode ser utilizada para criar tanto o *front-end*, através da sua *template-engine* Blade, quanto o *back-end* através da construção de APIs Restfull[40]. No caso do sistema, o *framework* foi utilizado apenas para a criação da API *Restfull*. Esta decisão foi tomada para que o *front-end* e o *back-end* ficassem desacoplados, de forma que pudessem evoluir paralelamente e que não houvesse grande impacto no caso de uma mudança de tecnologia no futuro. A justificativa para usar o Laravel no sistema é a sua simplicidade e praticidade para o desenvolvimento e as diversas facilidades para a criação de APIs, dentre as quais encontra-se a próxima tecnologia, o Eloquent ORM.

O Eloquent é um *framework* para *Object Relational Mapping* (ORM) embutido no *framework* Laravel[39]. Este *framework* abstrai todas especificidades relacionadas a comunicação com o banco de dados, tornando o processo de leitura e escrita muito mais simples. Sua utilização no sistema se deu pelos seguintes motivos: 1) é uma tecnologia que já vem incluída no *framework* Laravel; 2) Traz todas as facilidades de um ORM para a comunicação com o banco de dados; 3) em caso de mudança da base de dados, não haveria grande impacto na aplicação, sendo necessário apenas ajustar as informações do banco de dados.

O *Postman* é uma solução para auxiliar no desenvolvimento de APIs consistentes. É possível utilizar esta ferramenta para fazer testes de funcionamento, performance e diversas outras possibilidades apresentadas na documentação do projeto[56]. O *Postman* foi utilizado para testar a aplicação durante a etapa de desenvolvimento sem a necessidade de estar utilizando o *front-end* para tal.

O *Composer* é um gerenciador de pacotes e dependências para a linguagem PHP[54]. O *Composer* foi utilizado para ter maior controle sobre as dependências utilizadas na parte do *back-end*, como o *Laravel*, por exemplo.

Além da etapa de desenvolvimento, também houveram algumas questões relacionadas a infraestrutura. Para disponibilizar a aplicação para testes durante seu período de desenvolvimento, foi utilizado um *droplet* (máquina virtual) da Digital Ocean[38]. Esta máquina estava configurada com o Sistema Operacional Ubuntu (versão 16.04.2 x64), contando com 512mb de memória RAM e 20gb de armazenamento SSD.

5.4 Banco de Dados do Topsisfy

Para o desenvolvimento do sistema Topsisfy foi utilizado um banco de dados chamado MariaDB.

O MariaDB é um banco de dados desenvolvido em código aberto que foi criado pelos desenvolvedores originais do

MySQL após a aquisição do mesmo pela empresa Oracle. Esta iniciativa se deu devido a série de incertezas relativas ao futuro do MySQL[25].

Por se tratar de um *software* livre e *open source*, o MariaDB tem um processo de desenvolvimento mais aberto e vibrante, permitindo a qualquer desenvolvedor participar das discussões sobre a evolução da ferramenta[17].

Além disso, por ser mais constantemente atualizado do que o MySQL, o MariaDB tem uma segurança mais rápida e transparente do que o seu concorrente. Existe uma lista de *Common Vulnerabilities and Exposures* (CVE)[7] aberta do MariaDB para que os colaboradores possam contribuir aplicando correções no banco de dados[25].

Devido ao fato de estar em constante evolução, o MariaDB possui recursos mais avançados e melhor desempenho que o MySQL, permitindo inclusive a utilização de bases *Not Only SQL* (NoSQL)[25].

Além disso, por se tratar de um banco baseado no *MySQL*, é bastante fácil migrar para o MariaDB. Até sua versão 5.5, o banco de dados apresentou compatibilidade total com o concorrente (também na versão 5.5). Contudo, com o passar do tempo, houve quebra da compatibilidade binária, o que pode gerar algumas dificuldades específicas durante a migração[25].

Em resumo, o MariaDB foi utilizado por ser um banco de dados que está em constante evolução, tem boa performance e também por se tratar de um *software* livre, além de ser utilizado por grandes empresas como a Google, Wikipédia e o Wordpress[17].

6 Caso de Uso

Para validar o sistema Topsisfy foram utilizados os resultados do Programa Institucional de Bolsas de Iniciação Científica (PIBIC) do IFBA para as seleções de 2013-2014[44], 2014-2015[43] e 2015-2016[42]. Além disso, foram consultados 17 docentes do Instituto, sendo 16 desses professores do *campus* Salvador e 1 do *campus* Camaçari, que foi convidado por sua tese de doutorado ser uma das referências mais importantes que auxiliaram no desenvolvimento deste trabalho.

O primeiro passo consistiu em avaliar os editais dos projetos a fim de extrair informações importantes para o sistema, como potenciais critérios e/ou pesos de critérios, por exemplo. Na Tabela 1, abaixo representada, são apresentados os critérios utilizados pela PRPGI durante os processos de seleção do PIBIC.

Tabela 1: Critérios do PIBIC-IFBA. Fonte: Autor.

-	Critério de Avaliação	Nota Máxima
1	Perfil do orientador (Titulação)	08 pontos
2	Produção científico-tecnológica (2010-2016)	20 pontos
3	Experiências de orientações concluídas ou coorientações	12 pontos
4	Experiência de orientações em andamento ou coorientações	05 pontos
5	Perfil do Bolsista	10 pontos
6	Avaliação do projeto do pesquisador e Plano de Trabalho do estudante	45 pontos
Pontuação Total dos Critérios		100 pontos

O primeiro critério, relacionado ao perfil do orientador, leva em conta a sua titulação, considerando apenas as pós-graduações *stricto sensu*. No caso, mestrado e doutorado.

