

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

ERICK TAUÃ AMARAL DE PAULA

**PROTÓTIPO DE APLICAÇÃO WEB PARA IDENTIFICAÇÃO DE
TÉCNICAS DE LEVANTAMENTO DE REQUISITOS UTILIZANDO LÓGICA
FUZZY**

TOLEDO

2023

ERICK TAUÃ AMARAL DE PAULA

**PROTÓTIPO DE APLICAÇÃO WEB PARA IDENTIFICAÇÃO DE
TÉCNICAS DE LEVANTAMENTO DE REQUISITOS UTILIZANDO LÓGICA
FUZZY**

Trabalho de conclusão de curso de graduação apresentada como requisito para obtenção do título de Tecnólogo em Tecnologia em Sistemas para internet da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador(a): Suzan Kelly Borges Piovesan.
Coorientador(a): Ivan Luiz Salvadori.

TOLEDO

2023



[4.0 Internacional](https://creativecommons.org/licenses/by-sa/4.0/)

Esta licença permite remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es) e que licenciem as novas criações sob termos idênticos. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

ERICK TAUÃ AMARAL DE PAULA

**PROTÓTIPO DE APLICAÇÃO WEB PARA IDENTIFICAÇÃO DE
TÉCNICAS DE LEVANTAMENTO DE REQUISITOS UTILIZANDO LÓGICA
FUZZY**

Trabalho de conclusão de curso de graduação apresentado como requisito para obtenção do título de Tecnólogo em Sistemas para Internet da Universidade Tecnológica Federal do Paraná (UTFPR).

Data de aprovação: 27/novembro/2023

Suzan Kelly Borges Piovesan
Mestrado em Informática
Universidade Tecnológica Federal do Paraná

Rosane Fátima Passarini
Doutorado em Engenharia de Automação e Sistemas
Universidade Tecnológica Federal do Paraná

Roberto Milton Scheffel
Doutorado em Ciências da Computação
Universidade Tecnológica Federal do Paraná

**TOLEDO
2023**

AGRADECIMENTOS

Uso estes parágrafos para agradecer a todos que apoiaram este trabalho e a difícil jornada de chegar até esta etapa.

Agradeço à minha orientadora Professora Suzan Kelly Borges Piovesan por todo conhecimento passado, por me inspirar a ter disciplina e apoiar meu projeto do início ao fim.

Agradeço também ao coorientador Ivan Luiz Salvadori pelo auxílio durante o trabalho e pela disposição de fazer parte dessa escrita.

Sou muito grato aos meus amigos por me apoiarem com discursos motivacionais e por me ouvir minhas preocupações. E claro, gratidão a minha namorada, por acreditar em mim quando nem eu mesmo acreditava.

Também agradeço a minha família, que mesmo não fazendo ideia do que estava acontecendo, torciam pelo meu sucesso.

E por fim, sou grato aos professores que me ensinaram muito durante o curso.

RESUMO

Uma das fases mais importantes do desenvolvimento web é o levantamento de requisitos. Cada aplicação exige um levantamento de requisitos adequado para o contexto, o que traz o questionamento de como decidir a melhor técnica para o projeto. Para isso, foi desenvolvido uma aplicação web que utiliza a lógica fuzzy para auxiliar na escolha da técnica de levantamento de requisitos com base em informações sobre o projeto e a equipe de desenvolvimento. A validade da aplicação foi obtida através de questionários com profissionais da área de desenvolvimento de software.

Palavras-chave: levantamento de requisitos; lógica fuzzy; desenvolvimento web.

ABSTRACT

One of the most crucial phases in web development is requirement elicitation. Each application requires a suitable requirement elicitation process tailored to its context, raising the question of how to decide on the best technique for the project. To address this, a web application has been developed that utilizes fuzzy logic to assist in choosing the requirement elicitation technique based on information about the project and the development team. The validity of the application was obtained through questionnaires with professionals in the software development field.

Keywords: requirement elicitation; fuzzy logic; web development.

LISTA DE ILUSTRAÇÕES

Figura 1 - Representação na forma de conjuntos da altura de uma pessoa, sob o ponto de vista da Lógica convencional (à esquerda) e do da Lógica Fuzzy (à direita)	19
Figura 2 - Conjuntos fuzzy de uma variável temperatura	20
Figura 3 - Funções de pertinência para a variável estatura	21
Figura 4 - Lógica da função de pertinência triangular	22
Figura 5 - Gráfico da função triangular	22
Figura 6 - Lógica da função de pertinência trapezoidal	23
Figura 7 - Gráfico da função trapezoidal	23
Figura 8 - Copos	24
Figura 9 - Variável Linguística “Copo” e seus termos linguísticos .	25
Figura 10 - Comparativo entre as técnicas de levantamento de requisitos	28
Quadro 1 - Critérios utilizados no modelo e seus valores	29
Figura 11 - Representação da variável Tempo na modelagem Fuzzy	30
Figura 12 - Representação da variável Custo na modelagem Fuzzy	31
Figura 13 - Representação da variável Pessoas_Responsáveis na modelagem Fuzzy	31
Figura 14 - Representação da variável Imposição na modelagem Fuzzy	32
Figura 15 - Representação da variável confiabilidade na modelagem Fuzzy	32
Figura 16 - Representação da variável Padronização na modelagem Fuzzy	33
Figura 17 - Representação da variável Produtividade na modelagem Fuzzy	33
Figura 18 - Representação da variável de saída na modelagem Fuzzy	34
Figura 19 - Diagrama de classes da aplicação	38
Figura 20 - Diagrama de caso de uso da aplicação	39
Figura 21 - Diagrama de sequência da aplicação	40
Figura 22 - Tela de apresentação	49
Figura 23 - Pergunta final do questionário sobre o respondente	50
Figura 24 - Pergunta com campo de seleção e botão de finalizar	51
Figura 26 - Tela de resultado final	52
Quadro 2 - Quadro com os resultados obtidos 1	55

Quadro 3 - Quadro com os resultados obtidos 2.....	56
---	-----------

SUMÁRIO

1. INTRODUÇÃO.....	11
1.1. Objetivo geral	12
1.1.1. Objetivos específicos	12
1.2. Justificativa	12
2. REFERENCIAL TEÓRICO	13
2.1. Levantamento de requisitos	13
2.1.1. Elicitação de requisitos	13
2.1.2. Técnicas de levantamento de requisitos	14
2.1.2.1. Observação	14
2.1.2.2. Etnografia	14
2.1.2.3. Entrevista	15
2.1.2.4. Análise de protocolo	15
2.1.2.5. Prototipação	15
2.1.2.6. Cenários	16
2.1.2.7. Casos de uso	16
2.1.2.8. Brainstorming	16
2.1.2.9. JAD (Join Application Development)	17
2.1.2.10. Questionários	17
2.2. Lógica <i>fuzzy</i> e Inteligência Artificial	18
2.2.1. Conjuntos <i>fuzzy</i>	19
2.2.2. Variáveis linguísticas	20
2.2.3. Funções de pertinência	21
2.2.4. Inferência	24
2.2.5. Fuzzificação	24
2.2.6. Regras <i>fuzzy</i>	25
2.2.7. Defuzzificação	26
3. METODOLOGIA	27
3.1. Modelo Fuzzy para Escolha da Técnica de Levantamento de Requisitos	27
3.2. Tecnologias utilizadas	34
3.2.1. Flask	34
3.2.2. <i>SciKit-Fuzzy</i>	35
3.2.3. React JS	35
3.2.4. JSON	36

3.2.5. SQLite	36
3.3. Validação do questionário	36
4. RESULTADOS	37
4.1. Aplicação desenvolvida	37
4.2. Criação da base de conhecimento	41
4.3. Aplicação da lógica <i>fuzzy</i>	41
4.4. Questionário	42
4.5. Processamento das respostas	46
4.6. Interface do usuário	48
4.6.1. Apresentação	48
4.6.2. Questionário sobre o respondente	49
4.6.3. Questionário sobre o projeto	50
4.6.4. Saídas da Aplicação	51
4.7. Validação da aplicação (Testes)	52
5. CONCLUSÃO	58
6. REFERÊNCIAS	59

1. INTRODUÇÃO

Empresas com interesse em aprimorar algum setor, buscam pela implantação de softwares como ferramenta de negócio e para isso, procuram por empresas de desenvolvimento de software para tal. O desenvolvimento de um novo software é um processo com diversas etapas, umas das mais importantes, é o levantamento de requisitos. Geralmente é feito no início do projeto, porém, para o cliente, às vezes é difícil informar todos os requisitos logo no começo (RAMON, 2019). O levantamento de requisitos é onde se identifica junto do cliente todas as necessidades e expectativas em relação ao software a ser desenvolvido (MONGUILOD, 2022).

Aplicar uma técnica de levantamento de requisitos eficiente, permite conseguir informações mais relevantes para agregar valor ao projeto logo de início, e não apenas na entrega do projeto (MACHADO, 2018). Um software que atende os requisitos solicitados, pode ser considerado um software com qualidade. Neste aspecto, o levantamento de requisitos tem um papel fundamental no ciclo de vida do sistema. Podendo ser um importante guia para a elaboração quanto para a manutenção do projeto (SILVA, 2018).

Existem várias técnicas para o levantamento dos requisitos e para a escolha da melhor técnica é necessário considerar diversas variáveis do ambiente onde será aplicado, sendo assim um processo que exige um conhecimento prévio para atingir sucesso na decisão. Para alguns processos decisórios, atualmente são utilizados softwares desenvolvidos a partir de técnicas da Inteligência Artificial, que se baseiam em bases de conhecimento para classificar, inferir ou até mesmo apresentar resposta aos mais diversos problemas.

Dentre estas técnicas destaca-se a lógica fuzzy, lógica difusa, lógica nebulosa, ou até mesmo lógica multivalorada, que apresenta uma solução para definir conceitos que não são precisos e não podem ser definidos por apenas dois valores, como verdadeiro ou falso. De acordo com Marro (2010), lógica difusa tem como objetivo, a modelagem computacional do raciocínio humano, impreciso, ambíguo e vago. Com essa premissa, é possível aplicar lógica fuzzy para definir uma técnica de levantamento de requisitos eficiente

a partir de informações que categorizam o projeto em questão e a equipe de desenvolvimento.

1.1. Objetivo geral

O objetivo deste trabalho é apresentar o desenvolvimento de uma aplicação web para sugestão de técnica de levantamento de requisitos com base no modelo descrito no trabalho de MONGUILOD (2022), utilizando lógica fuzzy.

1.1.1. Objetivos específicos

1. Obter conhecimento sobre Lógica Fuzzy e construção de modelos usando esse método;
2. Pesquisar sobre bibliotecas Python que disponibilizam recursos para aplicar a Lógica Fuzzy.
3. Desenvolver uma aplicação web para sugestão de técnicas de levantamento de requisitos.
4. Realizar testes com profissionais da área de desenvolvimento de software para avaliar a validade da aplicação;
5. Disponibilizar um relatório com as informações recebidas.

1.2. Justificativa

Com base na necessidade de técnicas eficientes para levantar requisitos para agregar valor e atender as expectativas do cliente, a análise humana pode ser apoiada pelo uso de inteligência artificial e de algumas técnicas desse ramo, como a lógica fuzzy. O presente trabalho apresenta como a prática de inteligência artificial pode ser usada na escolha da técnica de levantamento de requisitos.

2. REFERENCIAL TEÓRICO

Este capítulo apresentará um contexto geral sobre levantamento de requisitos, lógica fuzzy e o modelo fuzzy para a escolha da técnica de levantamento de requisitos.

2.1. Levantamento de requisitos

De acordo com Machado (2018), a engenharia de requisitos proporciona o melhor entendimento das necessidades do cliente para garantir a qualidade de um software.

É a primeira etapa do processo de engenharia de software, a qual estuda a coleta, o entendimento, armazenamento, verificação e gerência dos requisitos. Tendo como principal preocupação entender os reais requisitos do sistema e a documentação deles. A própria engenharia de requisitos tem um processo dividido em fases, sendo elas: elicitación, o processo onde o cliente ou usuário é confrontado pelos desenvolvedores para explicitar o que querem no sistema; análise é o processo para avaliar as necessidades dos clientes para definir os requisitos; especificação é a documentação dos requisitos analisados; verificação é a busca por garantir que os requisitos especificados estão de acordo com as necessidades do cliente; por fim, o gerenciamento é o planejamento e controle dos demais processos (BELGAMO, 2008).

A aplicação desenvolvida foi construída com o objetivo de auxiliar na etapa de elicitación de requisitos. Portanto, as demais fases não serão abordadas em detalhes.

2.1.1. Elicitación de requisitos

Elicitación, também pode ser chamado apenas de levantamento de requisitos, é a atividade para entendimento dos requisitos de um sistema através da interação entre a equipe desenvolvedora e os clientes e usuários do sistema que se colocam à disposição para explicar as partes principais do sistema que precisam ser construídas (GALDINO, 2021).

É muito importante avaliar a experiência da pessoa que está fazendo o levantamento, tanto de quem está coletando, quanto de quem está

fornecendo as informações. Uma escolha errada pode acarretar desperdícios de tempo e prejuízos financeiros para a equipe e para os clientes (BELGAMO, 2008).

2.1.2. Técnicas de levantamento de requisitos

De acordo com Belgamo (2008), uma técnica deve explorar as características de um problema. Neste contexto, entende-se que cada projeto irá demandar um tipo de técnica, assim como a combinação de algumas técnicas podem ser úteis para ampliar a comunicação e o entendimento entre cliente e desenvolvedor. O levantamento de requisitos, porém, não pode ser resolvido apenas como soluções tecnológicas, deve ser considerado o contexto social que o sistema e as pessoas envolvidas estão inseridos. As técnicas a serem abordadas neste trabalho foram extraídas da pesquisa de Luana Monguilod (2022). Serão mostradas ainda outras técnicas de levantamento de requisitos, porém, não são utilizadas na aplicação vinculada ao trabalho.

2.1.2.1. Observação

Segundo Carvalho (2009), observar o comportamento e o ambiente de indivíduo que toma decisões pode ser uma forma eficaz de levantar informações que podem ser despercebidas em outras técnicas. Entre as principais vantagens da observação, é a interação pessoal do observador com o executor de uma tarefa. Isso permite ter uma noção do dia a dia e como é o ponto de vista da pessoa no processo. Porém, o método de observação pode ser criticado ao considerar a interpretação pessoal como a principal ferramenta para definição dos requisitos (BELGAMO, 2008).

