



Automação de atribuição de tarefas

NELSON GONÇALVES PINTO LOPES

Junho de 2022

Automação de atribuição de tarefas

Nelson Gonçalves Pinto Lopes

**Dissertação para obtenção do Grau de Mestre em
Engenharia Informática, Área de Especialização em
Sistemas Gráficos e Multimédia**

Orientador: João Paulo Pereira

Júri:

Presidente:

Vogais:

Porto, Junho 2022

Resumo

Com o crescer da população, existe um aumento de potenciais clientes para uma dada indústria. Isto pode representar um problema se a referida indústria não tiver capacidade para responder aos aumentos de consumidores. Para combater esta situação as indústrias recorrem à automação, no entanto existem tarefas que necessitam de intervenção humana. Normalmente estas tarefas são distribuídas pelos operários de forma manual. Este processo de alocação de tarefas a operários pode ser problemático, pois pode consumir bastante tempo, pode atribuir tarefas urgentes a operários com menos capacidade e podem existir tarefas por atribuir durante períodos indeterminados.

Para responder a este problema, é criada uma solução onde é analisado o problema e os respetivos requisitos. É depois desenhada uma solução e criados quatro algoritmos capazes de resolver este problema, dado o mesmo ser não determinístico. Os algoritmos são compostos por três partes. A primeira obtém as informações necessárias. A segunda aplica um algoritmo de decisão multi-critério (MAUT, PROMETHEE II, TOPSIS ou VIKOR) para ordenar as tarefas e os operários pelos respetivos critérios. E a última parte atribui as tarefas aos operários. Os resultados dos quatro algoritmos são depois comparados através do uso de um questionário que indica a preferência de um algoritmo sobre os restantes.

Palavras-chave: MCDM, MAUT, PROMETHEE II, TOPSIS, VIKOR

Abstract

With the growing of the population, there is an increase of potential buyers of a certain industry. This can represent a problem if the industry doesn't have the capabilities to respond to the increase of consumers. To fight this situation the industries resort to automation, however there are tasks that need human intervention. Usually, these tasks are manually distributed between workers. This process of allocating tasks can be problematic, since it can consume a lot of time, give urgent tasks to workers with lesser capabilities, and it can remain unallocated for indeterminate periods of time.

To resolve this problem, it was created a solution where the problem is analysed with its respective requirements. Then the solution is designed and four algorithms capable of solving this problem are created, due to the problem being not deterministic. The algorithms are composed of three parts. The first obtains the necessary information. The second applies a multi-criteria decision making algorithm (MAUT, PROMETHEE II, TOPSIS or VIKOR) to order the tasks and the workers by their respective criteria. And the last part gives the task to the workers. The results of the four algorithms are then compared through the usage of a questionnaire which indicates the preference of one algorithm over the rest.

Keywords: MCDM, MAUT, PROMETHEE II, TOPSIS, VIKOR

Índice

1	Introdução.....	1
1.1	Enquadramento/Contexto.....	1
1.2	Descrição do problema.....	4
1.2.1	Objetivos.....	5
1.2.2	Abordagem.....	6
1.2.3	Avaliação de resultados	7
1.2.4	Contributos	7
1.3	Estrutura do relatório	7
2	Estado da Arte.....	9
2.1	Trabalhos relacionados.....	9
2.2	Tecnologias existentes.....	12
2.3	Conclusão.....	14
3	Análise de valor.....	17
3.1	Processo de inovação.....	17
3.2	New Concept Development.....	18
3.2.1	Elementos do NCD.....	19
3.3	Valor.....	22
3.3.1	Valor percecionado para o cliente.....	22
3.4	Proposta de valor.....	22
3.5	Diagrama FAST.....	23
3.6	Conclusão.....	24
4	Análise e desenho da solução	25
4.1	Domínio do problema.....	25
4.2	Requisitos funcionais e não funcionais.....	28
4.2.1	Requisitos funcionais.....	28
4.2.2	Requisitos não funcionais.....	33
4.3	Arquitetura de Software.....	34

4.4 Modelo de Dados.....	36
4.5 Desenho.....	37
4.5.1 UC1 – Criar Operário.....	38
4.5.2 UC2 – Adicionar Critérios ao Operário.....	39
4.5.3 UC3 – Obter Operários.....	40
4.5.4 UC4 – Criar Tarefa.....	41
4.5.5 UC5 – Adicionar Critérios à Tarefa.....	42
4.5.6 UC6 – Obter Tarefas.....	43
4.5.7 UC7 – Atribuir Tarefas aos Operários.....	44
4.6 Conclusão.....	46
5 Implementação da solução	47
5.1 UC1 – Criar Operário.....	48
5.2 UC2 – Adicionar Critérios ao Operário.....	48
5.3 UC3 – Obter Operários.....	48
5.4 UC4 – Criar Tarefa.....	48
5.5 UC5 – Adicionar Critérios à Tarefa.....	49
5.6 UC6 – Obter Tarefas.....	49
5.7 UC7 – Atribuir Tarefas aos Operários.....	49
5.7.1 Primeira parte da solução do Algoritmo.....	50
5.7.2 Segunda parte da solução do Algoritmo.....	53
5.7.3 Terceira parte da solução do Algoritmo.....	72
5.8 Testes.....	74
5.9 Avaliação da solução.....	75
5.9.1 Resultados do questionário.....	75
5.9.2 Análise dos resultados obtidos	76
5.10 Conclusão.....	76
6 Conclusão	77
6.1 Objetivos Concretizados e Resultados obtidos.....	77
6.2 Limitações e trabalho futuro.....	78
6.3 Apreciação final.....	79