No segundo critério, é considerada a produção científica do orientador nos últimos anos. São contabilizados: autoria, ou organização de livros, além da escrita de capítulos; trabalhos completos ou resumos publicados em congressos; artigos publicados em periódicos nacionais ou internacionais, patentes ou softwares, entre outros.

Para o terceiro e quarto critérios, são contabilizadas as participações como orientador ou coorientador em trabalhos de conclusão de cursos de pós-graduação, monografias de graduação, iniciação científica (para o nível superior ou médio) e orientações de outras naturezas (estágios, monitorias, entre outros).

O quinto critério pondera sobre o perfil do bolsista a partir do seu coeficiente de rendimento (CR) e se ele foi voluntário em alguma edição anterior do Programa Institucional e Voluntário de Iniciação Científica ou Tecnológica (PIVIC / PIVIT).

E, por fim, é considerada a avaliação do projeto do pesquisador, bem como o plano de trabalho do estudante. Este sexto e último critério da PRPGI leva em conta a clareza dos objetivos propostos na pesquisa, a relevância do tema central, a viabilidade de execução do projeto, a adequação do plano de trabalho o projeto de pesquisa, adequação da metodologia e cronograma do plano de trabalho.

Conforme ilustrado na Tabela 2, foi possível avaliar esses critérios de forma a adequar estes pontos dos critérios da PRPGI para que eles fossem compatíveis com o sistema desenvolvido.

Tabela 2: Critérios do PIBIC-IFBA. Fonte: Autor.

Descrição	Critério	Peso
Critério 1	Perfil do orientador (Titulação)	08%
Critério 2	Produção científico-tecnológica (2010-2016)	20%
Critério 3	Experiências de orientações concluídas ou coorientações	12%
Critério 4	Experiência de orientações em andamento ou coorientações	05%
Critério 5	Perfil do Bolsista	10%
Critério 6	Avaliação do projeto do pesquisador e plano de trabalho do estudante	45%
Porcentagem Total		100%

Os critérios que foram aplicados no sistema são idênticos aos que foram originalmente propostos pela PRPGI. Contudo, conforme é apresentado na Tabela 3, foram necessárias duas adaptações:

1. Foi utilizada a grandeza de porcentagem ao invés de pontos;
2. Com objetivo de totalizar 100%, o Critério relativo ao Perfil do Bolsista foi arredondado para 10.

Tabela 3: Comparativo entre os critérios da PRPGI e os que foram utilizados no Topsisfy. Fonte: Autor.

Descrição	Nota Máxima	Peso
Critério 1 (C1)	08 pontos	08%
Critério 2 (C2)	20 pontos	20%
Critério 3 (C3)	12 pontos	12%
Critério 4 (C4)	05 pontos	05%
Critério 5 (C5)	10 pontos	10%
Critério 6 (C6)	45 pontos	45%
Totais	100 pontos	100%

Com o objetivo de facilitar a compreensão e utilização do sistema, foram criadas escalas que serviram para avaliar como cada pleiteante do processo seletivo atendeu a cada critério. Estas escalas são bastante similares à Escala Likert.

As escalas do primeiro critério, que é relativo à titulação do orientador, foram definidas como: Nenhuma das Anteriores, Mestrado e Doutorado, ilustradas Tabela 4.

Tabela 4: Escalas do Critério 1. Fonte: Autor.

Escala	Valor
Doutorado	100%
Mestrado	50%
Nenhuma das Anteriores	0%

Já para os demais critérios foram definidas seis escalas, que representam intervalos de 0 a 100%. Suas descrições ficaram como: Insuficiente, Baixo, Abaixo da Média, Média, Acima da Média e Alto, apresentadas na Tabela 5

Tabela 5: Escalas do Critério 2, 3, 4, 5 e 6. Fonte: Autor.

Escala	Valor
Alto	100%
Acima da Média	80%
Média	60%
Abaixo da Média	40%
Baixo	20%
Insuficiente	0%

Após determinar as escalas, utilizamos um questionário com 17 professores do IFBA para verificar sua opinião sobre a importância de cada critério do processo seletivo do PIBIC. Os professores que realizaram a avaliação destes critérios são das áreas Linguagem, Física e Computação.

O questionário consistiu de duas partes. Na primeira, foi verificado o perfil acadêmico do avaliador (Licenciatura ou Bacharelado) e sua titulação máxima (Graduado, Especialista, Mestre ou Doutor). Já na segunda etapa, foram listados os critérios a serem avaliados.

É importante ressaltar que o primeiro critério do processo, relativo a titulação máxima do orientador, não foi diretamente pontuada pelo avaliador. Neste caso, foi considerada a titulação máxima do próprio avaliador para pontuar este aspecto. Assim, é possível considerar a relevância do orientador no processo seletivo. Desta forma, as notas atribuídas pelos avaliadores correspondeu à 92% da nota total, enquanto o primeiro critério valeu 8%, que é proporcional à nota que a PRPGI estabelece para o critério de Titulação.

A Tabela 6 representa a relação de avaliadores e a forma com que os mesmos atendem e ponderam os critérios estabelecidos pela PRPGI.

Tabela 6: Avaliações dos Professores. Fonte: Autor.