2.1.2.2. Etnografia

A etnografia nada mais é do que uma técnica de observação aplicada na estrutura organizacional. Um observador anota sobre as tarefas reais do dia a dia em que os participantes estão envolvidos. A vantagem é que ela revela requisitos implícitos no contexto real das pessoas envolvidas no sistema, em oposição à visão formal definida pela organização (SOMMERVILLE, 2011).

Esse método tem como desvantagem, assim como a observação, segundo Belgamo (2008), a provocação de alterações no ambiente e na pessoa observada.

2.1.2.3. Entrevista

Carvalho(2009), explica a entrevista como uma conversa direcionada com um propósito específico. Isso inclui objetivos como saber a opinião do entrevistado; saber sentimentos do entrevistado sobre o estado do sistema; obter metas estabelecidas.

Entrevistas podem servir para um entendimento dos problemas e para elicitare requisitos gerais, de acordo com Belgamo(2008). Porém, muitas vezes, são menos efetivas para o entendimento do domínio da aplicação.

2.1.2.4. Análise de protocolo

Este método incentiva a pessoa a narrar a tarefa que está executando e explicar sua perspectiva do processo. A vantagem por trás da análise de protocolo é encontrada na ação de verbalizar o processo. Isso pode tornar o processo mais claro para a pessoa que está exercendo a análise. No entanto, a principal desvantagem, é a especificação do linguajar que acaba sendo entendido melhor por quem faz parte do negócio. Os clientes têm conhecimento sobre os negócios e o time de requisitos sobre as possibilidades técnicas, se essas pessoas não tiverem uma comunicação clara, pode haver distorções nos requisitos levantados (BELGAMO, 2008).

2.1.2.5. Prototipação

Na definição de Sommerville (2011), um protótipo é uma versão inicial do software que fornece algumas soluções breves para os problemas e serve para descobrir mais detalhes sobre as necessidades do cliente, já que este pode demonstrar em uma interface simples.

A principal vantagem ao criar um protótipo do que será feito, é poupar tempo e esforço para o desenvolvimento da aplicação real. Já o que pode ser considerado uma desvantagem, pelo tempo que precisa ser dedicado, não pela atividade em si, é a necessidade de uma documentação do protótipo

aprovado pelo usuário com todas as especificações que serão levadas adiante do processo (BELGAMO, 2008).

2.1.2.6. Cenários

Usuários finais e as pessoas no geral, acham mais fácil se relacionar com exemplos da vida real do que com descrições abstratas. Cenários são exercícios de interação do usuário e do sistema (BELGAMO, 2008).

Os cenários podem ser úteis para a adição de detalhes nos requisitos. Cada cenário cobre algumas interações possíveis, assim, diversas cenas podem ser criadas, variando o tipo e o detalhamento das informações (SOMMERVILLE, 2011).

Entre as desvantagens, pode-se dizer a falta de detalhes em alguns casos, seja pelo cenário não ser tão simples de explicar, ou não estar tão claro para o usuário.

2.1.2.7. Casos de uso

Um caso de uso identifica os envolvidos em uma interação e dá um nome ao tipo da interação. Junto com outras informações descritivas, permitem definir a interação com o sistema. (SOMMERVILLE, 2011).

A documentação é feita com diagramas de alto nível. O conjunto de casos de uso apresenta todas as interações possíveis com o sistema. Pelo fato de ser visual e explícito para o usuário, é uma excelente forma de ser compreendido. Em contrapartida, alguns casos podem ser o oposto, pela sua complexidade acabam sendo difíceis de entender. (MONGUILOD, 2022).

2.1.2.8. Brainstorming

São sessões de conversa entre um grupo de participantes que compartilham ideias e discussões sobre os requisitos diante de cenários específicos, como exemplifica Galdino (2021). A técnica preza pela interação e engajamento dos participantes, onde nenhuma ideia sugerida é descartada, o que deixa os envolvidos mais confortáveis para continuar interagindo.

Brainstorming quando bem conduzido, resulta em boas ideias e um grupo conectado e sentindo que contribuíram para a solução do problema.

(FIGUEIRA, 2012). Luana Monguilod (2022) coloca que as desvantagens estão ligadas ao custo e a sugestão de ideias inviáveis para o projeto.

2.1.2.9. JAD (Join Application Development)

Técnica desenvolvida pela IBM, consiste em workshops com diversas dinâmicas de grupo e recursos visuais para promover a troca de informações e o entendimento das mesmas entre os desenvolvedores e os *stakeholders* (FIGUEIRA, 2012). As sessões possuem um facilitador que conduz a dinâmica, questionando os participantes e gerando discussões com o intuito de obter informações para preencher os requisitos.

Na visão de Figueira (2012), possui duas grandes etapas: o planejamento, onde é feito o levantamento de requisitos; e o projeto em que se lida com o software em si. Porém, na visão de Belgamo (2008), consiste em 5 fases: definição do projeto e pesquisa, que lidam com a coleta de informações; preparação da sessão, a sessão JAD e o documento final, que acabam resultando na análise do projeto, complementando a ideia proposta anteriormente.

Como vantagem, é justo elencar que JAD promove cooperação e entendimento entre as partes envolvidas, resultando em diversos debates sobre os requisitos e, conseqüentemente, informações mais relevantes. A desvantagem seria a obrigatoriedade de um mediador para evitar conflitos e conduzir a reunião, mas sem ele, esse método de levantamento acabaria sendo mais demorado e custoso.

2.1.2.10. Questionários

Nessa técnica, os *stakeholders* do sistema respondem a perguntas pré-estabelecidas para a coleta de informações sobre as necessidades, expectativas e restrições do sistema. (SILVA, F, 2023).

De acordo com Chein (2018), os questionários têm a vantagem de não ter um entrevistador, o que garante tempo, além de ser possível alcançar um número maior de respondentes, já que é uma ferramenta que pode ser utilizada remotamente. Em contrapartida da presença direta de um mediador para analisar as respostas, as perguntas precisam ser claras e objetivas, o

que também limita a possibilidade de respostas, porque aceitar alegações subjetivas, podem levar a divergências ou dificuldade de entendimento.

As diversas técnicas de levantamento de requisitos utilizadas no mercado, bem como a maneira com que são escolhidas de acordo com o ambiente onde serão aplicadas, podem compor uma base de conhecimento passível de ser utilizada em sistemas inteligentes que auxiliam este processo decisório. A próxima seção apresenta algumas características importantes destes softwares.

2.2. Lógica *fuzzy* e Inteligência Artificial

Esta seção introduz os principais conceitos sobre Inteligência Artificial e alguns de seus métodos com destaque para a lógica fuzzy. Ainda aborda algumas aplicações e principais diferenças entre a lógica clássica e lógica difusa.

Sellitto (2002), sintetiza em sua pesquisa a definição de inteligência artificial como o campo da tecnologia que busca resolver problemas com a competência de um ser humano, utilizando recursos computacionais. Para alcançar esse objetivo existem diversas técnicas disponíveis para a aplicação desse conceito, como sistemas especialistas, raciocínio baseado em casos e a lógica fuzzy.

Sellitto explica que os sistemas especialistas utilizam uma base de conhecimento para realizar análises e reconhecer padrões, por isso, aprendem com a experiência. Já o raciocínio baseado em casos resolve problemas com base em soluções de problemas já conhecidos.

E a lógica fuzzy se baseia em variáveis linguísticas que emulam elementos do pensamento humano com bases comparativas. Enquanto a lógica convencional usa distinções bem definidas de seus conjuntos, colocando uma unidade em um conjunto ou em outro, classificando apenas em características extremas. A lógica fuzzy tenta demonstrar a maneira como as pessoas pensam, modelando o senso de palavras, a tomada de decisão e o senso comum. Ela utiliza a ideia de que todas as coisas podem ter graus de pertinência. (MARRO, 2010).

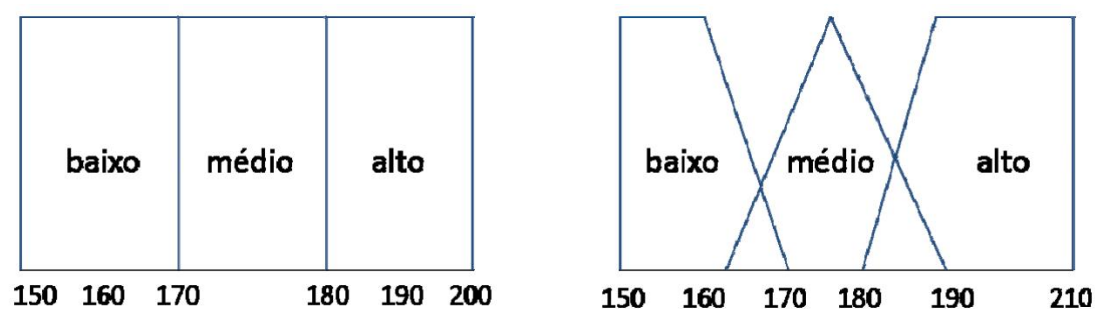
De acordo com Aguado (2010), a lógica fuzzy dá espaço para mensurar um grau de aproximação de uma solução para inferir algo. Diferente da lógica clássica, onde só existe verdadeiro e falso.

2.2.1. Conjuntos *fuzzy*

Como já discutido por Marro (2010), a lógica clássica limita os elementos a pertencerem a apenas um grupo. O exemplo apresentado para explicar o conceito de conjuntos é uma análise sobre a altura de uma pessoa. Uma pessoa com 1,69m poderia ser considerada de altura mediana em um conjunto de lógica clássica, já uma pessoa de 1,71m faz parte de um conjunto de pessoas altas. Porém, é uma comparação complexa de entender como duas pessoas de alturas tão próximas fazem parte de grupos distintos.

Pela ótica da lógica nebulosa, as duas pessoas teriam um certo grau de pertinência para pertencer às duas classificações. Dessa forma, é possível ter decisões mais humanas e menos rígidas. Para exemplificar, Marro apresenta dois gráficos de comparação entre a lógica clássica e a lógica fuzzy (figura 1).

Figura 1 - Representação na forma de conjuntos da altura de uma pessoa, sob o ponto de vista da Lógica convencional (à esquerda) e do da Lógica Fuzzy (à direita)



Fonte: MARRO (2010)

Na lógica clássica, os valores pertencem a um único grupo, como pode ser observado no gráfico da esquerda (figura 1). Já na lógica fuzzy, os valores podem pertencer tanto a um grupo quanto a outro. Por exemplo, uma pessoa com 1,87m de altura é categorizado como alto na lógica clássica, contudo, precisa ser levado em conta o contexto analisado. Se em um determinado contexto, a média de altura das pessoas é maior, então uma pessoa de 1,87m

não é necessariamente alta, ela pode pertencer ao grupo de pessoas médias. A lógica fuzzy permite adequar uma variável em um conjunto conforme o contexto aplicado.

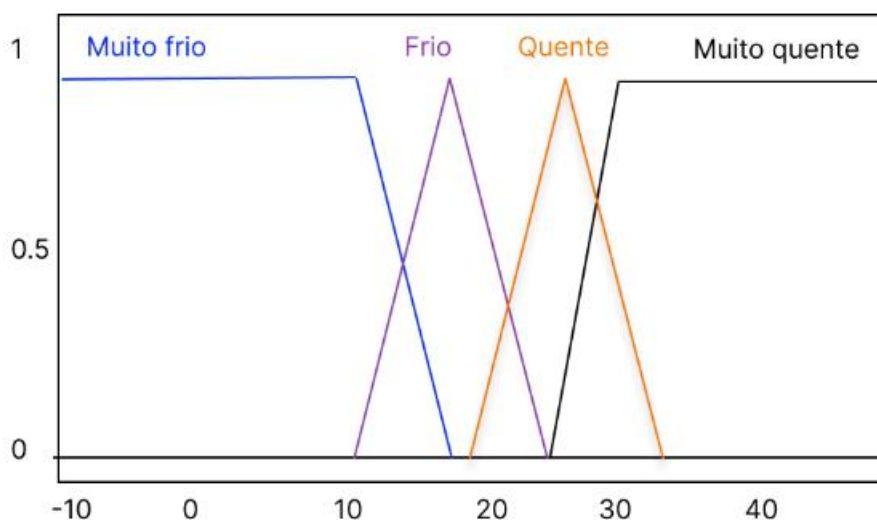
2.2.2. Variáveis linguísticas

Tanscheit (2004) aponta variável linguística como uma variável cujos valores são nomes de conjuntos *fuzzy*. Como a temperatura, que pode ser classificada como baixa, média e alta. Esses valores são descritos na lógica *fuzzy* por meio de funções de pertinência. A principal função dessas variáveis é fornecer uma maneira de caracterizar fenômenos complexos e com certo grau de incerteza.

Os valores de uma variável podem ser de uma linguagem específica usando termos primários como *alto*, *baixo*, *pequeno*, *grande*, também pode haver conectivos lógicos, como a negação *não* e os conectivos *e* e *ou*. Além de modificadores como *muito*, *pouco*, *levemente*, *extremamente*. Essas variações estão indicadas no formato de gráfico na figura 2.