Lista de Figuras

Figura 1: Relação de tipos de automação com a variedade e quantidade de produção.....	4
Figura 2: Processo de Inovação (A.Koen, et al., 2002).....	18
Figura 3: – Modelo NCD (A.Koen, et al., 2002).....	19
Figura 4: Modelo de Osterwalder para o projeto.....	23
Figura 5: Modelo de FAST do projeto.....	24
Figura 6: Modelo de domínio.....	26
Figura 7: Diagrama de casos de uso.....	29
Figura 8: Vista de implementação com granularidade de aplicação.....	35
Figura 9: Vista de lógica com granularidade de aplicação.....	36
Figura 10: Modelo de dados.....	36
Figura 11: Diagrama de sequência do UC1.....	38
Figura 12: Diagrama de sequência do UC2.....	39
Figura 13: Diagrama de sequência do UC3.....	40
Figura 14: Diagrama de sequência do UC4.....	41
Figura 15: Diagrama de sequência do UC5.....	42
Figura 16: Diagrama de sequência do UC6.....	43
Figura 17: Diagrama de sequência do UC7.....	45

Lista de Tabelas

Tabela 1: Distribuição automática de tarefas em produtos comerciais.....	12
Tabela 2: Tarefas geradas.....	51
Tabela 3: Operários gerados.....	52
Tabela 4: Tarefas normalizadas.....	54
Tabela 5: Operários normalizados.....	55
Tabela 6: Tarefas normalizadas com os pesos aplicados.....	55
Tabela 7: Operários normalizados com os pesos aplicados.....	56
Tabela 8: Resultado MAUT das tarefas.....	56
Tabela 9: Resultado MAUT dos operários.....	57
Tabela 10: Diferença avaliativa dos operários.....	57
Tabela 11: Preferências dos operários.....	59
Tabela 12: Matriz agregado de preferência das tarefas.....	62
Tabela 13: Matriz agregado de preferência dos operários.....	63
Tabela 14: Fluxos das tarefas.....	64
Tabela 15: Fluxos dos operários.....	64
Tabela 16: Resultado PROMETHEE II das tarefas.....	65
Tabela 17: Resultado PROMETHEE II dos operários.....	65
Tabela 18: Normalização TOPSIS das tarefas multiplicadas pelo respectivo peso de cada critério.....	66
Tabela 19: Normalização TOPSIS dos operários multiplicadas pelo respectivo peso de cada critério.....	66
Tabela 20: Distâncias às hipóteses das tarefas.....	67
Tabela 21: Distâncias às hipóteses dos operários.....	68
Tabela 22: Resultado TOPSIS das tarefas.....	68
Tabela 23: Resultado TOPSIS dos operários.....	69
Tabela 24: Medidas das tarefas.....	70
Tabela 25: Medidas dos operários.....	70

Tabela 26: Resultado VIKOR das tarefas.....	71
Tabela 27: Resultado VIKOR dos operários.....	71
Tabela 28: Atribuição MAUT.....	72
Tabela 29: Atribuição PROMETHEE II.....	73
Tabela 30: Atribuição TOPSIS.....	73
Tabela 31: Atribuição VIKOR.....	74
Tabela 32: Resultados do questionário.....	75
Tabela 33: Diferença avaliativa das Tarefas.....	83
Tabela 34: Preferências das Tarefas.....	89

Acrónimos e Símbolos

Lista de Acrónimos

FAST	<i>Function Analysis System Technique</i>
FURPS	<i>Funcionalidade, Usabilidade, Fiabilidade, Desempenho e Suportabilidade</i>
GRASP	<i>General Responsibility Assignment Software Patterns</i>
MAUT	<i>Multi-Attribute Utility Theory</i>
MAVT	<i>Multi-Attribute Value Theory</i>
MCDM	<i>Multiple-criteria decision-making</i>
MVC	<i>Model-View-Controller</i>
NCD	<i>New Concept Development Model</i>
NPD	<i>New Product Development</i>
PROMETHEE II	<i>Preference Ranking Organization Method for Enrichment Evaluation</i>
SAW	<i>Simple Additive Weighting</i>
SMART	<i>Simple Multi-Attribute Rating Technique</i>
SOLID	<i>Single responsibility, Open-closed, Liskov substitution, Interface segregation e Dependency inversion</i>
TOPSIS	<i>Technique for Order of Preference by Similarity to Ideal Solution</i>
TSG	<i>Technology Stage Gate</i>
UML	<i>Unified Modeling Language</i>
VIKOR	<i>Vlse Kriterijumska Optimizacija I Kompromisno Resenje</i>

1 Introdução

Esta secção fornece a informação requerida para a compreensão e enumeração dos objetivos propostos, a área em que estes possivelmente incidem, as diferentes soluções possíveis, o método de avaliação das mesmas, indicação do que é uma boa solução e a apresentação dos possíveis benefícios e valores decorrentes da concretização dos objetivos.