Avaliador	C1	C2	C3	C4	C5	C6
Avaliador 01	08	20	15	15	20	30
Avaliador 02	08	30	10	10	10	40
Avaliador 03	08	30	10	10	10	40
Avaliador 04	08	30	20	05	10	35
Avaliador 05	08	20	20	00	10	50
Avaliador 06	04	40	20	20	10	10
Avaliador 07	00	20	20	20	10	30
Avaliador 08	04	15	15	20	10	40
Avaliador 09	08	40	15	15	10	20
Avaliador 10	08	30	25	05	30	10
Avaliador 11	08	20	20	15	30	15
Avaliador 12	04	25	10	10	25	30
Avaliador 13	08	20	10	10	20	40
Avaliador 14	04	20	15	10	25	30
Avaliador 15	04	20	15	15	10	40
Avaliador 16	08	20	15	05	10	50
Avaliador 17	08	20	20	10	10	40

De posse das avaliações dos professores, a etapa seguinte consistiu em cadastrar as opções de cada processo seletivo para que estas informações obtidas pudessem ser inseridas no Topsisfy. Na Tabela 7, estão representadas as quantidades de inscritos para as três edições do processo seletivo do PIBIC avaliadas neste trabalho.

Tabela 7: Inscritos para o processo seletivo PIBIC-IFBA. Fonte: Autor.

Seleção	Qtd. de Inscritos
PIBIC 2013 - 2014	50
PIBIC 2014 - 2015	87
PIBIC 2015 - 2016	134

Uma vez que os projetos foram cadastrados, o avaliador pôde iniciar o processo de avaliação. A primeira etapa consistiu em seguir para a tela de avaliações e selecionar o projeto desejado.

Após selecionar o processo seletivo desejado, o sistema apresentou para o avaliador todos os critérios e opções do processo seletivo desejado. Nesta tela de cadastro, o avaliador teve a possibilidade de ponderar o quão importante ele julga que cada critério é, a partir da valoração do peso deste.

Após definir os pesos do critério, o avaliador pôde começar a determinar como cada opção atendeu a cada um dos critérios. Essa escolha aconteceu a partir de opções de múltipla escolha que apresentaram as escalas de cada critério

conforme ilustram as Tabelas 4 e 5.

Para considerar os resultados das avaliações que foram realizadas pela PRPGI, foi necessário realizar uma conversão das notas atribuídas para cada projeto. Tendo em vista que o sistema trabalha utilizando porcentagem, foi realizada uma regra de três para adaptar estes valores, conforme Equação 16. Nesta operação, a nota máxima equivaleu a 100%, sendo equiparada com a nota obtida para determinar o valor convertido como um percentual.

$$nota_convertida = \frac{nota_prpgi \times 100}{nota_maxima_do_critério} \quad (16)$$

Após realizar o processo de conversão, estes valores foram alocados a uma escala conforme representado nas tabelas 8 e 9, onde *nota* representa a pontuação do projeto no critério em questão após a conversão para percentual.

Tabela 8: Conversão do Critério 1 da PRPGI para Escala. Fonte: Autor.

Escala	Alocação
Doutorado	<i>nota</i> = 100%
Mestrado	<i>nota</i> = 50%
Nenhuma das Anteriores	<i>nota</i> = 0%

Tabela 9: Conversão dos Critérios 2, 3, 4, 5 e 6 da PRPGI para Escalas. Fonte: Autor.

Escala	Alocação
Alto	<i>nota</i> > 80% e <= 100%
Acima da Média	<i>nota</i> > 60% e <= 80%
Média	<i>nota</i> > 40% e <= 60%
Abaixo da Média	<i>nota</i> > 20% e <= 40%
Baixo	<i>nota</i> > 0% e <= 20%
Insuficiente	<i>nota</i> = 0%

Após concluir seu trabalho, o avaliador finaliza a avaliação. A partir daí, o algoritmo do Topsisfy irá executar a técnica de forma que gerará um *ranking* ordenado a partir dos projetos que melhor se aplicam e que, consequentemente, deveriam ser contemplados com a bolsa PIBIC.

Para ter acesso a este *ranking*, o usuário deverá se dirigir até a tela de visualização de resultados onde ele irá selecionar um dos projetos que já avaliou para poder consultar seu resultado.

Uma vez selecionado o projeto, é retornada uma lista contendo a relação de todas as opções do projeto e a sua proximidade relativa à solução ideal.

Nesta listagem, os projetos que tiverem uma proximidade relativa mais próxima de 1 são as melhores opções.

A proximidade relativa para esta tela foi considerada com até cinco casas decimais para obter maior precisão na ava-

liação, de forma que processos cujas avaliações sejam muito similares possam ser mais facilmente desempatados.

O fluxo de atividades para a avaliação de um projeto está ilustrado Figura 10 e consiste de:

1. O avaliador deve acessar a tela de avaliações através do menu principal;
2. Ao acessar a tela, é necessário selecionar um projeto para seguir;
3. O avaliador passa por cada um dos critérios atribuindo seus pesos;
4. Uma vez que todos os critérios estão ponderados, passa-se para etapa de avaliação, onde será necessário atender cada critério para cada uma das opções pleiteantes do projeto;
5. O usuário clica no botão “Salvar e Finalizar”, disparando, assim, a execução do TOPSIS.