Figura 2 - Conjuntos fuzzy de uma variável temperatura

Conjuntos fuzzy correspondentes à variável temperatura



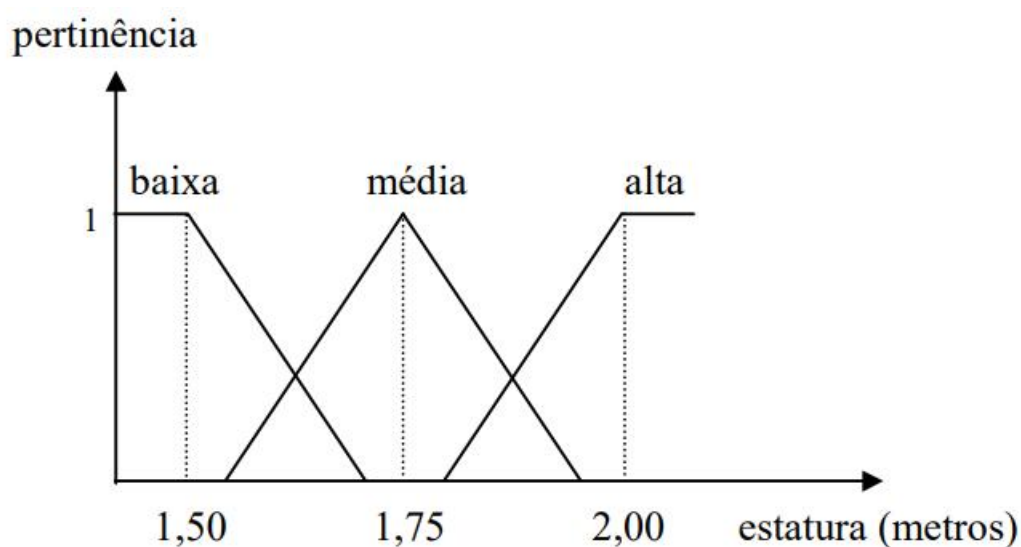
Fonte: Autoria própria (2023)

2.2.3. Funções de pertinência

As funções de pertinência variam conforme o contexto aplicado. Elas podem ser definidas conforme a experiência do usuário. Porém, algumas são mais usadas por terem cálculos diretos, como a função triangular, trapezoidal e gaussiana (Tanscheit, 2004).

Tanscheit também traz uma exemplificação por meio da estatura de uma pessoa, como mostra a figura abaixo, utilizando os seguintes termos: $T(\text{estatura}) = \{\text{baixa}, \text{média}, \text{alta}\}$, sendo esses os conjuntos com determinados graus de pertinência (Figura 3).

Figura 3 - Funções de pertinência para a variável estatura



Fonte: Tanscheit (2004)

No gráfico apresentado na figura 2, uma estatura de até 1,5 metros apresenta grau de pertinência igual a 1 no conjunto *baixa* e vai diminuindo sua pertinência conforme aumenta a altura. No conjunto *alta*, uma pessoa pode ser colocada nessa categoria a partir de 2 metros. Porém, essas definições variam de contexto, como complementa Tanscheit, exemplificando que existem padrões de alturas diferentes em cada país.

As funções de pertinência mais utilizadas são a função triangular e a função trapezoidal. A função de pertinência é triangular se as variáveis tiverem o comportamento $a < b < c$ a partir da fórmula apresentada na figura 4.

Figura 4 - Lógica da função de pertinência triangular

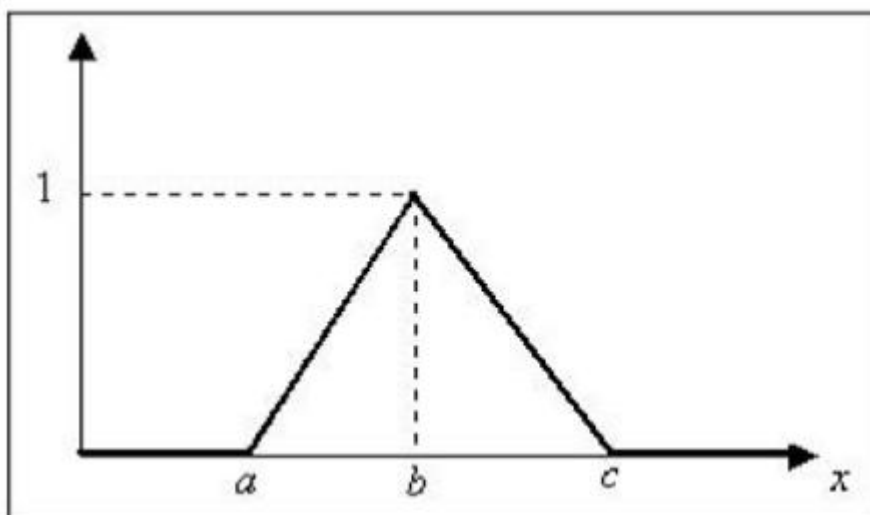
$$\mu_A(x) = \begin{cases} 0, & \text{se } x \leq a \\ \frac{x-a}{b-a}, & \text{se } a < x \leq b \\ \frac{x-c}{b-c}, & \text{se } b < x \leq c \\ 0, & \text{se } x > c \end{cases}$$

Fonte: CREMASCO (2010)

O gráfico da função fuzzy triangular tem o formato de um triângulo, utilizando como base o intervalo $[a,c]$ e um único vértice fora da base, o ponto b (Figura 4). (CREMASCO, 2010).

Serve para definir o grau de pertinência de uma variável x em relação a outras variáveis a , b e c . A pertinência será 0 no caso do valor x ser menor ou igual ao valor a , ou se a variável x ser maior que a variável c . Nos casos da variável x estar entre os valores a e b ou entre os valores b e c , é realizado um cálculo para definir seu grau de pertinência, cujo resultado será entre 0 e 1.

Figura 5 - Gráfico da função triangular



Fonte: CREMASCO (2010)

Já a função de pertinência trapezoidal, envolve uma variável a mais do que a triangular, seguindo a forma $a < b < c < d$, apresentada na figura 6.

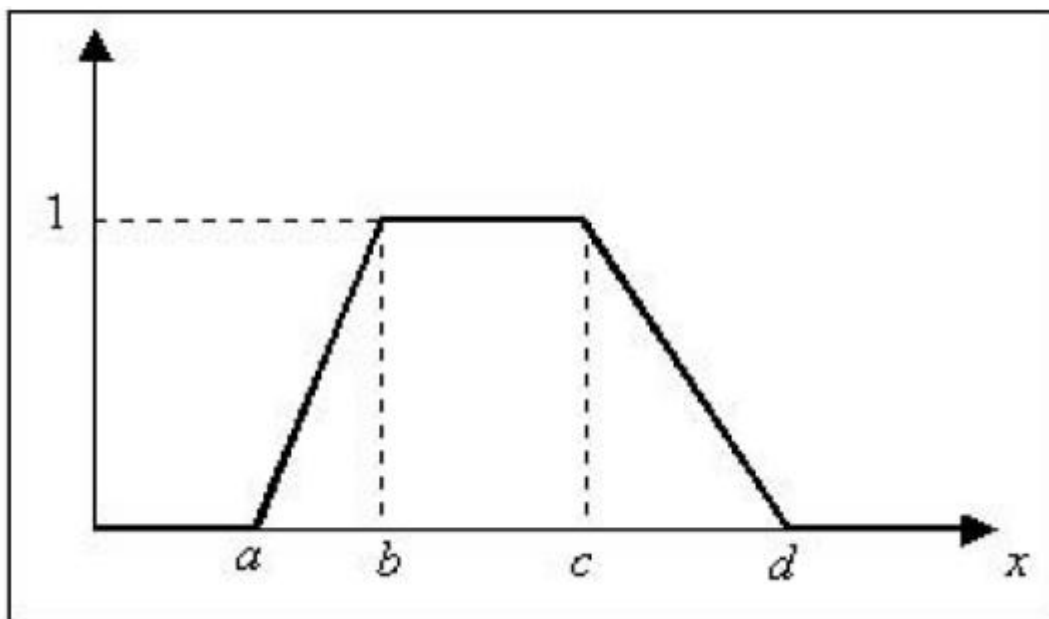
Conseqüentemente, seu gráfico se comporta com o intervalo $[a,d]$ próximos a base, e o intervalo $[b,c]$ ficam no ponto mais próximo de 1 (Figura 7).

Figura 6 - Lógica da função de pertinência trapezoidal

$$\mu_A(x) = \begin{cases} 0, & \text{se } x \leq a \\ \frac{x-a}{b-a}, & \text{se } a < x \leq b \\ 1, & \text{se } b < x \leq c \\ \frac{x-c}{b-c}, & \text{se } c < x \leq d \\ 0, & \text{se } x > d \end{cases}$$

Fonte: CREMASCO (2010)

Figura 7 - Gráfico da função trapezoidal



Fonte: CREMASCO (2010)

O cálculo para o grau de pertinência de uma função trapezoidal (figura 6) é semelhante ao da função triangular, contudo, a variável x é observada com base em quatro outras variáveis: a , b , c e d . A pertinência será 0 no caso do valor x ser menor ou igual ao valor a , ou se a variável x ser maior que a variável d . Nos casos da variável x estar entre os valores a e b ou entre os valores c e d , é realizado um cálculo para definir seu grau de pertinência, cujo

resultado será entre 0 e 1. E por fim, se a variável x estiver entre os valores de b e c , o resultado será 1.

2.2.4. Inferência

Segundo Marro (2010), a inferência é a avaliação das entradas com o objetivo de obter conclusões a partir de regras predefinidas. Existem modelos de inferência para essa tarefa. O mais utilizado e que foi escolhido para a modelagem do sistema deste trabalho foi o Mamdani.

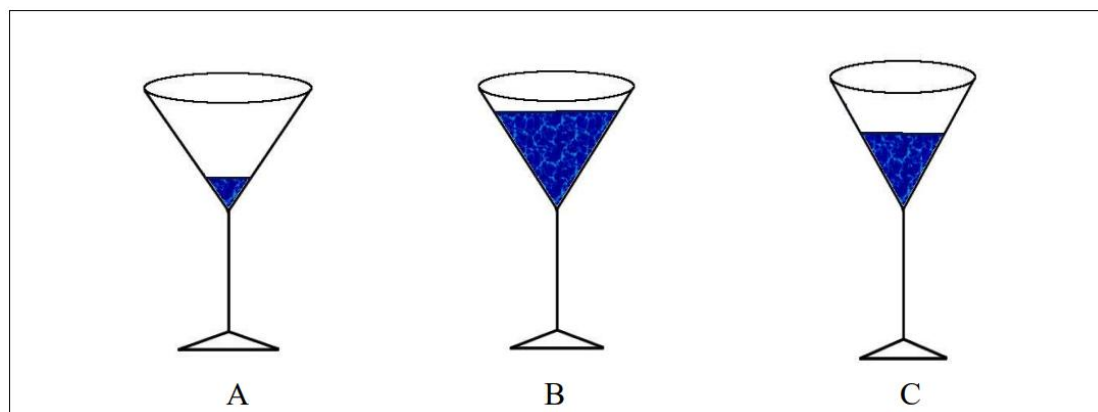
O método Mamdani utiliza a lógica de SE-ENTÃO, onde cada regra é implementada com base nos operadores lógicos para o processamento do antecedente e da função de implicação que define o consequente. Ou seja, se x é A e y é B , então z é C . A união dos conjuntos A e B por meio de um operador lógico, retorna um número de pertinência correspondente a algum valor do conjunto (ANDRADE, 2008).

As regras e a operação serão explicadas mais detalhadamente em seguida.

2.2.5. Fuzzificação

Nessa etapa, o problema é analisado e os dados de entrada são transformados em variáveis linguísticas. (AGUADO, 2010). O exemplo apresentado é de um conjunto de copos com diferentes níveis na figura 8.

Figura 8 - Copos

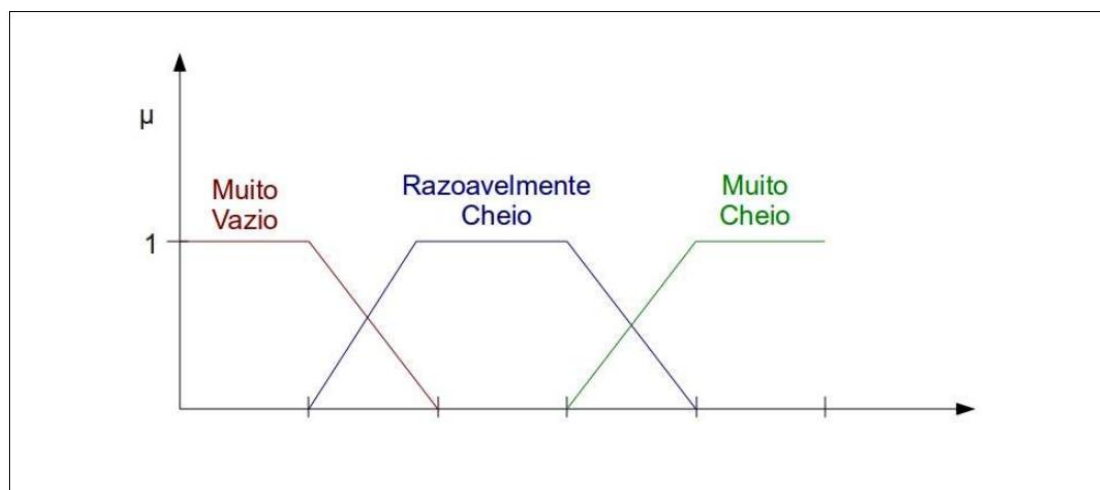


Fonte: AGUADO (2010)

Como mostrado na figura 8, a variável linguística “Copo” pode receber três termos distintos: *muito vazio*, *muito cheio* e *parcialmente cheio*. Aplicando

funções de pertinência nesse exemplo é possível obter o gráfico mostrado na figura 9 como resultado. Nesse exemplo, a variável linguística foi representada com palavras, mas como será mostrado na aplicação desenvolvida, é atribuído valores numéricos para cada variável. (AGUADO, 2010).

Figura 9 - Variável Linguística “Copo” e seus termos linguísticos



FONTE: AGUADO(2010)

Como mostra o gráfico da figura 9, a variável linguística “Copo” pode ter três termos linguísticos que vão variar de acordo com o grau de pertinência. Para isso são utilizados conjuntos de valores numéricos atribuídos a cada um dos termos. Para definir a qual conjunto a variável “Copo” pertence, é atribuído um valor numérico para ele e calculado sua função de pertinência. Quanto mais próxima de 1, maior o grau de pertencimento a um conjunto. Caso o valor numérico esteja presente em mais de um conjunto, então a variável pertence aos dois conjuntos.

2.2.6. Regras *fuzzy*

Os conjuntos *fuzzy* representados por variáveis relacionadas por funções de pertinência são chamadas de base de conhecimento. Para manipular essa base de conhecimento é utilizado um conjunto de regras de inferência. Para representar a linguagem natural, é utilizada a expressão lógica mais próxima da fala humana, como: SE (antecedente) ENTÃO (consequente). Apesar das incertezas presentes na base de conhecimento,

isso é mitigado com a entrada e saída de conjuntos *fuzzy* para manipulação da base. (Baroni, 2014).

Após a inserção dos valores fuzzyficados na modelagem, aplica-se nos antecedentes e retorna o valor do consequente para cada regra estabelecida. Podem haver antecedentes compostos utilizando os operadores *e* e *ou*. (Marro, 2010).