1.1 Enquadramento/Contexto

O contínuo crescimento populacional implica o crescimento do público-alvo de qualquer produto, mas se este for realizado de forma personalizada torna-se incrementalmente mais difícil de responder às necessidades dos consumidores. Para responder a este aumento de indivíduos as indústrias podem aumentar o custo do produto, diminuir a quantidade possível de personalização e padronizar os respetivos produtos, ou optar por uma outra combinação possível dos anteriores. As empresas que escolhem diminuir ou até mesmo eliminar a hipótese de personalização, através da padronização, desenvolveram a automação. Esta leva à redução dos preços dos produtos e ao aumento escalável de produção. No entanto, a capacidade de ter algo único é uma força atrativa na venda de um produto, ao ponto de as empresas tentarem

capitalizar nas opções personalizáveis. Atualmente as empresas utilizam táticas de marketing em conjunto com anúncios para aumentar o interesse de um determinado produto, mostrando que é possível obtê-lo de acordo com o gosto de cada indivíduo.

A área de automação permitiu aumentar as quantidades e reduzir o custo dos produtos e com os desenvolvimentos mais recentes nesta área, é possível integrar sistemas de personalização de produtos de forma viável.

A automação é um termo que descreve a diminuição da intervenção humana num dado processo (IBM, 2021) . Um exemplo disto são as linhas de montagem de veículos. Antes do processo de automação a montagem de um veículo era realizada através do esforço e interferência humana. Atualmente, conseguimos ver as linhas de montagem povoadas por braços robóticos a realizar as mesmas tarefas do seu antecessor humano de forma autônoma.

É possível atingir a automação de processos através de múltiplas formas, como por exemplo mecanismos hidráulicos, mecânicos, pneumáticos, elétricos, computacionais e também através das diferentes combinações dos referidos. Devido à variedade de formas possíveis de atingir a automação, esta consegue ser dividida em múltiplas formas de acordo com os mecanismos usados. Um exemplo disto seria a robótica que utiliza mecanismos mecânicos combinados com eletrônica e mecanismos computacionais. Dentro dos mecanismos computacionais temos a possível utilização de inteligência artificial e de algoritmos de decisão. Embora a robótica não exclua o uso dos processos computacionais de automação referidos, este exemplo é utilizado para evidenciar a existência de variadas técnicas de automação de acordo com cada tipo de mecanismo.

Dentro dos mecanismos computacionais, uma possível divisão consiste em:

- Automação básica (*Basic automation*)
- Automação de processos (*Process automation*)
- Automação através de integração (*Integration automation*)
- Automação através de Inteligência Artificial (*Artificial intelligence automation*)

A automação básica automatiza tarefas simples e rudimentares. Exemplos disto é enviar uma mensagem quando um processo acaba de ser realizado.

A automação de processos gere práticas de negócio trazendo uma maior transparência e uniformidade sobre o mesmo. É normalmente realizado através de software dedicado, os quais podem trazer uma perspectiva nova sobre um negócio e sugerir soluções inovadoras. Exemplo disto seria um software de gestão de redes de comunicação.

A automação através de integração copia a maneira que os humanos realizam tarefas dentro de um conjunto de regras específico. Exemplo disto é o “trabalhador digital”, como por exemplo a aplicação IFTTT (IFTTT, n.d.).

A automação através de Inteligência Artificial usa mecanismos que conseguem aprender e/ou realizar decisões baseadas em dados anteriores. Exemplo disto pode ser um assistente virtual que, de acordo com o problema reportado e os detalhes do mesmo, vai realizar um conjunto de operações baseadas em resoluções do mesmo problema ou parecidos que foram encontrados anteriormente.

Existe também um outro tipo de divisão dos tipos de processos de automação, sendo esta baseada na variedade de produtos e na quantidade de volume de produção. Esta divisão separa a automação em três categorias:

- Automação rígida (*Fixed Automation*)
- Automação programável (*Programmable Automation*)
- Automação flexível (*Flexible Automation*)

A automação rígida é uma sequência de várias operações que não podem ser customizáveis, ou seja, é um processo que realiza um conjunto de ações e devolve sempre o mesmo resultado. Este tipo de automação é o ideal para a produção de elevadas quantidades do mesmo produto (Industrialautomation360.com, n.d.).

Automação programável permite a mudança das operações com base nos requerimentos indicados por um utilizador. Permite atingir múltiplos resultados e variações do mesmo produto, no entanto está fortemente dependente da intervenção humana e requer mais atenção da parte do operador humano pois o mínimo erro deste pode criar produtos com as especificações erradas.

E finalmente, a automação flexível tenta chegar a um meio termo entre os dois tipos anteriormente referidos. Permite alguma customização, mas com algumas restrições.(Industrialautomation360.com, n.d.)(GeeksforGeeks, 2020)

As relações dos tipos de automação anteriormente referidos com a quantidade e a variedade de produtos pode ser exemplificada pela Figura 1.