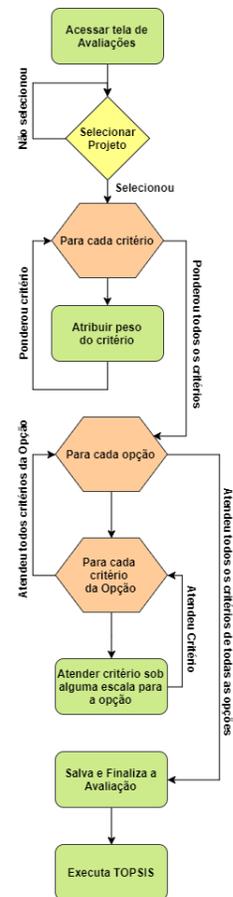


Figura 10: Representação do processo de avaliação. Fonte: Autor.

A etapa de avaliação dos projetos está integrada ao fluxo geral do negócio do sistema, que é composto pelos seguintes passos:

1. O processo seletivo é criado. Exemplo: Processo do PIBIC 2013-2014;

2. Em seguida, os projetos que irão concorrer aos critérios poderão ser definidos;
3. Concomitantemente ao cadastro de projetos, os critérios poderão ser definidos e cadastrados, bem como suas escalas serão propostas;
4. Após a etapa supra, é iniciada a fase de apreciação do resultado do processo. Inicialmente são ponderados os critérios de acordo com a determinação do avaliador;
5. Após ponderar os critérios, é necessário avaliá-los. Esta avaliação verifica a forma com que cada uma das opções atende aos critérios determinados no começo do projeto;
6. O avaliador poderá clicar no botão “Salvar e Finalizar”. Após salvar os dados do formulário é executado o algoritmo da TOPSIS, de forma que são calculados os resultados para o avaliador;
7. A partir do resultado do TOPSIS o conjunto de dados pode ser analisado a fim de determinar os projetos aprovados;

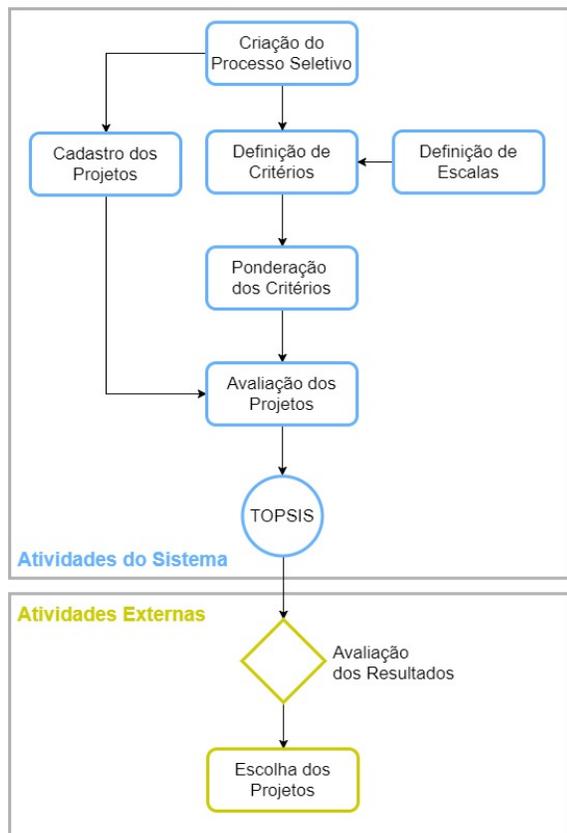


Figura 11: Representação do fluxo geral do sistema. Fonte: Autor.

7 Discussão e Resultados

Com objetivo de facilitar a visualização dos resultados, foram elaboradas duas visualizações. A primeira delas consiste em apresentar um comparativo entre os pesos estabelecidos

por cada avaliador para os critérios da PRPGI, além dos pesos que a mesma já havia estabelecido anteriormente. Já no segundo caso, a verificação de incidências de um projeto na avaliação de todos os decisores foi viabilizada, inclusive da Pró-reitoria de Pós-graduação e Inovação.

Tendo em vista que a PRPGI classificou os critérios com o mesmo peso para os três processos seletivos, o mesmo se aplicou às avaliações realizadas pelos professores. Desta forma, o gráfico, ilustrado no Apêndice 10.1, se configura da mesma maneira para os três estudos de caso.

Ao analisar a variação das avaliações dos docentes, apresentada na Tabela 10, é possível verificar que houve variação em todos os critérios, sendo mais acentuada no último, que é relativo ao Projeto do Pesquisador..

Tabela 10: Variação em Avaliações de Critérios. Fonte: Autor.

Critério	Mínimo	Máximo	Variação
Critério 01	0%	8%	8%
Critério 02	15%	40%	25%
Critério 03	10%	25%	15%
Critério 04	0%	20%	20%
Critério 05	10%	30%	20%
Critério 06	10%	50%	40%

A segunda visualização representa o número de incidências de cada projeto entre os cinco primeiros colocados das seleções de cada um dos avaliadores.

Como houveram três processos seletivos, foram geradas três visualizações de incidências de projetos.

Para o processo seletivo PIBIC 2013-2014, representado na Tabela 11, houveram 50 inscritos. Nesta seleção, foi bastante perceptível a convergência de boa parte dos resultados. Houve, inclusive, unanimidade na aprovação do projeto 14. Além disso, houveram também dois outros projetos, 12 e 32, que tiveram um desempenho excelente, aparecendo em dezessete avaliações. Além destes, os projetos 47 e 44 apareceram em apenas uma das avaliações.

Tabela 11: Legenda do Gráfico de Incidências entre os Primeiros Colocados do PIBIC 2013-2014. Fonte: Autor.