É importante ressaltar que tanto as regras, quanto as variáveis precisam ser validadas por especialistas do domínio. O modelo se torna inviável em caso de ausência de uma equipe especializada no assunto, como justifica Silva (2020).

2.2.7. Defuzzificação

É a etapa onde os valores *fuzzy* são convertidos em números reais, segundo Aguado (2010). Existem técnicas diferentes para defuzzificação, sendo as mais usadas centróide e média dos máximos.

Centróide obtém o resultado a partir do centro de gravidade do conjunto, ou seja, o número mais próximo do centro do valores.

A média dos máximos toma a média dos dois valores mais ao extremos do conjunto em questão, sendo os maiores valores da função de pertinência do conjunto *fuzzy* de saída.

3. METODOLOGIA

Neste capítulo será apresentado o modelo utilizado para a construção da parte de lógica fuzzy do sistema, bem como as tecnologias utilizadas para o desenvolvimento do front-end e do back-end, o banco de dados criados para o projeto e a ferramenta de transação de dados. Além de explicar o método de utilização dos dados coletados da modelagem fuzzy e a maneira de validação do sistema.

3.1. Modelo Fuzzy para Escolha da Técnica de Levantamento de Requisitos

Este trabalho foi realizado com base na pesquisa realizada por Monguilod (2022), cujo objetivo foi desenvolver a modelagem *fuzzy* para a identificação de métodos de levantamento de requisitos. Esta seção é destinada a apresentar os resultados obtidos desta pesquisa que permitiram o desenvolvimento da aplicação.

A premissa do trabalho foi utilizar a ferramenta Matlab para sugerir a melhor técnica para levantamento de requisitos, de acordo com um cenário específico.

A partir de pesquisas bibliográficas e questionários respondidos por profissionais da área de engenharia de software, Monguilod foi capaz de definir as técnicas mais utilizadas para levantamento de requisitos e também definiu quais parâmetros para analisar a escolha da técnica.

Para chegar nas técnicas explicadas na seção 2.1.2, foram utilizados os seguintes critérios:

- Tempo: o tempo utilizado para a aplicação da técnica, considerando também o tempo de preparação, sendo classificado em médio ou longo;
- Custo: o custo gasto para a aplicação da técnica, classificado em baixo, médio ou alto;

- Pessoas responsáveis: quantidade de pessoas participantes da aplicação da técnica, podendo ser classificado em indivíduo, grupo;
- Imposição: termo explicado como o nível de exposição que o participante precisa para ser entendido. Esse critério foi classificado em baixo, médio ou alto;
- Confiabilidade: qualidade das informações coletadas, se elas são confiáveis para o desenvolvimento do sistema ou não. Classifica-se em baixa, média ou alta;
- Padronização: se há algum tipo de regra ou padrão para ser seguido durante a aplicação da técnica. Pode ser apenas sim ou não;
- Produtividade: nível de produtividade da técnica, classificada como baixa, média ou alta.

A combinação desses critérios resulta em uma variável de saída, a técnica. Os valores qualitativos e as técnicas resultantes estão descritos na figura 10.

Figura 10 - Comparativo entre as técnicas de levantamento de requisitos

	Pessoas Responsáveis	Tempo	Custo	Imposição	Confiabilidade	Padronização	Produtividade
OBSERVAÇÃO	GRUPO/INDIVÍDUO	LONGO	BAIXO	BAIXO	MÉDIA	NÃO	BAIXA
ENTREVISTA	INDIVÍDUO	MÉDIO	MÉDIO	MÉDIO	ALTA	NÃO	MÉDIA
ANÁLISE DE PROTOCOLO	INDIVÍDUO	MÉDIO	MÉDIO	BAIXO	BAIXA	SIM	BAIXA
PROTOTIPAÇÃO	GRUPO/INDIVÍDUO	MÉDIO	ALTO	MÉDIO	ALTA	SIM	ALTA
CENÁRIOS	GRUPO/INDIVÍDUO	LONGO	ALTO	BAIXO	ALTA	SIM	ALTA
CASODE USO	INDIVÍDUO	LONGO	ALTO	BAIXO	ALTA	SIM	ALTA
ETNOGRAFIA	GRUPO/INDIVÍDUO	LONGO	BAIXO	BAIXO	MÉDIA	NÃO	BAIXA
BRAINSTORMING	GRUPO	LONGO	MÉDIO	BAIXO	ALTA	NÃO	MÉDIA
JAD	GRUPO	LONGO	ALTO	ALTO	ALTA	SIM	ALTA
QUESTIONÁRIOS	INDIVÍDUO	MÉDIO	BAIXO	BAIXO	MÉDIA	NÃO	MÉDIA

Fonte: MONGUILOD, 2022

Além da definição qualitativa dos critérios, a pesquisa também trouxe valores numéricos para serem utilizados na fase de fuzzificação. As variáveis de entrada são os critérios apresentados e como variável de saída, foi definido a técnica de levantamento de requisitos que mais se adequa aos critérios.

Para cada variável de entrada foi determinado um valor de entrada durante a fuzzificação, após calculado, o modelo retornava uma resposta com a defuzzificação por meio das variáveis de saída. No quadro 1 foi informado os valores escolhidos para cada variável e seus respectivos termos.

Quadro 1 - Critérios utilizados no modelo e seus valores

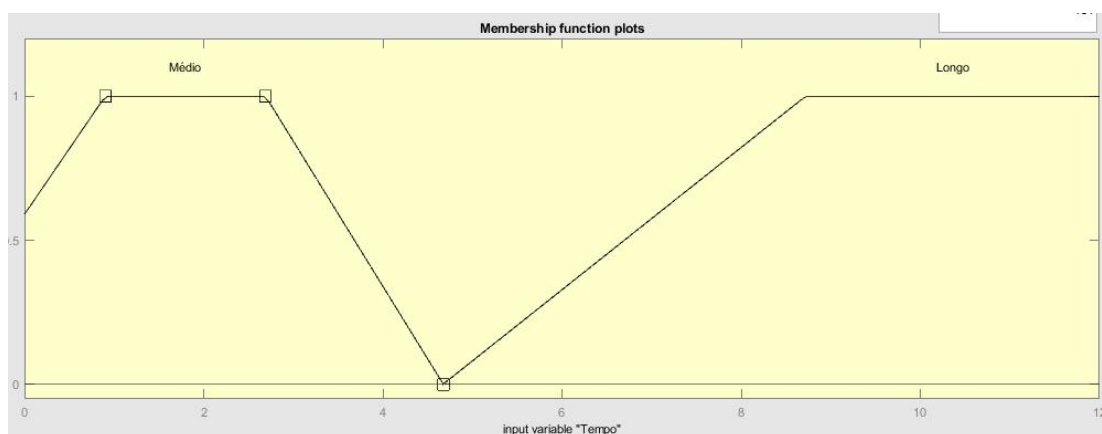
Tempo	Médio: [-1.3, 0.902, 2.69, 4.677] Longo: [4.677, 8.722, 12.05, 13.39]
Custo	Baixo: [-5, 0, 70, 95] Médio: [95.5, 110, 130] Alto: [130, 160, 180, 250]
Pessoas envolvidas	Ind_ou_grupo Grupo Indivíduo
Imposição	Baixo: [2, 4, 6, 8] Médio: [8, 12, 20, 24] Alto: [24, 32, 40, 51]
Confiabilidade	Baixa: [-4.086, 16.01, 34.01] Média: [30, 45, 60] Alta: [59.95, 87.38, 129.1]
Padronização	Sim Não

Produtividade	Baixa: [0, 15, 30] Média: [30, 45, 60] Alta: [60, 75, 90]
---------------	---

Fonte: Autoria própria (2023)

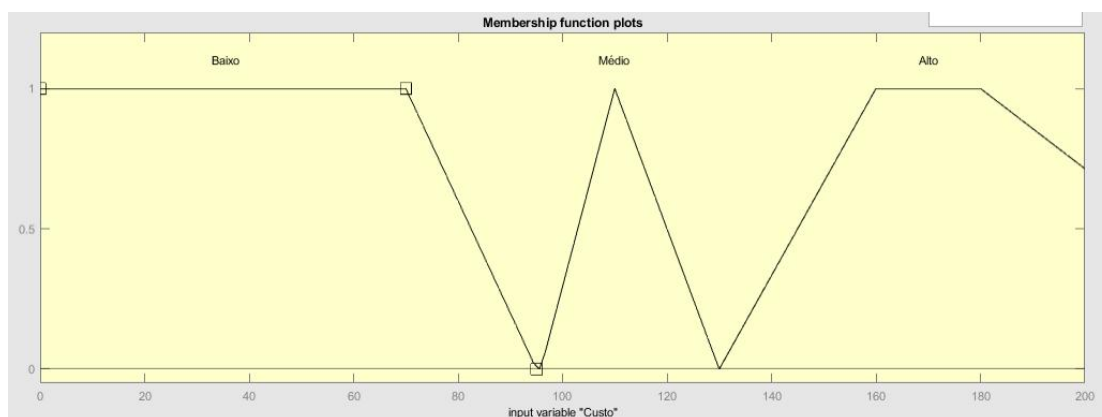
Para o critério tempo foi aplicado a função trapezoidal para definir um intervalo correspondente aos termos. Ao analisar a figura 11, tem se o intervalo utilizado.

Figura 11 - Representação da variável Tempo na modelagem Fuzzy



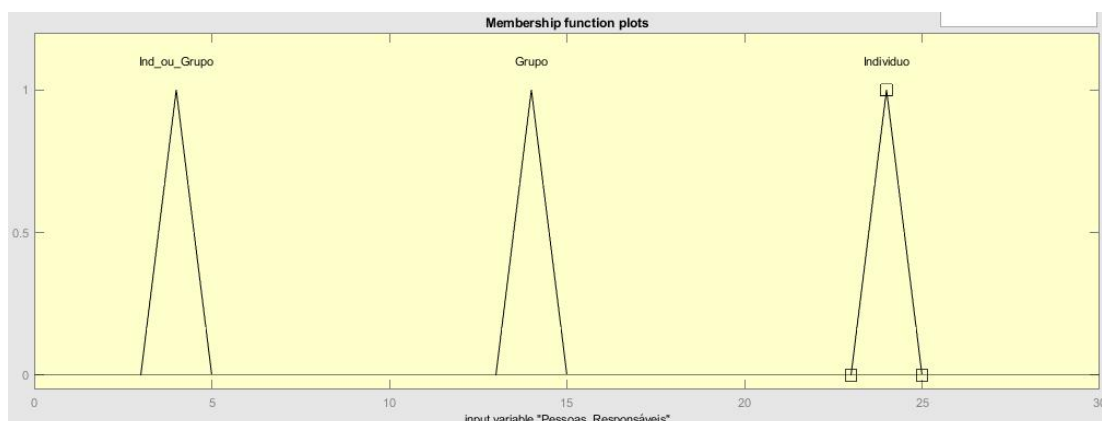
Fonte: MONGUILOD (2022)

Para o critério custo foi utilizado a função trapezoidal para definir os intervalos para as características baixo e alto, e a função triangular para definir um ponto médio para a terceira característica, como demonstrado na figura 12.

Figura 12 - Representação da variável Custo na modelagem Fuzzy

Fonte: MONGUILOD (2022)

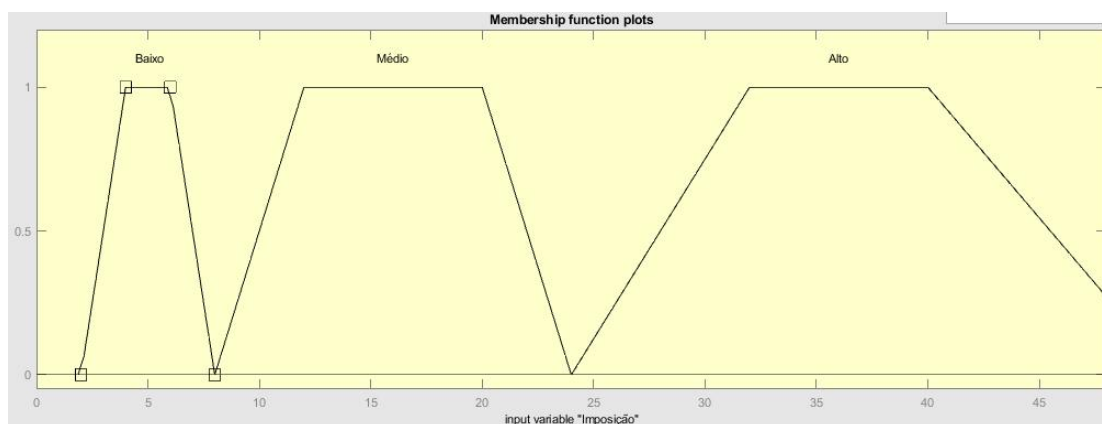
A variável pessoas envolvidas é um valor discreto, portanto, não há valores definidos. Sendo assim, a linha de cada termo não se encontra, como pode ser visto na figura 13.

Figura 13 - Representação da variável Pessoas_Responsáveis na modelagem Fuzzy

Fonte: MONGUILOD (2022)

Para a variável imposição, foi utilizado função trapezoidal para definir um intervalo de valores, como mostra a figura 14.

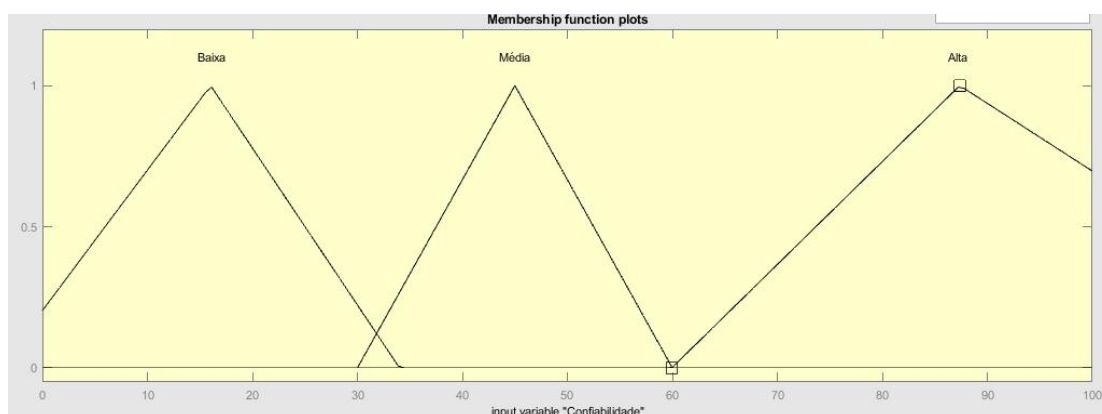
Figura 14 - Representação da variável Imposição na modelagem Fuzzy



Fonte: MONGUILOD (2022)

A variável confiabilidade utiliza função triangular para definir um ponto máximo em cada termo, como é mostrado na figura 15.