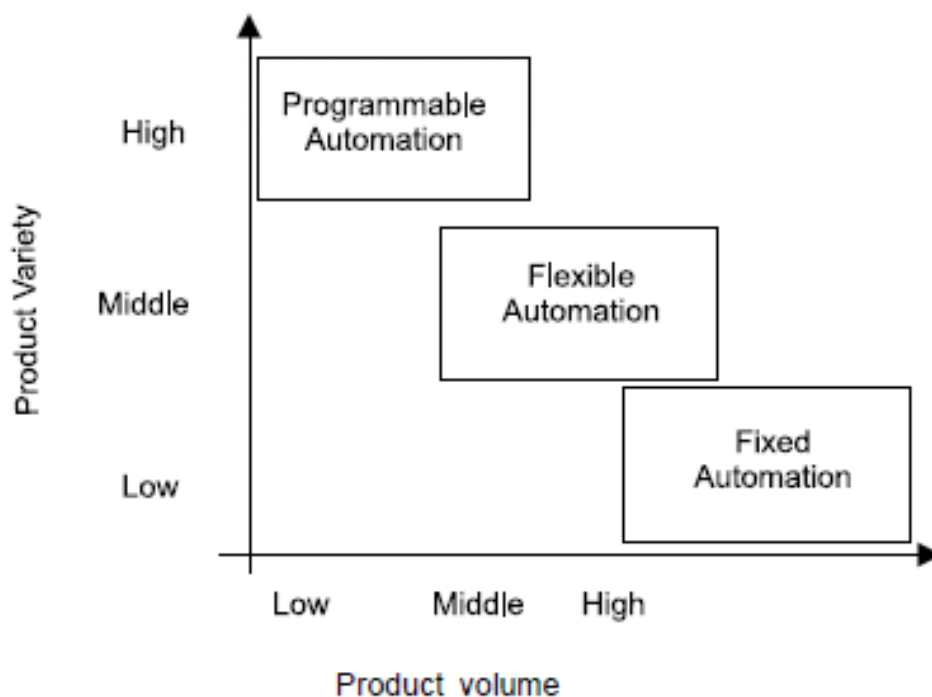


Figura 1: Relação de tipos de automação com a variedade e quantidade de produção

1.2 Descrição do problema

Em qualquer tipo de trabalho é uma boa prática planejar e dividir tarefas. Compreende-se que uma tarefa é um conjunto de ações manuais realizadas por um operário para atingir um objetivo específico. A divisão de tarefas melhora os processos logísticos e organizacionais de uma empresa, pois permite ver dependências entre tarefas, indicar as necessidades das mesmas e

realizar um esboço temporal para a conclusão de um determinado trabalho. No entanto, estas tarefas precisam de ser atribuídas aos operários. O processo de atribuição é a delegação de um conjunto de tarefas a um grupo de operários. Este processo pode consumir bastante tempo, pois alguns operários podem ser mais eficientes a realizar um determinado tipo de tarefas do que as outras tarefas. Também devido à natureza analítica do processo de atribuição, algumas tarefas podem ficar por realizar, especialmente quando este envolve um elevado número de tarefas, ou quando surgem tarefas de realização esporádica. Estas situações podem trazer atrasos, não completação de encomendas/produtos e custos de funcionamento acrescidos. Estes problemas podem resultar em conflitos com os clientes, pois estes podem ter um prazo para obter o produto vendido e/ou terem um orçamento rígido para o mesmo produto.

1.2.1 Objetivos

Verifica-se a existência de algumas falhas no processo de atribuição de tarefas e a elevada dependência de certos indivíduos para realizar tarefas específicas. É imperativo obter a informação acerca das tarefas completadas por cada indivíduo, traduzir essa informação para um formato mais útil e, baseado na informação tratada, distribuir automaticamente as tarefas pelos operários de forma equilibrada entre eles, ou seja, um operário não pode só ter as tarefas de maior relevância, nem ter a maior parte das tarefas. Nenhuma tarefa deve ficar por atribuir. Prevê-se que o cumprimento dos objetivos irá ajudar a certificar que o conhecimento do funcionamento do produto seja distribuído pelos operadores. Também se espera a remoção de um indivíduo encarregue pela distribuição de tarefas e pela responsabilidade ética e moral das distribuições das mesmas.

1.2.2 Abordagem

Este problema é não determinístico, ou seja, não tem só uma única solução. O ato de distribuição pode ter uma única solução, mas esta é somente alcançada quando tratamos de dados quantificáveis e para situações de complexidade reduzida. No entanto, o problema está dependente da proficiência dos indivíduos para realizar tarefas, o que pode conter um nível de incerteza associado. A capacidade de um indivíduo realizar uma tarefa é difícil de quantificar, e o processo de quantificação pode estar predisposto a fatores subjetivos alargando a margem de erro na avaliação.

Tendo em conta a natureza deste problema, inicialmente vão ser recolhidas as possíveis categorias de informação relevantes para a avaliação e quantificação dos indivíduos. Este tipo de informação pode ser a taxa de completção de tarefas dentro do tempo alocado à realização da mesma (número de tarefas completadas dentro do prazo em relação ao número total de tarefas realizadas), a quantidade de tarefas realizadas, o nível de complexidade das tarefas completadas, entre outros.

Após ter as categorias é necessário criar um conjunto de dados para estudar o desenvolvimento realizado. Estes dados serão gerados aleatoriamente e guardados nas tabelas presentes nos anexos. Embora estes dados não sejam representativos do mercado de trabalho, deste modo é possível obter um espectro mais alargado das possíveis classificações, facilitando na avaliação de resultados através da presença de operários com diferenças mais distintas.

A seguir, é aplicada a informação recolhida aos algoritmos de atribuição. Devido a este problema ser não determinístico vão ser estudadas variações dos algoritmos para realização da atribuição de tarefas. Estes resultados estão presentes nas tabelas em anexo.