Projeto	Qtd. Incidências
Projeto 14	18
Projeto 12	17
Projeto 32	17
Projeto 22	13
Projeto 03	11
Projeto 23	7
Projeto 15	5
Projeto 47	1
Projeto 44	1

Ao longo da seleção do PIBIC 2014-2015, ilustrado na Tabela 12, houveram 87 pleiteantes. Destes, cinco resultados bastante convergentes. Contudo, diferente do ano anterior, nenhum projeto foi selecionado de forma unânime.

Tabela 12: Legenda do Gráfico de Incidências entre os Primeiros Colocados do PIBIC 2014-2015. Fonte: Autor.

Projeto	Qtd. Incidências
Projeto 29	17
Projeto 38	16
Projeto 69	16
Projeto 48	16
Projeto 65	16
Projeto 66	6
Projeto 15	2
Projeto 66	1
Projeto 63	1
Projeto 40	1

Durante o processo do PIBIC 2015-2016, apresentado na Tabela 13, foram levados em conta 134 projetos. Quando os resultados desta seleção são comparados com os dos anos anteriores pode-se perceber que a quantidade de projetos cresceu bastante, o que provavelmente implicou em um resultado menos concentrado em projetos específicos. Assim como o a edição de 2014-2015, não houve unanimidade na seleção de nenhum projeto.

Tabela 13: Legenda do Gráfico de Incidências entre os Primeiros Colocados do PIBIC 2015-2016. Fonte: Autor.

Projeto	Qtd. Incidências
Projeto 39	16
Projeto 37	16
Projeto 84	13
Projeto 56	12
Projeto 45	15
Projeto 20	5
Projeto 117	3
Projeto 67	1
Projeto 114	1
Projeto 1	1
Projeto 51	1
Projeto 43	1

A ideia desta visualização das incidências é viabilizar um panorama geral das avaliações, considerando todos aqueles que ponderaram os critérios, permitindo uma seleção muito mais concisa e colaborativa.

A sensibilidade do sistema Topsisfy permitiu evidenciar as especificidades existentes nas crenças dos avaliadores. Nas Tabelas 11, 12 e 13 é possível constatar que, apesar de haver certa congruência em boa parte dos projetos selecionados, houve, em todos os processos seletivos, diversos projetos que apareceram em apenas uma avaliação. Esta característica é a maior colaboração deste trabalho pois, a partir dela e do perfil dos professores apresentados no Sistema Estela Experta, (Figura 1), é comprovado que a experiência subjetiva dos avaliadores influencia diretamente no resultado do processo seletivo.

Devido a esta maior abrangência, projetos que dificilmente seriam contemplados com uma bolsa de pesquisa, quando se considera apenas a avaliação da PRPGI, poderiam ter maior oportunidade de participar do PIBIC. Acredita-se que esta característica poderia incentivar, ainda mais, professores e estudantes a estarem desenvolvendo mais pesquisas no IFBA.

8 Considerações Finais

8.1 Conclusões

Este trabalho buscou apresentar a aplicação do método TOPSIS como uma alternativa para a seleção de projetos em Instituições de Ensino, de forma a padronizar e aumentar a confiabilidade e isonomia no processo.

Para tanto, foi desenvolvido o sistema Topsisfy, que implementa a técnica TOPSIS. Com o objetivo de validar esta ferramenta, foram utilizados dados reais, cedidos pela PRPGI do IFBA, reguladora do processo seletivo do PIBIC na instituição. Estes dados corresponderam às avaliações de três anos do programa.

Além do parecer da própria Pró-reitoria de Pós-graduação e Inovação sobre os critérios definidos para a seleção, foi

considerada também a opinião de 17 professores do IFBA que lecionam em diferentes áreas do saber. Esta verificação foi feita com o objetivo de verificar a sinergia e estabelecer os pesos para os critérios da avaliação deste órgão com a comunidade docente do IFBA.

Ao longo deste trabalho, foi possível verificar que a opinião dos avaliadores destoa bastante acerca da importância dos critérios. Contudo, ao comparar a incidência dos projetos com melhor avaliação, verificou-se que boa parte dos resultados são similares, mas que ainda existem casos de projetos que são aprovados apenas em uma ou duas das avaliações.

Além disso, também foi questionado se o uso de um método de suporte à decisão poderia deixar estes processos seletivos mais ágeis e confiáveis. Partindo do pressuposto de que a TOPSIS é utilizada em diversas áreas do saber, conforme apresentado na Seção 2, o processo seletivo que a implementa tende a ser mais confiável. Em termos de desempenho, a partir da reavaliação dos processos mais antigos, houve um indicativo de melhora. Ainda assim, só será possível assegurar que há um ganho real de tempo quando o sistema for implementado para outros processos seletivos, de forma que seja possível comparar o tempo dos mesmos em relação aos anteriores à utilização do Topsisfy.

Entre os ganhos encontrados na utilização do Topsisfy, encontram-se a fundamentação da seleção na técnica TOPSIS, que é amplamente utilizada em diversas áreas. Além disso, existe também a facilidade de uma seleção colaborativa através das incidências de projetos nas seleções de vários avaliadores, que pode ser utilizada como um critério final para a seleção dos projetos aprovados.

Apesar de ter sido utilizado com o caso da seleção do PIBIC-IFBA, a modelagem do sistema permite que o Topsisfy também seja aplicado em outros contextos. Ou seja, o Topsisfy não é um *software* específico para a seleção de bolsistas, mas sim um sistema *web* que pode ser utilizado em quaisquer processos seletivos.