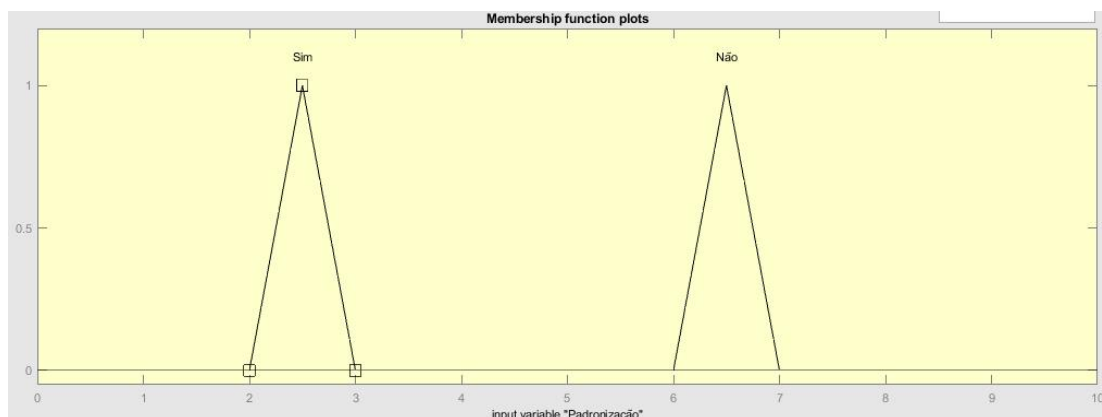
Figura 15 - Representação da variável confiabilidade na modelagem Fuzzy



Fonte: MONGUILOD (2022)

A variável de padronização é bi-valorada, por essa razão, não tem valores definidos. Cada termo possui um ponto específico, como mostra a figura 16.

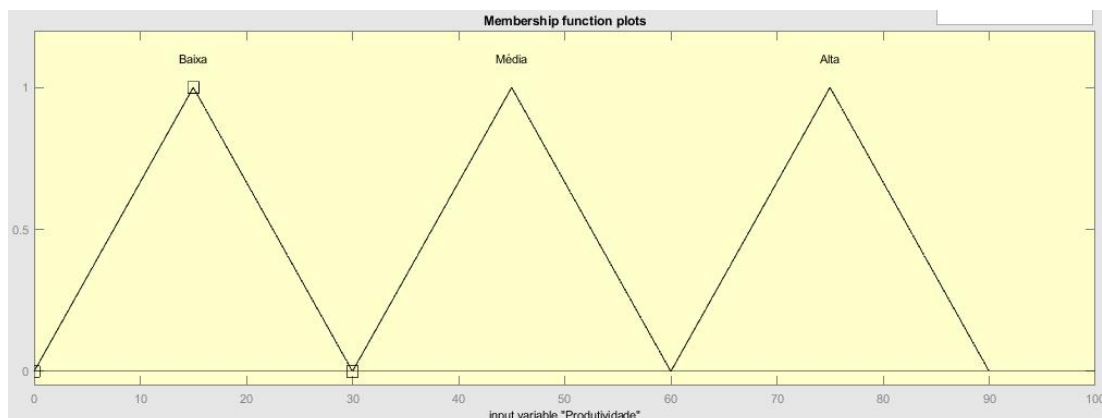
Figura 16 - Representação da variável Padronização na modelagem Fuzzy



Fonte: MONGUILOD (2022)

Por fim, a variável produtividade foi definida com função triangular para definir um ponto máximo em cada termo, como mostra a figura 17.

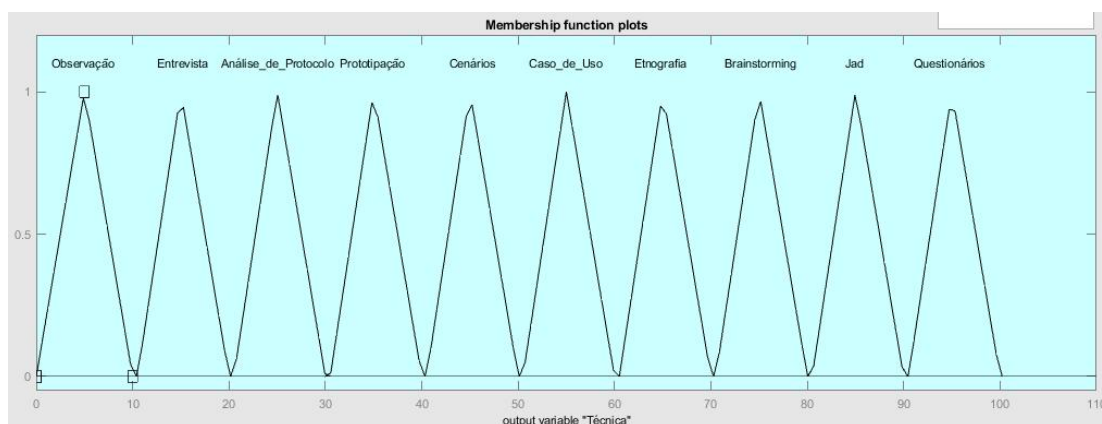
Figura 17 - Representação da variável Produtividade na modelagem Fuzzy



Fonte: MONGUILOD (2022)

A variável de saída técnica é um valor exato para elencar cada técnica, por isso, foram utilizados valores bi-valorados. Os pontos máximos das técnicas podem ser observados na figura 18.

Figura 18 - Representação da variável de saída na modelagem Fuzzy



Fonte: MONGUILOD (2022)

A modelagem Fuzzy realizada forneceu os valores e resultados esperados a serem utilizados na aplicação. Em seguida será apresentado as tecnologias utilizadas para implantar este modelo na aplicação web.

3.2. Tecnologias utilizadas

Para aplicação deste projeto, foram utilizadas as linguagens de programação Python e JavaScript, através dos frameworks Flask e React JS. Para persistência de dados, foi utilizado o banco de dados SQLite. E para a transferência de dados, foi utilizado a ferramenta JSON. Para o tratamento de lógica fuzzy do sistema, foi utilizado a biblioteca Python SciKit-Fuzzy. Para a coleta de informações foi utilizado Google Forms. Na sequência, será explicado cada uma das tecnologias utilizadas.

3.2.1. Flask

Flask é um *microframework* de código aberto para desenvolvimento web. Possui funcionalidades simples mas com capacidade de criar aplicações complexas.

De acordo com sua apresentação, disponível no Github da biblioteca, Flask fornece apenas sugestões de código, ainda é muito dependente de outras bibliotecas.

Tem fácil integração com qualquer banco de dados, além de ter modelos prontos de HTML para serem tratados. Porém, para a aplicação deste trabalho, não foram utilizadas as ferramentas de front-end da biblioteca.

Esse *framework* foi escolhido pela sua facilidade de aplicação e funcionalidades simples de criação de APIs.

3.2.2. SciKit-Fuzzy

SciKit-Fuzzy é uma biblioteca de código aberto desenvolvida em Python com o objetivo de auxiliar a comunidade que utiliza Python para meios científicos.

Essa biblioteca fornece funcionalidades para trabalhar com lógica *fuzzy*. As principais delas são: criar antecedentes e consequentes; estabelecer regras de inferências; fazer a fuzzificação e defuzzificação de maneira simples.

A ferramenta foi escolhida como alternativa à linguagem MatLab utilizada na pesquisa base de Monguilod (2022). O MatLab possui uma curva de aprendizagem muito alta. Além disso, após fazer a modelagem dos dados era necessário exportar os arquivos em código C, mas não havia forma de inserir dados via código integrados à aplicação web, por isso, essa ferramenta foi substituída.

3.2.3. React JS

A biblioteca React serve para a criação de interfaces de usuário que utilizam JSX para combinar a linguagem de marcação HTML com o código em JavaScript.

Como as opções de Flask para front-end são limitadas a modelos estáticos, React JS foi escolhido para criar uma interface de usuário mais dinâmica.

Ao utilizar duas linguagens diferentes para o front e o back-end do sistema, o projeto fica melhor dividido e com menos acoplamento. Dessa forma, as interações com o usuário não interferem com o processamento dos dados e vice-versa.

3.2.4. JSON

Como descrito no site oficial, JSON é uma formatação leve de troca de dados. Por isso, é a forma mais fácil, ágil e prática para estabelecer a troca de informações entre o back-end e o front-end.

Como a sintaxe do JSON pode ser facilmente escrita na criação de objetos tanto em JS com objetos e Python com dicionários, sua escolha foi pertinente.

3.2.5. SQLite

A opção de banco de dados utilizada no projeto foi SQLite, um banco de dados embutido na própria aplicação que não precisa de um servidor para rodar.

Como é uma aplicação que não precisa de uma escalabilidade tão grande, um banco de dados integrado atende os requisitos de persistência de dados.

3.3. Validação do questionário

A coleta das respostas foi realizada via Google Forms

O questionário foi enviado para 11 pessoas de diferentes empresas, diferentes cargos e diferentes níveis de experiência. Dessa maneira, foi possível avaliar diversos pontos de vista sobre levantamento de requisitos.

Após a coleta das informações, um relatório foi extraído e os dados foram enviados ao back-end da aplicação para utilizar a lógica *fuzzy*.

4. RESULTADOS

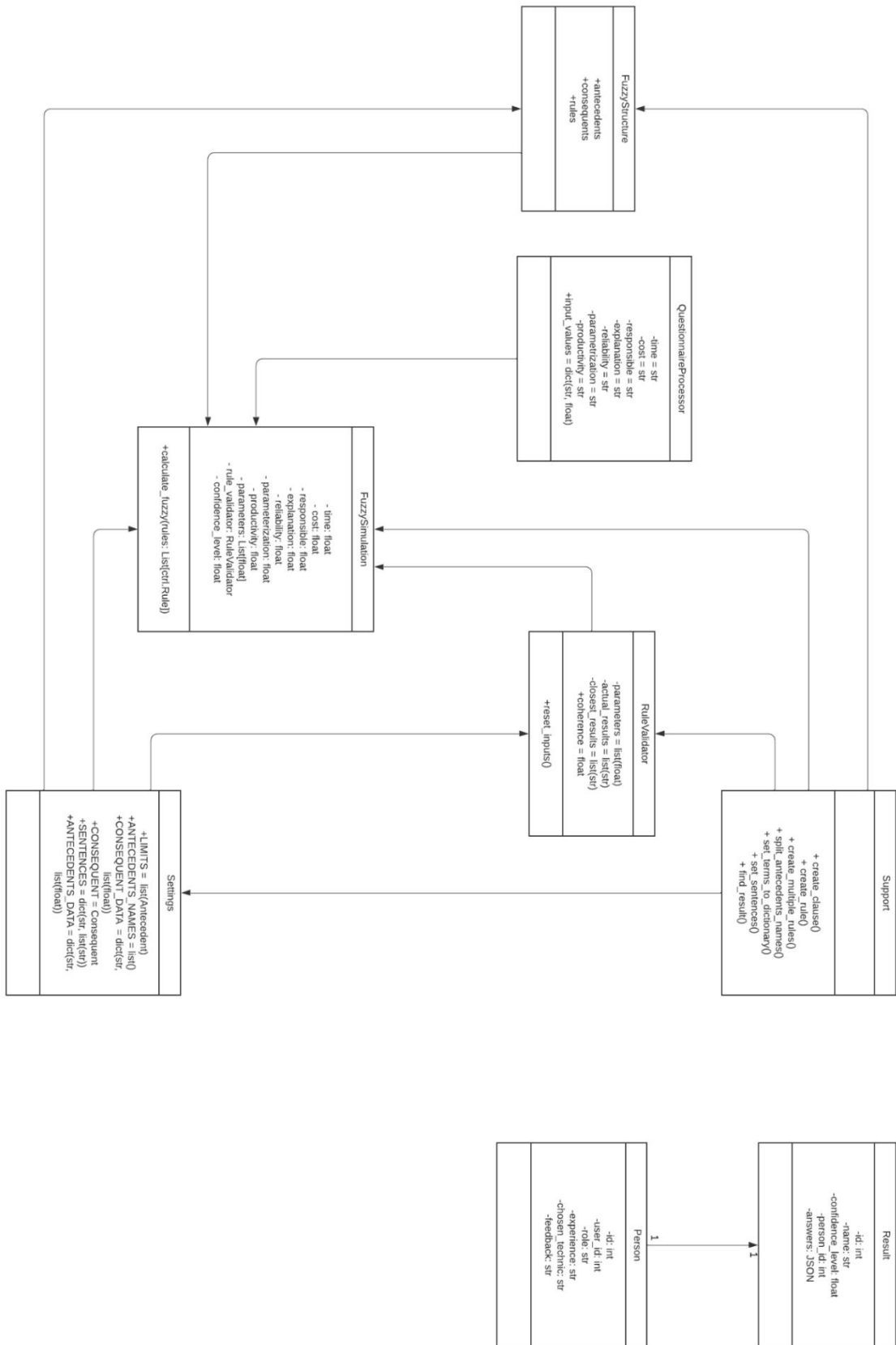
Tendo o embasamento das tecnologias utilizadas e modelo fuzzy construído, este capítulo explicará como foi criado a base de conhecimento e como foi aplicada a lógica fuzzy dentro da aplicação, além dos módulos de apresentação, o questionário construído e a página de resultado. Além disso, mostrará os testes realizados para a validação, as respostas recebidas e a análise dos resultados.

4.1. Aplicação desenvolvida

Por meio das tecnologias citadas na seção anterior, a aplicação foi construída partindo do back-end, com a criação de classes específicas para a manipulação das informações e regras de negócio. Posteriormente, vinculando as informações a modelos armazenados no banco de dados. Então, foram criadas APIs REST para estabelecer comunicação entre o back e o front-end. Por fim, foi desenvolvido a interface do usuário.