Finalmente, após a obtenção dos resultados é feita uma análise crítica dos mesmos, tendo em consideração os algoritmos utilizados.

1.2.3 Avaliação de resultados

Considerando que este problema pode ter uma componente subjetiva, é possível influenciar os resultados inadvertidamente. Portanto, é realizado um questionário anónimo. Este questionário pede a cada indivíduo para avaliar as atribuições realizadas por cada variação do algoritmo através da classificação relativa entre eles (se preferem as atribuições de um algoritmo sobre outro). Estes resultados estão presentes nas tabelas da análise dos resultados e os mesmos são discutidos na análise dos resultados.

1.2.4 Contributos

Este tipo de funcionalidade mostra a um cliente a disponibilidade e a compreensão do fornecedor do serviço, pois este necessita de conhecer a área de funcionamento do cliente e o esforço que este tem na gestão e atribuição de tarefas. A utilização deste tipo de serviço por um cliente implica que ele está disposto a confiar no fornecedor, com a parte logística da distribuição de tarefas do negócio. Deste modo o cliente poupa esforço, tempo e recursos, enquanto o criador do serviço acresce valor do produto e é reconhecido pela confiança alocada neles por parte dos clientes.

Este serviço também permite a diminuição de tarefas que ficam por concluir devido à falha de alocação de indivíduos nas mesmas, e à redução de esforço devido à automatização do processo. E dependente da situação poderá realizar uma melhor distribuição em menor tempo do que o processo manual.

1.3 Estrutura do relatório

Esta dissertação é composta por seis capítulos, incluindo o primeiro capítulo correspondente à introdução. No segundo, apresenta-se o estado da arte, que estuda trabalhos relacionados, assim como as tecnologias existentes. No terceiro, faz-se a análise de valor seguido no quarto

capítulo pela análise e o desenho da solução, estudando os algoritmos escolhidos com mais detalhe. No quinto capítulo, dá-se a conhecer a implementação da solução, os resultados obtidos, a análise dos resultados e realizada uma avaliação sobre os mesmos. Finalmente, no sexto capítulo, apresenta-se uma conclusão, na qual o estudo é revisto e é dada uma apreciação final.

2 Estado da Arte

Este capítulo é utilizado para comentar acerca de plataformas de gestão e organização de tarefas, e quais as possíveis tecnologias existentes para alcançar uma possível solução.

No intuito da realização da análise de plataformas de gestão e organização de tarefas, é de notar que o conceito base de criação e gestão de tarefas é significativamente similar. Considerando que o projeto está relacionado com o desenvolvimento de automatização dos processos de atribuição de tarefas, assumiu-se que as plataformas desenvolvidas são projetos ou trabalhos que estão relacionados com o desenvolvimento do projeto, pois estas podem expandir o desenvolvimento delas para incorporar este projeto.

2.1 Trabalhos relacionados

Gestão e organização de tarefas é algo cada vez mais importante no contexto empresarial, especialmente em empresas em crescimento, pois permite um funcionamento eficaz juntamente com um registo eficiente de tarefas necessárias para manter o funcionamento da empresa e satisfazer os pedidos dos clientes.

Com a corrida à informatização, foram criados múltiplos softwares e desenvolvimentos do método operacional, no que refere à gestão e organização de tarefas, resultando na implantação informática de ferramentas que assistem a metodologia *Scrum*.

Scrum é uma metodologia que permite equipas gerirem o seu trabalho, identificando que tarefas estão por fazer, as que estão em progresso, as que estão concluídas e a quem estão atribuídas. Existem variações da caracterização das tarefas, mas a base consiste habitualmente no que foi anteriormente referido (mijacobs, n.d.).

Tendo o anterior em conta, podemos afirmar que são as funcionalidades adicionais e os preços que irão distinguir as plataformas. Foram escolhidos quatro produtos de gestão e organização de tarefas, os quais vão ser avaliados de acordo com o tema deste projeto, ou seja, identificando se o produto vendido tem um mecanismo para atribuir tarefas de forma automática e, caso tenha, identificando o que sustenta o mecanismo. As empresas escolhidas foram:

- *Profit.co*
- *Miro*
- *monday.com*
- *PollUnit*

A *Profit.co* tem um produto cujas funções são limitadas de acordo com o plano de pagamento, no entanto, este oferece uma extensa quantidade de funcionalidades, cujas podem ser expandidas através da integração com outras aplicações. O produto da *Profit.co* não apresenta nenhum sistema de automação de distribuição de tarefas (Profit.co., n.d.).

A *Miro*, tal como a *Profit.co*, apresenta uma lista de funções de acordo com o plano de pagamento escolhido. Também, tal como a *Profit.co*, a *Miro* apresenta uma lista extensa de funcionalidades, mas algumas destas são diferentes da *Profit.co*. E finalmente, tal como a *Profit.co*, o produto da *Miro* não tem a funcionalidade de automação de distribuição de tarefas (Miro, n.d.).

Os produtos da *Miro* e da *Profit.co* foram escolhidos por exemplificarem múltiplos outros produtos que existem no mercado e que não apresentam a capacidade de automação de distribuição de tarefas.

O produto *monday.com* apresenta funcionalidades diferentes para planos de pagamentos diferentes, semelhante aos produtos referidos anteriormente. De igual modo apresenta uma lista extensa de funcionalidades diferentes dos produtos anteriores. No entanto, o *monday.com* afirma ter automação de tarefas. Após análise do produto, o *monday.com* refere a automação de tarefas como a criação das mesmas após um determinado período de tempo. Esta solução não realiza a distribuição automática das mesmas, pelo que não corresponde ao tema a ser estudado neste projeto (monday, n.d.).