Ao longo do desenvolvimento deste trabalho foi possível constatar que a estadia do autor no curso de Tecnologia em Análise e Desenvolvimento de Sistemas do IFBA foi crucial para a criação deste trabalho. Houveram diversas disciplinas que foram imprescindíveis para a concepção, criação e análise dos resultados deste trabalho, a exemplo de: Lógica de Programação e Laboratório de Programação permitiram uma boa estruturação do passo a passo do algoritmo da TOPSIS, de forma que o mesmo pudesse ser utilizado no sistema; As disciplinas Orientação a Objetos, Padrões de Projeto, Banco de Dados e Engenharia de Software viabilizaram a estruturação do sistema de forma que o mesmo pudesse atender a demanda para qual foi criado; A disciplina de Redes ajudou bastante durante a configuração do *droplet* na Digital Ocean; A disciplina Programação Web foi imprescindível, por trazer os aspectos básicos para a programação de sistemas *web*, explorando conceitos como HTTP *Request/Response*, por exemplo; A disciplina de Arquitetura de Software permitiu ao autor ser mais crítico acerca da arquitetura da aplicação, tomando cuidados especiais com a utilização de *Components of the Shelf* (COTS) como bibliotecas e *frameworks*, além de atentar para as questões relacionadas aos componentes e conectores da aplicação.

8.2 Trabalhos Futuros

Levando em conta o escopo do trabalho e o seu potencial de crescimento, sugere-se para trabalhos futuros o desenvolvi-

mento das seguintes funcionalidades:

- Desenvolvimento de um módulo para auxiliar na determinação do peso dos critérios utilizados;
- Criar um módulo que viabilize o cadastro dos projetos a partir dos próprios pleiteantes;
- Desenvolver um mecanismo de tipos de usuário com permissões de acesso específicas ao sistema;
- Desenvolver um módulo de comparação de resultados entre diferentes avaliadores;
- Implementar a análise multi-cenário, para poder realizar uma análise de sensibilidade do processo ao manipular os pesos dos critérios durante a avaliação;
- Utilizar escalas com critérios estabelecidos por especialistas para utilizar conceitos da Lógica *Fuzzy* de forma a ter ainda mais precisão nos resultados obtidos;

9 Referencias

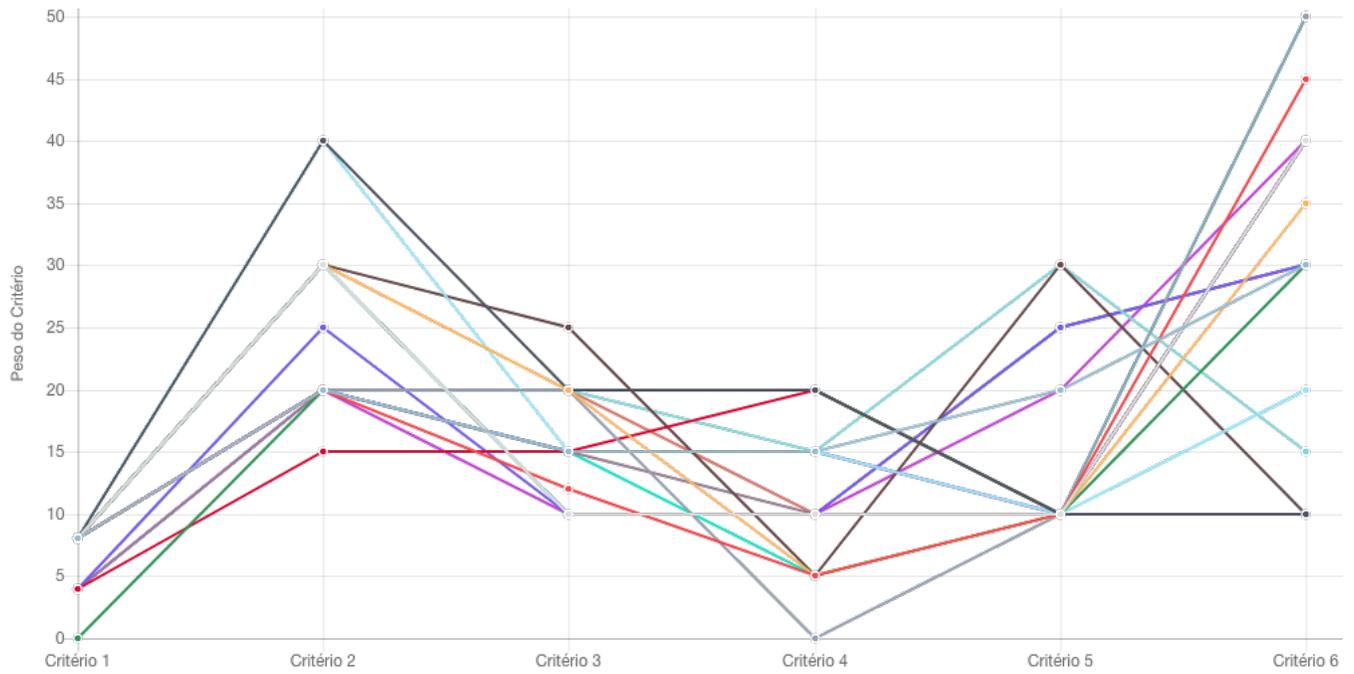
- [1] Y. Almoghatawi, K. Barker, C. M. Rocco, and C. D. Nicholson. A multi-criteria decision analysis approach for importance identification and ranking of network components. *Reliability Engineering & System Safety*, 158:142–151, 2017.
- [2] M. Bruno. 10 heurísticas de nielsen, uma fórmula para evitar erros básicos de usabilidade. <https://goo.gl/ozKoR7>, 2016. Último Acesso: 12 de Novembro de 2017.
- [3] S. BÜYÜKÖZKAN, Gülçin; GÜLERYÜZ. Multi criteria group decision making approach for smart phone selection using intuitionistic fuzzy topsis. *International Journal of Computational Intelligence Systems*, 2016.
- [4] A. G. Campolina, P. C. D. Soárez, F. V. d. Amaral, and J. M. Abe. Multi-criteria decision analysis for health technology resource allocation and assessment: so far and so near? *Cadernos de Saúde Pública*, 33(10), 2017.
- [5] T. A. Cardoso. Estruturação do processo decisório para reforma de edificações escolares públicas do ensino fundamental utilizando o método multicritério de apoio à decisão-constructivista (mcda-c). 2017.
- [6] CNPQ. Pibic. <http://cnpq.br/pibic>. Acesso em 05/10/2017.
- [7] M. Corporation. Common vulnerabilities and exposures. <http://cve.mitre.org/index.html>, 2017. Último Acesso: 13 de Novembro de 2017.
- [8] V. U. P. da Costa Miranda. Constructivismo e racionalismo como paradigmas para modelos de tomada de decisão estratégica. *Revista Eletrônica de Administração*, 5(2), 2006.
- [9] P. D. G. De Risco and D. D. I. S. a Lógica. Transparência, corrupção e fraudes.
- [10] P. M. R. de Souza. *UM ESTUDO SOBRE PADRÕES E TECNOLOGIAS PARA O DESENVOLVIMENTO WEB-FRONT-END*. PhD thesis, Universidade Federal do Rio de Janeiro, 2016.