A modelagem da aplicação pode ser analisada no diagrama da figura 19.

Figura 19 - Diagrama de classes da aplicação



Fonte: Autoria própria (2023)

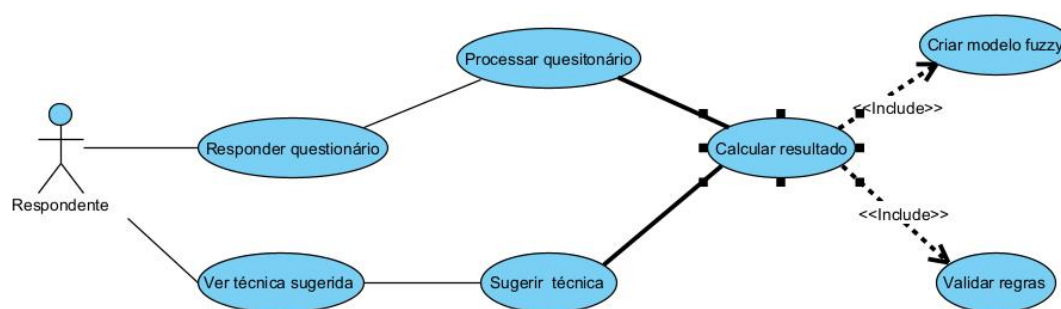
O diagrama de classes apresentado na figura 19 mostra todas as classes envolvidas no processo de aplicação da lógica fuzzy. As classes *Settings* e *Support* fornecem recursos para os demais objetos, e as classes *FuzzySimulation*, *RuleValidator* e *FuzzyStructure* dependem de seus atributos e métodos.

A classe *FuzzyStructure* cria os objetos relacionados à lógica fuzzy, como os antecedentes, o conseqüente e as regras. As regras geradas por esse objeto são utilizadas para calcular o resultado na *FuzzySimulation*, que instancia um objeto *RuleValidator* para validar as regras e retornar um resultado e um valor de confiança. Os dados para serem utilizados no validador vem da classe *QuestionnaireProcessor*.

As classes *Result* e *Person* são relacionadas ao usuário, que recebe um resultado atribuído a ele. Assim, como um resultado só pode pertencer a uma única pessoa.

A partir das classes foi possível utilizar a base de conhecimento para executar os requisitos da aplicação. Para entender melhor como funciona o sistema, foram incrementados um diagrama de casos de uso e um diagrama de sequência, apresentados nas figuras 20 e 21, respectivamente.

Figura 20 - Diagrama de caso de uso da aplicação

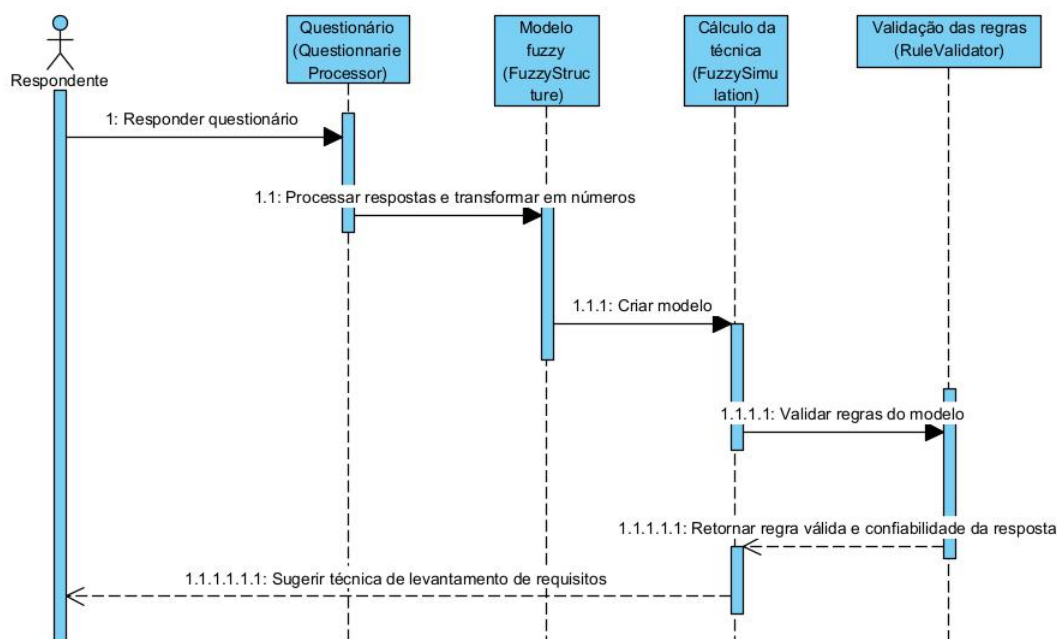


Fonte: Autoria própria (2023)

O diagrama de caso de uso geral apresentado na figura 20 tem o respondente como ator principal. Ele interage diretamente com o caso responder questionário e ver a técnica sugerida. Já o ato de responder o questionário tem como relação o processamento do questionário, que na

seqüência interage com o caso para calcular o resultado, que inclusive tem os casos criar o modelo fuzzy e validar regras diretamente vinculados a ele. Por fim, calcular o resultado interage com a função de sugerir a técnica de levantamento de resultados que leva a técnica sugerida até o respondente.

Figura 21 - Diagrama de seqüência da aplicação



Fonte: Autoria própria (2023)

Para ajudar a entender como acontece o processo dentro da aplicação, foi desenvolvido o diagrama de seqüência representado na figura 21. A primeira fase começa com um ator respondendo o questionário. Em seguida, a entidade *QuestionnaireProcessor* envia a mensagem de processamento das respostas e transformação em números para a entidade *FuzzyStructure*. O modelo Fuzzy envia as informações para a classe de cálculo, *FuzzySimulation* que se comunica com a classe responsável por validar as regras, *RuleValidator* que retorna para a entidade anterior a regra válida a partir das respostas, juntamente com o nível de confiança dessa regra. Por fim, a entidade de cálculo finaliza o processo devolvendo para o respondente a técnica sugerida.

4.2. Criação da base de conhecimento

A primeira etapa foi armazenar os valores estabelecidos pela pesquisa de Monguilod em um arquivo CSV para poder ser manipulado posteriormente pelo código. Também se fez necessário criar um outro arquivo CSV com os dados da tabela informada na figura 10.

Com os dados disponíveis, foi possível utilizar a biblioteca *SciKit-Fuzzy* para criar objetos de três classes internas do pacote: *Antecedent*, responsável pelos antecedentes do modelo, no caso, foram os parâmetros definidos; *Consequent*, para representar o resultado final, no caso, a técnica de levantamento de requisitos; e por último, *Rule*, equivalente a regra de inferência para cada uma das técnicas apresentadas.

Então, com o objetivo de reduzir a sintaxe e tornar o sistema mais prático, primeiro foi lido os arquivos CSV com os dados a serem utilizados e transformados em dicionários com o nome do antecedente como chave, e como valor, um outro dicionário com cada termo como chave e sua lista de valores como valor. Para as possíveis regras, foi definido dicionários com as técnicas como chaves e uma lista com os termos correspondentes a cada antecedente como valor. Após isso, foi criada uma função para criar os antecedentes e consequente e outra para criar as regras automaticamente.

4.3. Aplicação da lógica *fuzzy*

Com base no modelo de Monguilod (2022), usando a base de regras e extraíndo antecedentes, consequentes, a fuzzificação foi realizada. Foram informados valores numéricos para cada uma das variáveis linguísticas estabelecidas, ou seja, para cada um dos parâmetros de análise foi informado um valor de entrada.

Foi criado um objeto da classe interna *ControlSystem* para definir as inferências conforme as regras *fuzzy* criadas.

Por fim, foi criado um objeto da classe interna *ControlSystemSimulation*, que ao receber os valores de entrada, faz a defuzzificação e retorna o valor calculado pelas funções internas que realizam as fórmulas *fuzzy*.

Para poder retornar o termo linguístico correspondente ao valor calculado, foi necessário criar uma função que acessa os valores do dicionário com as técnicas e confere em qual lista o valor defuzzificado pertence. Então, retorna a chave correspondente a lista e assim é definido o resultado linguístico.

4.4. Questionário

O front-end desenvolvido em React JS se trata de um questionário com 10 perguntas correspondentes aos critérios estabelecidos e 3 perguntas sobre o respondente. As respostas do respondente servem para validar a qualidade das respostas.

As perguntas referentes aos parâmetros de análise (variáveis linguísticas fuzzy) foram pensadas de um ponto de vista contrário ao que foi desenvolvido na modelagem usada como base. Na pesquisa inicial, as técnicas já estavam definidas, por consequência, já sabia quais critérios foram atendidos. O questionário segue a linha reversa, a partir das respostas é definido um termo para cada parâmetro e com base na combinação das respostas e nos critérios estabelecidos é definido uma técnica. Por essa razão, o contexto de cada parâmetro precisou ser minuciosamente alterado para ser validado pela lógica *fuzzy*.

a) Tempo

Para o critério tempo a pergunta utilizada foi a seguinte: “Qual o tempo planejado para o desenvolvimento (em meses)?”. A resposta é informada em um campo numérico. Com base nos meses de desenvolvimento de um projeto, entende-se que um período é dedicado ao levantamento de requisitos. Para a lógica fuzzy modelada, um projeto com o tempo programado entre 1 a 4 meses é considerado médio. Acima disso, é considerado o tempo como longo.

b) Custo

O critério custo foi criado com base no custo hora de um desenvolvedor, considerando também o seu nível de experiência, onde quanto maior o nível, mais o valor aumenta. De acordo com Monguilod (2022), a estimativa do custo pode variar conforme a empresa.

A pergunta elaborada para obter o valor desse critério foi “Em relação ao nível de experiência da equipe, como está distribuído?”. A resposta é selecionada de um conjunto de opções: Maioria Sênior; Maioria Plenos; Maioria Juniors; Mais seniores e plenos do que juniors; Mais seniores e juniores do que plenos; Mais plenos e juniores do que seniors; Mesma quantidade em todos os níveis.

Para classificar as respostas foi analisado o seguinte: se uma equipe tem mais integrantes de nível sênior e pleno, então, o custo é alto; se a equipe tem mais integrantes de nível pleno ou a mesma quantidade em todos os níveis, o custo é médio; caso a equipe tenha mais juniors, então o custo de desenvolvimento pode ser considerado baixo.

c) Pessoas responsáveis

Para este critério foi analisado quantas pessoas de uma equipe participam da etapa de levantamento de requisitos, independente da técnica.

Foram elaboradas duas perguntas, uma mais direta, respondida com um valor numérico: “Quantas pessoas são responsáveis pelo levantamento de requisitos?”; e outra com as opções de sim ou não: “É possível apenas uma pessoa fazer o levantamento?”. Contudo, levando em conta que o respondente não sabe qual técnica irá utilizar, é necessário considerar se todas as pessoas envolvidas no projeto são qualificadas para a atividade.

Para obter o valor fuzzy foi necessário combinar as respostas. Caso apenas uma pessoa faça o levantamento de requisitos e não seja necessário mais ninguém, então, “indivíduo” é retornado como valor. Se mais de uma pessoa participa da aplicação e não é possível ser feita por apenas um, então o valor é “grupo”. E por fim, se um grupo aplica, mas é possível apenas uma pessoa aplicar, a resposta será “indivíduo ou grupo”.

d) Explicação

Esse é o critério equivalente a imposição usado na modelagem de Monguilod. Foi trocado o termo porque explicação faz mais sentido quando se trata de demonstrar conhecimentos para entendimento.

Esse parâmetro possui duas perguntas para definir o valor fuzzy. “Quais ferramentas são utilizadas pela equipe para o levantamento de requisitos?”, com as seguintes opções para serem selecionadas: Ferramenta automatizada; Observação; Conversas; Workshops. Essa primeira pergunta leva em conta a opinião do respondente com relação ao tipo de ferramenta ele tem preferência para o levantamento dos requisitos. Cada uma das opções exige um nível de explicação para ser completamente entendido o requisito a ser analisado.

A segunda pergunta é “Quantas horas por semana, em média, o cliente será consultado?”, que serve para analisar o quanto o cliente ou usuário precisará ser consultado para obter todos os requisitos. Como isso não pode ser definido previamente, considera-se uma resposta estimada com base no desempenho que a equipe tem em extrair informações do cliente e também a disponibilidade deste com a equipe.

A combinação de respostas ficou assim: se há preferência por uma ferramenta automatizada ou por observar uma atividade e o cliente será consultado menos de 5 horas na semana, então, a explicação necessária para entendimento é baixa, porque não será necessário interagir com frequência com o cliente e as ferramentas escolhidas informam o necessário precisamente; se a preferência foi por conversar com o cliente e este ter que ser consultado por mais de 5 horas e menos de 16 horas na semana, então a explicação necessária é mediana; caso seja escolhido fazer workshops, vai exigir bastante tempo do cliente, por isso, a explicação é alta porque são utilizados diversos tipos de ferramentas como cards, diagramas, além de ser uma combinação de diversas outras técnicas.

e) Confiabilidade

Esse é o parâmetro para definir a qualidade das informações recebidas, por essa razão, foram definidas duas perguntas.

A primeira questão, “Nível de experiência do aplicador do levantamento de requisitos?”, refere-se ao nível de experiência do aplicador, se ele é sênior, pleno ou junior. Espera-se que alguém mais experiente consiga aplicar técnicas de maneira mais produtiva.

Já a segunda questão, “O quanto você confia na fonte de informações?”, refere-se a confiança do aplicador em relação ao cliente do projeto, se este está comprometido e disponível para passar informações precisas e coerentes. As alternativas para essa questão são: Muita confiança; Confio parcialmente; Não confio.