E finalmente temos o *PollUnit*. Igualmente aos produtos anteriores este apresenta funcionalidades diferentes dependente do plano de pagamento, uma lista extensa de funcionalidades diferentes e, como o *monday.com*, afirma ter automação de tarefas. A automação de tarefas para o *PollUnit* é a distribuição automática de tarefas pelos utilizadores. No entanto as tarefas são distribuídas de acordo com a preferência dos utilizadores para cada tarefa. Este produto realiza de facto a distribuição de tarefas de forma automática, mas pode ser considerado como limitado devido a este utilizar só as preferências dos utilizadores para cada tarefa (PollUnit. n.d.).

Concluindo, a existência de um mecanismo que realiza a distribuição de tarefas automaticamente é raro embora múltiplos produtos apresentem condições para integrar ou melhorar um mecanismo que realiza a função pretendida. A informação referida anteriormente pode ser estruturada sob a forma da Tabela 1.

Tabela 1: Distribuição automática de tarefas em produtos comerciais

Empresa	Realiza Distribuição automática de tarefas	Limitações
<i>Profit.co</i>	Não	Não aplicável
<i>Miro</i>	Não	Não aplicável
monday.com	Não	Não aplicável
PollUnit	Sim	Só utiliza a preferência dos utilizadores

2.2 Tecnologias existentes

Para o desenvolvimento deste projeto foi optado por não adotar nenhum produto comercial com capacidades de gestão e organização de tarefas, pois o foco deste projeto é a capacidade de distribuir automaticamente tarefas, ou seja, neste projeto é pretendido desacoplar e estudar a funcionalidade anteriormente descrita, para que seja facilmente integrada noutros projetos ou produtos. Deste modo, as tecnologias e dependências existentes para este trabalho são significativamente reduzidas.

Considerando o que foi descrito anteriormente e tendo em conta os objetivos deste projeto podemos identificar duas tecnologias fundamentais para o alcance dos objetivos do projeto.

Estas são:

- linguagem de programação
- tipo de algoritmo

Para a linguagem de programação, é necessário com que esta tenha capacidade de reconhecer, analisar, identificar e decidir de acordo com um sistema de regras para a manipulação de dados (*Turing complete*). A linguagem também necessita ter a capacidade de aceder a informação guardada sob qualquer forma, desde ficheiros de texto a bases de dados. A maior parte das linguagens de programação comuns (Java, Python, C, C++, Javascript, PHP, entre outras) cumprem estes requisitos, no entanto, para a realização deste projeto foi selecionada Java.

Para o algoritmo é necessário com que ele tenha a capacidade de receber informação acerca dos critérios a serem analisados, sintetizar a mesma de forma que seja utilizável, e depois realizar uma atribuição de tarefas aos diversos operários. Para a primeira parte algoritmos de decisão multi-critério (*Multiple-criteria decision-making (MCDM)*) conseguem cumprir o objetivo enquanto a atribuição pode ser realizada por algoritmos de escalonamento de processos.

Algoritmos de decisão multi-critério é uma disciplina que tenta avaliar decisores baseados em múltiplos critérios conflituosos. Um exemplo disto seria a compra de um carro. Na compra de um carro é necessário analisar o consumo, a reparabilidade, o custo, a distância percorrida pelo veículo, entre outros critérios. Cada um destes critérios são avaliados de forma diferente e o conjunto deles resulta na priorização de um carro em relação aos restantes. Muitas destas decisões são realizadas diariamente pelo ser humano baseando-se na experiência e intuição dele. Dado a subjetividade da experiência e intuição, nota-se que não existe uma solução única para problemas que necessitem algoritmos de decisão multi-critério. No entanto, estes mesmos algoritmos tentam encontrar uma solução que se aproxime ou chegue ao melhor resultado possível (Wątróbski et al., 2019).

Algoritmos de decisão multi-critério é uma área de investigação relativamente ativa, existindo uma vasta variedade dos mesmos. Mas estes podem ser filtrados de acordo com os objetivos, os pesos, a incerteza, as escalas e o tipo de dados utilizados. A escolha e os algoritmos em questão serão aprofundados na análise da solução.

Algoritmos de escalonamento são normalmente utilizados em sistemas computacionais que atribuem recursos computacionais a tarefas computacionais. Este tipo de algoritmos devem contemplar um ou múltiplos dos seguintes objetivos:

- maximizar a quantidade de trabalho realizado por unidade de tempo
- minimizar o tempo de espera de execução de tarefas
- minimizar o tempo de resposta para utilizadores
- maximizar o equilíbrio da utilização de recursos

Alguns exemplos destes algoritmos são:

- *First In, First Out*
- *Priority Queue*
- *Round-Robin*
- *Multiple Queue*
- etc.

First In, First Out é um algoritmo onde as tarefas são imediatamente tratadas de acordo com a ordem de aparecimento. É um algoritmo de implementação simples, no entanto, é muito sensível à ordem em que as tarefas aparecem (Jacek Blazewicz and AI, 2001).