- [11] H. Deiró. Web starter kit (weskit). <https://github.com/hdeiro/weskit>, 2017. Último Acesso: 2 de Novembro de 2017.
- [12] P. Demo. Educar pela pesquisa. In *Educação contemporânea*. Autores Associados, 2011.
- [13] G. N. S. ENSSLIN, L; MONTIBELLER. *Apoio a Decisão, Metodologias para Estruturação de Problemas e Avaliação Multicritério de Alternativas*. Ed. Insular, 2001.
- [14] L. Ensslin, S. R. Ensslin, S. Rocha, A. D. Marafon, and T. A. Medaglia. Modelo multicritério de apoio à decisão construtivista no processo de avaliação de fornecedores. *Production*, 23(2):402–421, 2013.
- [15] S. Experta. Sistema stela experta. <http://www.stelaexperta.com.br/ifba>. Acesso em 05/10/2017.
- [16] I. O. for Standardization. Iso/iec 9075-1. <https://www.iso.org/standard/45498.html>. Acesso em 11/11/2017.
- [17] M. Foundation. Mariadb. <https://mariadb.org/about/>, 2017. Último Acesso: 2 de Novembro de 2017.
- [18] N. Franz. Guia de profissões: Desenvolvedor(a) back-end. <http://tutano.tramos.co/10943-guia-de-profissoes-desenvolvedor-back-end/>. Acesso em 09/11/2017.
- [19] W. O. Galitz. *The essential guide to user interface design: an introduction to GUI design principles and techniques*. John Wiley & Sons, 2007.
- [20] S. Greco, J. Figueira, and M. Ehr Gott. Multiple criteria decision analysis. *Springer's International series*, 2005.
- [21] T. Guardian. What is front-end development? <https://www.theguardian.com/help/insideguardian/2009/sep/28/blogpost>, 2009. Último Acesso: 2 de Novembro de 2017.
- [22] GulpJS. Gulp. <https://gulpjs.com/>, 2017. Último Acesso: 2 de Novembro de 2017.
- [23] C. E. Hampton Catlin, Natalie Weizenbaum. Syntactically awesome style sheets (sass). <http://sass-lang.com/>, 2017. Último Acesso: 2 de Novembro de 2017.
- [24] He:labs. Desenvolvedor web! 6 dicas para quem precisa de um! <https://helabs.com/blog/desenvolvedor-web-seis-dicas-para-quem-precisa-de-um/>, 2017. Último Acesso: 2 de Novembro de 2017.
- [25] HostTime. 10 razões para migrar o mysql para o mariadb. <http://blog.hostdime.com.br/cpanel/10-razoes-para-migrar-o-mysql-para-mariadb/>, 2017. Último Acesso: 13 de Novembro de 2017.
- [26] IFBA. Prpgi. <http://www.prpgi.ifba.edu.br/>. Acesso em 05/10/2017.
- [27] InfoWorld. What's so special about google's angularjs. <https://www.infoworld.com/article/2612801/javascript/what-s-so-special-about-google-s-angularjs.html>, 2013. Último Acesso: 13 de Novembro de 2017.
- [28] E. International. Ecma international tc39-tg1. <https://www.ecma-international.org/memento/TC39.htm>, 2017. Último Acesso: 1 de Novembro de 2017.
- [29] D. S. Jasri and R. Rahim. Decision support system best employee assessments with technique for order of preference by similarity to ideal solution. *INTERNATIONAL JOURNAL OF RECENT TRENDS IN ENGINEERING & RESEARCH*, 3(3):6–17, 2017.
- [30] A. KROHLING, R; PACHECO. A-topsis - an approach based on topsis for ranking evolutionary algorithms. *Information Technology and Quantitative Management*, 2015.
- [31] A. A. Longaray, V. A. G. Tondolo, A. P. Gonçalves, P. R. Munhoz, M. E. Camargo, and R. d. R. P. Tondolo. Análise da produção científica brasileira sobre seleção de fornecedores apoiada em métodos multicritério. *Revista GEINTEC-Gestão, Inovação e Tecnologias*, 7(3):3970–3985, 2017.
- [32] C. E.-J. Maranhão. Topsis versus ahp: Métodos multicritério aplicados em um estudo de caso.
- [33] B. P. Menezes, P. M. Menezes, R. P. C. do Nascimento, and F. G. Rocha. Strategic planning methodology for information security–peseg 1.0. In *14th CONTECSI-International Conference on Information Systems and Technology Management*, 2017.
- [34] V. U. P. D. C. MIRANDA. Construtivismo e racionalismo como paradigmas para modelos de tomada de decisão estratégica. *REA-Revista Eletrônica de Administração*, 5(2), 2011.
- [35] Mozilla. Por que usar css? https://developer.mozilla.org/pt-BR/docs/Web/CSS/Getting_Started/Porque_usar_CSS, 2015. Último Acesso: 13 de Novembro de 2017.
- [36] Mozilla. Javascript. https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Introduction_to_Object-Oriented_JavaScript, 2017. Último Acesso: 13 de Novembro de 2017.
- [37] NPMJS. Node package manager (npm). <https://www.npmjs.com/>, 2017. Último Acesso: 2 de Novembro de 2017.
- [38] D. Ocean. Digital ocean. <https://www.digitalocean.com/>, 2015. Último Acesso: 13 de Novembro de 2017.
- [39] T. Otwell. Eloquent orm. <https://laravel.com/docs/5.5/eloquent>, 2017. Último Acesso: 2 de Novembro de 2017.
- [40] T. Otwell. Laravel. <https://laravel.com/docs/4.2/introduction>, 2017. Último Acesso: 2 de Novembro de 2017.
- [41] A. PACHECO. Topsis, algoritmo de tomada de decisão. <http://www.computacaointeligente.com.br/algoritmos/topsis-algoritmo-de-tomada-de-decisao/>. Acesso em 18/08/2016.
- [42] PRPGI-IFBA. Edital n° 01/2015/prpgi – pibic ifba/fapesb, pibiti e pibic-em ifba – seleção de bolsistas de ic&t– 2015/2016. <http://www.prpgi.ifba.edu.br/projpesq/>. Acessado em: 20170-07-10.
- [43] PRPGI-IFBA. Edital n° 02/2014/prpgi – pibic ifba/fapesb – seleção de bolsistas – 2014/2015. <http://www.prpgi.ifba.edu.br/projpesq/>. Acessado em: 20170-07-10.
- [44] PRPGI-IFBA. Edital n° 07/2013/prpgi –