A combinação de respostas ficou assim: se o aplicador é sênior e há muita confiança ou confiança parcial na fonte de informações, o nível de confiabilidade é alto; a confiabilidade também é alta caso o aplicador seja de experiência plena e a fonte de informações seja confiável; para um aplicador sênior e uma fonte de informações não confiável, a confiabilidade ainda é média, porque alguém experiente consegue interpretar e obter informações relevantes mesmo se a fonte não é muito confiável; a confiabilidade também será média em caso o aplicador for pleno e a fonte parcialmente confiável; e no caso do aplicador ser júnior, mas a fonte de informações ser clara e coerente; a confiabilidade será baixa quando o aplicador for de nível pleno e a fonte de informação não for de confiança; ou nos casos que o aplicador for junior e a fonte não for de confiança ou confiança parcial.

f) Padronização

O parâmetro padronização é bastante subjetivo, já que o objetivo é calcular qual a técnica de levantamento de requisitos, por essa razão, foi levado em conta a opinião do respondente sobre métodos dinâmicos ou padronizados, sendo essas as opções.

Para validar a resposta, considera-se método dinâmico para projetos que permite um levantamento de requisitos de maneira mais subjetiva e discutível entre os participantes, retornando que não há padronização. Métodos padronizados seriam os que precisam de respostas mais objetivas e precisas.

g) Produtividade

Parâmetro para definir a produtividade que a equipe tem em desenvolver o projeto no tempo esperado com base no levantamento de requisitos e também na própria etapa de elicitação, referente a produtividade que a equipe ou o aplicador tem de extrair informações úteis.

É uma pergunta bem direta, “Qual o nível de produtividade da equipe para atender os requisitos no prazo estabelecido?”, sendo as opções já correspondentes aos termos utilizados para esse parâmetro, baixa, média e alta.

4.5. Processamento das respostas

As respostas são enviadas pelo front-end em um texto JSON e recebidas no back-end por uma requisição HTTP do tipo POST.

Após recebimento das respostas, é realizada a criação de uma instância da classe *QuestionnaireProcessor*, que realiza o processamento de cada uma das regras em funções individuais.

Em seguida, é realizada a criação de uma instância da classe modelada *FuzzyStructure*, que cria os antecedentes, consequentes e regras. Depois é instanciado a classe modelada *FuzzySimulation* que recebe os valores de entrada das respostas, já definidos pela classe processadora do questionário. A instância de *FuzzySimulation* chama a função *calculate_fuzzy* que recebe as regras da estrutura como parâmetro para fazer a fuzzificação, depois a defuzzificação e retornar a técnica definida.

O maior empecilho desse processamento é a possibilidade do usuário responder o questionário e as respostas retornarem valores que não sejam contemplados em nenhuma regra *fuzzy*. Caso isso acontecesse, a aplicação

retornaria um erro. Para mitigar a possibilidade de retornar um erro ao informar certas combinações de respostas, foi estabelecido um nível de confiança para a técnica retornada. A confiança é um valor descrito em porcentagem para quantificar o pertencimento ao conjunto. Na aplicação, quanto mais parâmetros forem utilizados no processamento da resposta, maior será o nível de confiança.

A quantidade de parâmetros é sete ao todo, e a possibilidade de técnicas possíveis são dez, porém, observação e etnografia possuem a mesma regra de inferência, então são nove, portanto, a quantidade de respostas possíveis seria altíssima e inviável de configurar. Mas não se faz necessário utilizar todas as combinações possíveis porque os resultados da pesquisa de Monguilod já definiram os critérios a serem atendidos. Por essa razão, foi programada uma medida de contingência para o possível erro.

Caso a defuzzificação retorne um valor válido, a aplicação consegue informar uma técnica normalmente, com nível de confiança de 100%. Para o caso de falhar, é feita uma alteração nos valores numéricos de entrada informados. Para cada um dos parâmetros, é validado qual termo *fuzzy* antecedente será retornado pelo valor de entrada. Após validar todos os termos, é validado quais termos equivalem a alguma regra na tabela de regras, essa validação retorna a regra com o maior número de termos encontrados. Por exemplo, a regra criada utiliza os parâmetros tempo, custo e produtividade, o conjunto esperado é [tempo médio, custo baixo, e produtividade alta], contudo, a resposta do usuário acabou retornando o conjunto [tempo alto, custo baixo e produtividade média] e esse conjunto não existe na lista de regras criadas. A partir desse conjunto de termos, é modificado o valor numérico dos critérios que não foram atendidos para poder terminar a defuzzificação com uma inferência válida, ou seja, as variáveis tempo e produtividade teriam seus valores numéricos mudados para o conjunto retornado ser o conjunto que está mapeado. Sendo assim, o nível de confiança é reduzido de acordo com a quantidade de critérios não atendidos precisamente, portanto, como o conjunto modificado teve apenas 1 termo válido, o nível de confiança diminui para 33,33%. O cálculo da

confiança atribui o mesmo valor de importância para cada termo, então a divisão é sempre feita pelo número de termos válidos pelo total.

A resposta do usuário no questionário não é alterada, apenas desconsiderada. Nesses casos, são utilizados apenas os valores válidos para retornar a técnica com um nível de confiança mais baixo.

4.6. Interface do usuário

A aplicação foi desenvolvida em dois projetos diferentes, separando o back-end do front-end.

O front-end contém todas as interações com o usuário, onde é coletado suas informações e respostas sobre um projeto qualquer escolhido por ele para poder definir a técnica de levantamento de requisitos que poderá ser utilizada.

Após respondido o questionário, o usuário é direcionado para uma página de relatório sintetizado onde ele pode visualizar o resultado retornado pelo sistema, seu nível de confiança e as respostas enviadas por ele. Também foi incluído um campo para o usuário deixar um comentário com a sua opinião sobre o resultado.

4.6.1. Apresentação

A primeira tela é a apresentação do questionário com o título do projeto e um texto explicativo sobre cada parâmetro (Figura 22).

Essa página contém uma breve saudação para o usuário. Em seguida, é apresentado um resumo sobre o objetivo geral do questionário com a explicação de cada parâmetro utilizado para a definição da técnica.

Após explicar os parâmetros, foi informado a maneira como serão utilizadas as informações respondidas. O texto é finalizado por uma solicitação para o usuário ser honesto em suas respostas e uma saudação de agradecimento.

Ao final da página, foi colocado um botão para iniciar o questionário. Esse componente tem a funcionalidade de direcionar o usuário para a primeira parte do formulário.

Figura 22 - Tela de apresentação

Bem-vindo ao Nosso Questionário sobre Técnicas de Levantamento de Requisitos

Este questionário tem o intuito de ajudar na escolha da melhor técnica de levantamento de requisitos para o seu projeto. As perguntas abordarão os seguintes aspectos:

- Tempo do Projeto:** Avaliaremos o tempo estimado para o desenvolvimento;
- Custo:** Analisaremos quanto custa a equipe de desenvolvimento;
- Equipe:** Verificaremos a quantidade de pessoas necessárias para levantar os requisitos;
- Complexidade:** Mediremos o quanto é preciso explicar sobre o projeto para que ele seja compreendido;
- Confiabilidade:** Levar em conta o nível de experiência do aplicante e a qualidade da fonte de informações;
- Padronização:** Consideraremos a necessidade de padronização da técnica utilizada;
- Produtividade:** Analisaremos o impacto no nível de produtividade da equipe na implementação dos requisitos levantados.

As respostas serão utilizadas em uma aplicação que utiliza lógica fuzzy, método de aprendizado de máquina destinado a estabelecer uma técnica de levantamento com base nas combinações de respostas com um determinado grau de confiança.

Por favor, preencha o questionário com atenção e honestidade. Agradecemos por sua colaboração!

[Iniciar Questionário](#)

Fonte: Autoria própria (2023)

4.6.2. Questionário sobre o respondente

A página Questionário sobre o respondente funciona como uma página de login para poder registrar o usuário no sistema e vincular as respostas e o resultado da técnica definida a ele. Caso ele não se registre, não será possível acessar a página com as perguntas sobre o processo, conseqüentemente, não será possível acessar a página de resultados. Se ele tentar acessar a página do questionário através da URL do navegador, ele é redirecionado para a tela de apresentação.

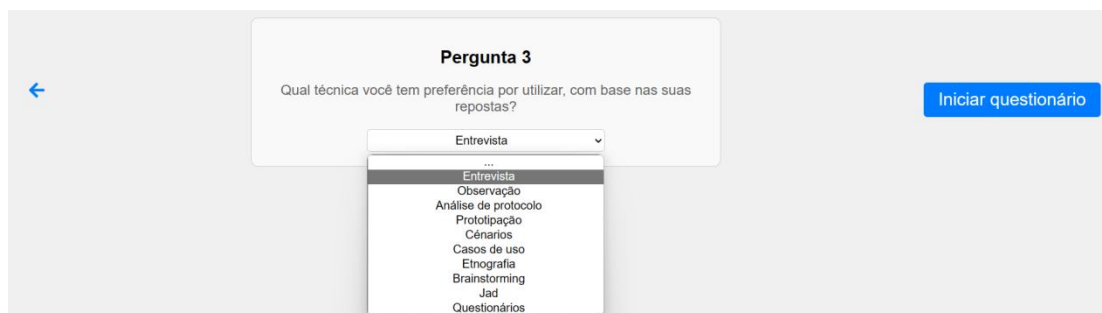
Foram definidas três perguntas sobre o respondente: “Qual o seu cargo?”; “Qual o seu nível de experiência?”, sendo este um item de seleção com as opções sênior, pleno e junior; e ao final do questionário tem a última pergunta, “Qual técnica você tem preferência por utilizar em seus projetos?” (Figura 23), que serve para comparar, posteriormente, a análise do respondente sobre a técnica que deve ser usada no projeto e qual a técnica que será retornada pela aplicação.

Todas as perguntas são carregadas do banco de dados ao acessar a página, entretanto, foi programado para aparecer cada pergunta isoladamente. É possível acessar a pergunta seguinte ou retornar para a

pergunta anterior (caso não seja a primeira) por meio de botões com formato de setas.

Ao chegar na última pergunta sobre o respondente (figura 23), a seta de avanço é substituída por um botão para iniciar o questionário sobre o projeto.

Figura 23 - Pergunta final do questionário sobre o respondente



←

Pergunta 3

Qual técnica você tem preferência por utilizar, com base nas suas repostas?

Entrevista

- ...
- Entrevista
- Observação
- Análise de protocolo
- Prototipação
- Cenários
- Casos de uso
- Etnografia
- Brainstorming
- Jad
- Questionários

Iniciar questionário

Fonte: Autoria própria (2023)

4.6.3. Questionário sobre o projeto

O questionário sobre o projeto é acessível apenas depois do usuário ser registrado e direcionado para a página com as perguntas sobre o projeto. Essa página foi construída dinamicamente, da mesma forma que o questionário sobre o respondente. As perguntas são carregadas do banco de dados e exibidas separadamente, sendo acessíveis por meio de setas de avanço ou retorno. Ao chegar na última pergunta, a seta é substituída por um botão para finalizar o questionário e enviar a resposta (Figura 24).

As perguntas são respondidas em dois tipos de campos, dependendo da pergunta, numérico (figura 25) e seleção de opções (figura 24). Essa funcionalidade foi inserida para limitar as respostas do usuário e retornar valores objetivos e já esperados pela aplicação.

Figura 24 - Pergunta com campo de seleção e botão de finalizar

A captura de tela mostra uma interface de usuário para uma pergunta. No topo, há um ícone de seta para a esquerda. O título da pergunta é "Pergunta 10". O texto da pergunta é "Qual o nível de produtividade da equipe para atender os requisitos no prazo estabelecido?". Abaixo do texto, há um menu suspenso com o valor "Média" selecionado. O menu está aberto, mostrando as opções "Alta", "Média" e "Baixa". À direita da pergunta, há um botão azul com o texto "Finalizar questionário".

Fonte: Autoria própria (2023)

Figura 25 - Pergunta com campo numérico

A captura de tela mostra uma interface de usuário para uma pergunta. No topo, há um ícone de seta para a esquerda. O título da pergunta é "Pergunta 1". O texto da pergunta é "Qual o tempo planejado para o desenvolvimento (em meses)?". Abaixo do texto, há um campo de entrada numérica com um ícone de seta para cima e para baixo. À direita da pergunta, há um ícone de seta para a direita.

Fonte: Autoria própria (2023)

4.6.4. Saídas da Aplicação

A última página do sistema só pode ser acessada após o usuário responder todas as perguntas. Essa página contém um título de identificação do assunto, seguido da técnica calculada pelo sistema com base nas respostas e logo abaixo, contém o nível de confiança do resultado medido pela aplicação, que pode variar de 54 a 100%, com base na quantidade de respostas válidas (figura 26).

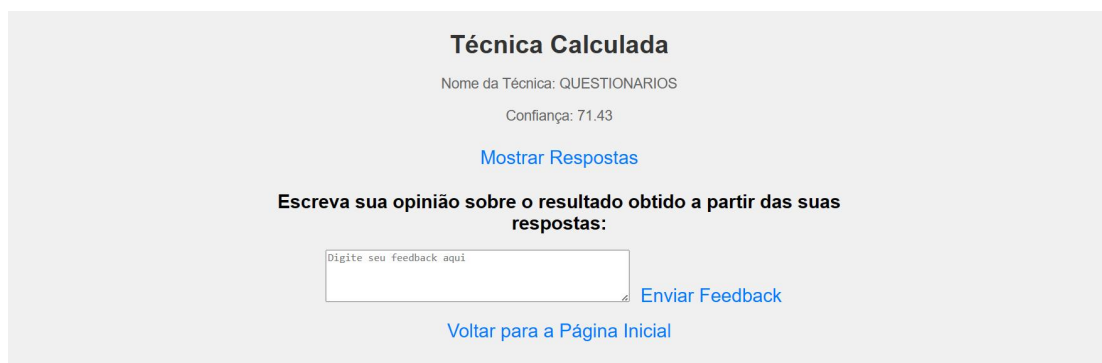
Abaixo do resultado, existe um texto em destaque para exibir as respostas informadas, caso o usuário queira revisar suas respostas. Caso seja clicado novamente, as respostas são recolhidas e não são mais exibidas.

As outras funcionalidades presentes nesta página são o envio de um comentário e o retorno para a página de apresentação. O usuário pode colocar comentários com a sua opinião sobre o resultado e enviar no botão indicado. Esses comentários são armazenados no banco de dados e não são devolvidos para o usuário.

Por fim, existe o botão indicado para retornar à tela inicial de apresentação. Ao ser redirecionado, o usuário automaticamente faz logoff do

sistema e caso deseje responder novamente, será contabilizado como um novo usuário.