Priority Queue implica as tarefas terem uma prioridade associada e são resolvidas de acordo com a prioridade apresentada. Este algoritmo permite alguma flexibilidade devido a ser possível combinar com outros algoritmos para as tarefas com a mesma prioridade (Jacek Blazewicz and AI, 2001).

Round-Robin tenta dividir as tarefas de forma igual, por exemplo, se tenho três tarefas e três operários, cada operário irá ter uma tarefa. Este algoritmo permite a existência de bons tempos de resposta de acordo com o número de tarefas (Jacek Blazewicz and AI, 2001).

Multiple Queue é muito parecido com o algoritmo de prioridade, mas permite utilizar outros critérios para a ordem de resolução de tarefas em vez da prioridade. Tal como o algoritmo de prioridade este é possível ser combinado com outros (Jacek Blazewicz and AI, 2001).

Existem mais algoritmos de escalonamento computacional além daqueles referidos. Estes algoritmos são constantemente estudados e utilizados em sistemas informáticos, onde irão ter mais vantagens, desvantagens e complexidades de acordo com o ambiente utilizado (Jacek Blazewicz and AI, 2001).

2.3 Conclusão

Neste capítulo, foi verificado a necessidade de utilizar algoritmos de decisão multi-critério em combinação com algoritmos de escalonamento de processos, e comprovada a variedade destes dois tipos de algoritmos. Também foi verificado que não existem produtos que satisfaçam

a resolução deste problema e quando excepcionalmente respondem não têm em consideração as características das tarefas e dos operários no momento da atribuição.

3 Análise de valor

A análise de valor é um processo sistemático utilizado para identificar a aptidão de um produto ou de um serviço para alcançar os objetivos impostos pelo respetivo cliente. Para a compreensão do produto ou serviço, estes devem ser analisados na qual a finalidade dos mesmos deve ser considerada.

Esta secção é utilizada para comunicar o processo de inovação e aplicar uma análise de valor a este projeto.

3.1 Processo de inovação

O processo de inovação procura melhorar o valor de um produto ou de um serviço, através de técnicas analíticas. Estas técnicas tem o propósito de identificar as possíveis maneiras de alcançar um determinado objetivo e de reduzir as ineficiências de um produto ou de um serviço, diminuir risco associado com os anteriores e diminuir o número de decisões com resultados prejudiciais. O processo de inovação pode ser dividido em três fases, tal como demonstrado pela Figura 2, sendo estas:

- *Fuzzy Front End*
- *New Product Development*
- *Commercialization*

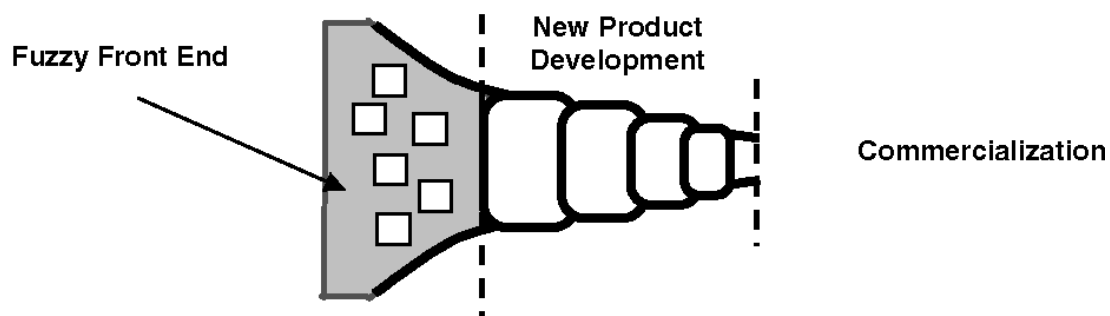


Figura 2: Processo de Inovação (A.Koen, et al., 2002)

A fase de *Fuzzy Front End* exemplifica a incerteza de conhecimento dos recursos disponíveis e o processo inicial de experimentação e de criação de protótipos que irá permitir gerar a ideia para a criação de um produto ou de um serviço.

A fase de *New Product Development* representa a definição rígida dos objetivos a serem alcançados e o planeamento da utilização de todos os recursos disponíveis.

Finalmente, a fase de *Commercialization* indica a distribuição e comercialização de um produto ou de um serviço.

O processo de inovação foi aprofundado através da utilização do modelo *New Concept Development Model* (NCD), desenvolvido por Koen.

3.2 New Concept Development

O modelo NCD separa o processo de inovação em múltiplas partes, demonstradas pela Figura 3. Este processo começa pela geração de uma ideia ou pela identificação de uma oportunidade, acabando na definição de conceito. A definição de conceito permite atingir as fases de

New Product Development (NPD) ou de *Technology Stage Gate* (TSG). A forma circular deste modelo tenta indicar a indispensabilidade da informação transitar por todos os elementos do modelo.