- pibic/fapesb/ifba e pibiti/ifba – seleção de bolsistas – 2013/2014. <http://www.prpgi.ifba.edu.br/projpesq/>. Acessado em: 2017-07-10.
- [45] R. B. Qi Zhang, Lu Cheng. Cloud computing: state-of-the-art and research challenges. *Journal of Internet Server Applications*, 2010.
- [46] S. N. Ramez Elmasri. *Sistemas de Banco de Dados*. Pearson, 2006.
- [47] B. Roy and D. Vanderpooten. The european school of mcda: Emergence, basic features and current works. *Journal of Multi-Criteria Decision Analysis*, 5(1):22–38, 1996.
- [48] RubyGems. Sass. <https://rubygems.org/gems/sass>, 2017. Último Acesso: 13 de Novembro de 2017.
- [49] S. SAGHAFIAN, Sourosh; HEJAZI. Multi-criteria group decision making using a modified fuzzy topsis procedures. *International Conference on Computational Intelligence for Modelling, Control and Automation*, 2005.
- [50] R. Sales. O processo de decisão. portal da administração. <http://www.portal-administracao.com/2014/03/o-processo-decisorio-nas-organizacoes.html>. Acesso em 25/10/2015.
- [51] A. R. Sanches. Fundamentos de armazenamento e manipulação de dados. <https://www.ime.usp.br/~andres/aulas/bd2005-1/aula3.html>. Acesso em 09/11/2017.
- [52] E. I. TC39. Especificação ecma-262. <https://www.ecma-international.org/publications/standards/Ecma-262.htm>, 2017. Último Acesso: 1 de Novembro de 2017.
- [53] A. Team. Repositórios do angular. <https://github.com/angular>, 2017. Último Acesso: 2 de Novembro de 2017.
- [54] C. Team. Composer. <https://getcomposer.org/>, 2017. Último Acesso: 2 de Novembro de 2017.
- [55] P. Team. O que é php? https://secure.php.net/manual/pt_BR/intro-what-is.php, 2017. Último Acesso: 2 de Novembro de 2017.
- [56] P. Technologies. Postman. <https://www.getpostman.com/>, 2017. Último Acesso: 2 de Novembro de 2017.
- [57] F. Teixeira. *Introdução e boas práticas em UX Design*. Editora Casa do Código, 2014.
- [58] E. O. TELES. Modelagem e simulação de apoio à tomada de decisão na reativação de campos maduros terrestres e economicamente marginais de petróleo e gás. *Programa de Pós-Graduação em Engenharia Industrial*, 2016.

10 Apêndice

10.1 Resultados das Avaliações



10.2 Diagrama Entidade-Relacionamento