Figura 26 - Tela de resultado final



The screenshot shows a web interface with a light gray background. At the top, the title "Técnica Calculada" is centered. Below it, the text "Nome da Técnica: QUESTIONARIOS" and "Confiança: 71.43" are displayed. A blue link "Mostrar Respostas" is positioned below the confidence score. The main instruction reads "Escreva sua opinião sobre o resultado obtido a partir das suas respostas:". Below this is a text input field with the placeholder text "Digite seu feedback aqui". To the right of the input field is a blue link "Enviar Feedback". At the bottom, another blue link "Voltar para a Página Inicial" is centered.

Fonte: Autoria própria (2023)

Isso encerra a explicação sobre o front-end da aplicação desenvolvida, a próxima seção apresentará os resultados obtidos com o questionário enviado.

4.7. Validação da aplicação (Testes)

A coleta das respostas foi realizada via Google Forms.

O questionário foi enviado para 11 pessoas de diferentes empresas, diferentes cargos e diferentes níveis de experiência. Dessa maneira, foi possível avaliar diversos pontos de vista sobre levantamento de requisitos.

Após a coleta das informações, um relatório foi extraído e os dados foram enviados ao back-end da aplicação para utilizar a lógica *fuzzy*.

Nesta seção será contemplado as respostas de usuários reais, obtidas através do formulário disponibilizado pelo Google Forms. As respostas de cada respondente foram inseridas na aplicação para obter um resultado calculado sobre a técnica de levantamento de requisitos e comparada com as técnicas informadas pelo usuário. Foram obtidas 11 respostas úteis para o projeto. Contudo, é uma amostra pequena, e seus resultados não são tão precisos porque a maior parte dos respondentes tem um nível de experiência baixo em relação a levantamento de requisitos, sendo exatamente 6 profissionais juniores, 4 profissionais plenos e apenas 1 profissional de nível sênior.

As perguntas são exatamente as mesmas presentes na aplicação, porém, as respostas permitiam mais de uma resposta em alguns casos específicos de seleção, além de perguntas abertas. Mas para aplicar no sistema *fuzzy* desenvolvido, foi considerado apenas uma das respostas. Foram feitas mais de uma validação para caso o respondente respondesse mais de uma opção na pergunta sobre quais ferramentas

Após a validação das respostas, foi comparado a precisão da técnica calculada com as técnicas escolhidas pelo respondente. Para julgar o resultado da aplicação eficiente, foi considerado se as técnicas retornadas pela aplicação estão presentes na lista de técnicas informadas pelo respondente. As respostas e resultados estão apresentados no quadro 2 e 3.

As linhas em verde da tabela, representam os casos em que a técnica sugerida pela aplicação foi eficiente.

As linhas apresentadas em laranja, correspondem a casos parcialmente eficientes, significa que pelo menos uma sugestão foi coerente com as respostas do respondente.

Por fim, as linhas em vermelho são os casos que a sugestão se demonstrou ineficiente, por não aparecer na resposta do usuário.

Apesar do número de respostas ineficientes ser maior que o número de respostas eficientes, alguns pontos precisam ser elencados. O nível de experiência do respondente pode interferir no resultado final, porque sua análise pode ser equivocada sobre as técnicas pertinentes ao projeto. A falta de informações também pode ser um empecilho, como foi o caso da linha 4 do primeiro quadro, em que o usuário não respondeu a pergunta sobre a quantidade de horas consultadas com o cliente, portanto, foi considerado o valor 0, que acabou levando a diminuição do grau de pertinência. A mesma coisa aconteceu na linha 8 do primeiro quadro, onde nas perguntas relacionadas ao tempo e a explicação, o respondente informou que pode ser variado conforme o projeto, portanto, também foi utilizado 0 em ambos os termos.

Outro fator a ser considerado é o contexto do projeto. Um projeto específico pode demandar uma técnica que não foi considerada pelo analista mas foi sugerida pela aplicação.

Em resumo, os testes atestam que aplicação pode ser usada, mas ainda se faz necessário uma análise humana para garantir a eficiência.

Quadro 2 - Quadro com os resultados obtidos 1

Respondente	Técnica escolhida pelo respondente	Tempo	Custo	Pessoas responsáveis		Explicação/Complexidade		Confibilidade		Padronização	Produtividade	Técnica retomada pela aplicação
				Quantas pessoas são responsáveis pelo levantamento de requisitos?	E possível apenas uma pessoa fazer o levantamento?	Quais ferramentas são utilizadas pela equipe para o levantamento de requisitos?	Quantas horas por semana, em média, o cliente será consultado?	Nível de experiência do aplicador do levantamento de requisitos?	O quanto você confia na fonte de informações?			
Assistente de Processos Júnior	Observação, Entrevista, Prototipação, Casos de uso, Brainstorming, Questionários	1	Maioria Juniors	3	Sim	Observação, Conversas	1	Júnior	Confio parcialmente	Métodos dinâmicos	Média	Questionários e entrevistas (57,14%)
QA Júnior	Observação, Casos de uso	8	Maioria Juniors	3	Não	Observação, Conversas	10	Pleno	Confio parcialmente	Métodos dinâmicos	Alta	Prototipação (57,14%)
Analista de Sistemas Pleno	Prototipação, Casos de uso	1	Mais plenos e Juniors do que seniores	2	Sim	Gillab	2	Pleno	Confio parcialmente	Métodos padronizados	Alta	Centrais (57,14%)
Programador Júnior	Observação, Prototipação	6	Maioria Plenos	2	Não	Observação, Conversas		Sênior	Muita confiança	Métodos padronizados	Média	JAD (57,14%) e Brainstorming (71,43%)
Product Owner Júnior	Entrevista, Análise de protocolo, Prototipação	3	Mais plenos e Juniors do que seniores	1	Sim	Observação, Conversas	8	Pleno	Confio parcialmente	Métodos dinâmicos	Média	Entrevista (71,43%) e Questionários (85,71%)
Lider de Desenvolvimento Back End Pleno	Entrevista, Análise de protocolo, Casos de uso, Questionários	2-3	Mais seniores e plenos do que Juniors	1	Sim	Ferramenta automatizada, Conversas	3	Pleno	Muita confiança	Métodos dinâmicos	Média	Entrevista (71,43%) e Questionários (85,71%)
Gerente de Projetos Júnior	Observação, Casos de uso	3	Mais plenos e Juniors do que seniores	2	Sim	Observação, Conversas, Workshops	2	Júnior	Confio parcialmente	Métodos dinâmicos	Média	Questionários e entrevista (57,14%)
Analista de Processos/Desenvolvedor RPA Pleno	Análise de protocolo, Casos de uso, Brainstorming	Varia conforme tamanho do Projeto	Maioria Plenos	2	Sim	Ferramenta automatizada, Observação, Conversas, Video	Varia conforme foi entregue o levantamento	Pleno	Muita confiança	Métodos dinâmicos	Alta	Brainstorming (71,43%) e JAD (57,43%)

Fonte: Autoria própria (2023)

Quadro 3 - Quadro com os resultados obtidos 2

Respondente	Técnica escolhida pelo respondente	Tempo	Custo	Pessoas responsáveis		Explicação/Complexidade	Confiabilidade		Padronização	Produtividade	Técnica retornada pela aplicação	
				Quantas pessoas são responsáveis pelo levantamento de requisitos?	É possível apenas uma pessoa fazer o levantamento?		Explicação/Complexidade	Confiabilidade				
Cargo e nível de experiência	Qual técnica você tem preferência por utilizar em seus projetos?	Qual o tempo planejado para o desenvolvimento (em meses)?	Em relação ao nível de experiência da equipe, como está distribuído?	Quantas pessoas são responsáveis pelo levantamento de requisitos?	É possível apenas uma pessoa fazer o levantamento?	Quais ferramentas são utilizadas pela equipe para o levantamento de requisitos?	Quantas horas por semana, em média, o cliente será consultado?	Nível de experiência do aplicador do levantamento de requisitos?	O quanto você confia na fonte de informações?	Tem preferência por levantamento de requisitos padronizados e com respostas objetivas ou aceita trabalhar com métodos mais dinâmicos e com respostas mais elaboradas?	Qual o nível de produtividade da equipe para atender os requisitos no prazo estabelecido?	Resultado obtido pela aplicação e nível de confiança
Analista de automação Pleno	Observação, Prototipação, Canários, Casos de uso, Questionários	1	Maioria Plenos	2	Sim	Observação, Conversas, Vídeos tutoriais com a execução do projeto	1	Pleno	Muita confiança	Métodos padronizados	Média	Questionários e prototipação (57,14%)
Dev Junior	Observação, Entrevistas, Casos de uso, Questionários	8	Maioria Juniors	2	Sim	Observação, Conversas	2	Pleno	Muita confiança	Métodos dinâmicos	Alta	Brainstorming (71,43%) e JAD (57,14%)
Tech Lead Mobile Sênior	Prototipação, Brainstorming, Refinamentos técnico e de negócios	6	Maioria sênior	3	Não	Observação, Conversas, Workshops	3	Sênior	Confio parcialmente	Métodos dinâmicos	Alta	JAD (57,14%) e brainstorming (71,45%)

Fonte: Autoria própria (2023)

Como é possível observar nos quadros 2 e 3, a maior parte dos resultados obtidos com as respostas foram negativos ou parcialmente válidos, portanto, uma reavaliação desse protótipo pode ser realizada para atestar maior eficiência nos resultados. O ideal é obter respostas de profissionais mais diversificados em experiência, além disso, para o resultado final ser satisfatório e ser possível garantir que a aplicação, ao ser utilizada por um gerente de projetos, retorne um resultado adequado, será necessário que mais de 50% dos testes retornem positivos.

5. CONCLUSÃO

Este trabalho visou mostrar o desenvolvimento de uma aplicação web que aplica um questionário cujo as respostas são utilizadas em um sistema de lógica *fuzzy* usado para definir a técnica de levantamento de requisitos adequada para o cenário respondido nas questões.

Foram utilizadas tecnologias atuais para o desenvolvimento e uma estrutura de projeto onde o back-end desenvolvido com Python Flask ficou dividido da front-end desenvolvido em React JS.

Foi possível utilizar a lógica fuzzy dentro do projeto e obter resultados parcialmente eficientes para cada caso, portanto, para gerentes de projeto poderem utilizar de forma mais precisa, será necessário uma análise humana para cada caso separadamente para garantir a eficiência.

Sugere-se como trabalhos futuros a revisão da etapa de processamento dos valores fuzzy para melhorar o processo de definição da confiança da sugestão, atribuindo valores de importância para cada termo para obter valores mais precisos; a expansão do projeto com a inclusão de novas metodologias de elicitação de requisitos e novos parâmetros que ajudem a definir com mais precisão as técnicas; rever as variáveis linguísticas com especialistas da área de levantamento de requisitos e em sequência fazer uma nova pesquisa para validar os resultados em busca de uma amostra maior, mais diversificada e mais eficiente; bem como a transformação do código atual em uma biblioteca generalista para ser aplicado a estrutura fuzzy proposta em outros contextos.

6. REFERÊNCIAS

- AGUADO, A. G., & CANTANHEDE, M. A. **Lógica fuzzy**. Artigo sem. 2010.
- ANDRADE, M; JAQUES, M. A. P. **Estudo comparativo de controladores de Mamdani e Sugeno para controle de tráfego em interseções isoladas**. Transportes, v. 16, n. 2, 2008.
- BELGAMO, A., & MARTINS, L. E. G. **Estudo Comparativo sobre as Técnicas de Elicitação de Requisitos do Software**. Universidade Metodista de Piracicaba-Unimep, Piracicaba. 2008.
- CARVALHO, F. P. **Técnicas de Levantamento de Requisitos**. 2009.
- CHIEN, F. **Estudo comparativo de técnicas para levantamento de requisitos de aplicativos móveis**. 2018.
- CREMASCO, C. P.; GABRIEL FILHO, L. R. A; CATANEO, A. **Metodologia de determinação de funções de pertinência de controladores fuzzy para a avaliação energética de empresas de avicultura de postura**. Energia na agricultura, v. 25, n. 1, p. 21-39, 2010.
- FIGUEIRA, A. M. D. S. **Análise das técnicas de levantamento de requisitos para desenvolvimento de software nas empresas de Vitória da Conquista-BA**. [S.l.], 2012.
- GALDINO, P. R. **Análise da aplicação de Design Thinking para o levantamento de requisitos: estudos de casos de projetos de sistemas de saúde**. 2021.
- MACHADO, F. N. R. **Análise e Gestão de Requisitos de Software: Onde nascem os sistemas**. 3. ed. São Paulo: Érica, 2018.
- MARRO, A. A., Souza, A. D. C., CAVALCANTE, E. D. S., Bezerra, G. S., & NUNES, R. O. **Lógica fuzzy: conceitos e aplicações**. Natal: Universidade Federal do Rio Grande do Norte (UFRN), 2. 2010.
- MONGUILOD, L. M. **Modelagem fuzzy para identificação de métodos de levantamento de requisitos**. Universidade Tecnológica Federal do Paraná, Câmpus Santa Helena, 2022.
- SELLITTO, M. A. **Inteligência artificial: uma aplicação em uma indústria de processo contínuo**. Gestão & Produção, v. 9, p. 363-376, 2002.
- SILVA, A. L. . **Um estudo sobre sistemas baseados em regras fuzzy**. 2020.
- SILVA, F. G. C. D. **Levantamento de requisitos aplicado à pesquisa e desenvolvimento de produtos de software**. 2023.
- SOMMERVILLE, I. **Engenharia de Software**. Ian Sommerville; tradução Ivan Bosnic e Kalinka G. de O. Gonçalves; revisão técnica Kechi Hiramã. — 9. ed. — São Paulo: Pearson Prentice Hall, 2011.
- SOUZA, A., & PORCILE, G. **Aplicação da lógica fuzzy em processos de decisão econômica**. In IDEAS (pp. 1-20). 2009.
- TANSCHKEIT, R. **Sistemas fuzzy**. Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 338-353. 2004

TIAGO, G. M., BARONI, M. P. M. A., & DA FONSECA, R. F. **Avaliação discente: uma proposta utilizando a Lógica Fuzzy**. Revista eletrônica de educação matemática, 9(2), 87-109. 2014.

TRIGO, R. A.; BARRETO, L. C. **Evolução dos métodos de desenvolvimento de software em microempresas**. Revista Gestão em Foco, Amparo, SP, n.11, p. 114-122, 2019.