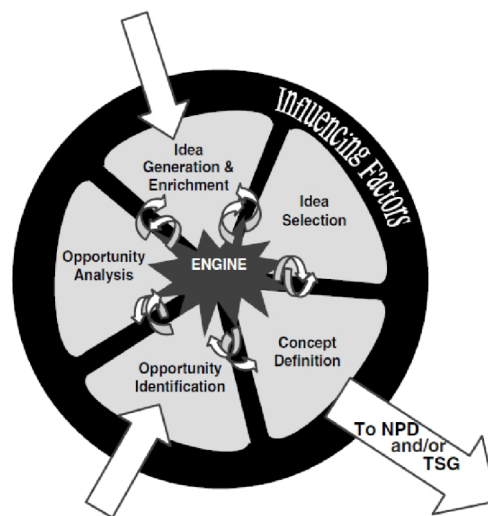


Figura 3: – Modelo NCD (A.Koen, et al., 2002)

3.2.1 Elementos do NCD

O modelo NCD é composto por seis elementos:

- Motor (*Engine*)
- Identificação da Oportunidade (*Opportunity Identifications*)
- Geração de Ideias (*Idea generation & Enrichment*)
- Análise da Oportunidade (*Opportunity Analysis*)
- Seleção de Ideias (*Idea Selection*)
- Definição do conceito (*Concept Defenition*)

3.2.1.1 Motor

O motor é o elemento responsável pela interação dos elementos do modelo.

3.2.1.2 Identificação da Oportunidade

Uma oportunidade pode ser expressa sob múltiplas formas. Problemas, avanços tecnológicos e melhoramento de processos podem constituir exemplos de oportunidades.

Na secção 1.2 foi identificado e explorado um problema, a existência de falhas no processo de atribuição de tarefas, o consumo de recursos para a realização da distribuição de tarefas e a presença de dependências indevida de indivíduos especializados na realização de tarefas específicas.

Através da automação deste processo é possível resolver, ou mitigar, os problemas identificados tornando o processo de atribuição de tarefas mais fluido e eficiente a nível da utilização de recursos temporais e económicos.

3.2.1.3 Análise da Oportunidade

Esta análise foi efetuada tendo em conta os produtos existentes no mercado e a corrida a sistemas automáticos.

No período da realização deste documento, podemos observar que o mercado depende e incentiva a incrementação da utilização de sistemas automáticos. Aqueles que aparecem no mercado costumam mostrar uma capacidade suficientemente superior ou com custos menores do que o processo manual. Exemplos de processos automatizados podem ser gestão, recolha e armazenamento de produtos num armazém (por exemplo, robôs que trabalham nos centros de distribuição da Amazon), sistemas de gestão de redes, entre outros.

Atualmente, é possível identificar variados produtos de gestão de tarefas, no entanto, são poucos os que oferecem um sistema automatizado para a atribuição de tarefas. E estes produtos que têm o sistema referido normalmente baseiam-se em fatores subjetivos introduzidos pelos utilizadores como por exemplo a preferência da realização de uma tarefa.

Com os dados anteriores podemos verificar que a automatização do processo baseada em fatores objetivos pode ser um fator de escolha de um produto sobre outro, considerando a carência deste tipo de automação no mercado.

3.2.1.4 Geração de Ideias

Tendo em conta a informação referida, foi pensado como atingir os objetivos propostos e os recursos necessários para alcançá-los.

Este documento pretende focar no processo de automação de tarefas, portanto a criação de um protótipo permite alcançar o objetivo sem requerer a utilização ou criação de um sistema de gestão de tarefas completo. Deste modo conseguimos o desacoplamento deste serviço facilitando a integração noutros produtos já existentes em mercado. Neste protótipo ainda é possível encontrar múltiplas divergências na criação do algoritmo para a realização do processo de automatização. Devem ser realizados múltiplos protótipos com algoritmos diferentes para permitir a exploração da solução.

3.2.1.5 Seleção de Ideias

O processo de distribuição de tarefas dificilmente tem sempre a mesma solução, especialmente quando tratamos de ambientes com um elevado número de tarefas. Esta situação acontece também a nível da automação, onde diferentes algoritmos podem gerar diferentes soluções.

Considerando a informação anterior a solução deve ser fundamentada na preferência do algoritmo realizada por indivíduos que pratiquem a distribuição de tarefas.

3.2.1.6 Definição do conceito

A solução escolhida é o algoritmo que realiza a distribuição de tarefas mais preferida pelos indivíduos questionados, tendo em conta o custo operacional e temporal do algoritmo.

3.3 Valor

A definição de valor varia de acordo com o contexto, pois este difere de acordo com o observador. O que pode ser considerado valioso para uns, pode ser visto como banal por outros. No entanto, através da adoção desta funcionalidade num produto já existente, poderemos observar um acréscimo de qualidades e funções do mesmo. Para os utilizadores desse produto poderão diminuir o risco de uma distribuição de tarefas deficiente, reduzir custos temporais e económicos através da remoção do processo de distribuição de tarefas de forma manual e a conveniência de reduzir a quantidade de funções necessárias ao funcionamento de uma empresa. Atendendo ao que foi referido, para o cliente poder adotar este tipo de funcionalidade, terá que adotar uma maior atenção às estatísticas dos seus funcionários e ceder controlo na distribuição de tarefas.

3.3.1 Valor percebido para o cliente

O valor percebido pode ser definido como a relação entre benefícios e custos. Para um cliente adotar uma solução os benefícios têm que superar os custos. É de notar que o peso dos benefícios e dos custos varia de acordo com as prioridades e os contextos percebidos.

Tal como indicado anteriormente, a adoção desta funcionalidade permite o aumento da qualidade de um produto de gestão de tarefas, traz uma maior flexibilidade, confiança e consistência, diminuindo os custos a quem utiliza o respetivo produto.

3.4 Proposta de valor

A proposta de valor visa sintetizar os possíveis ganhos e diferenciar da competição através da aquisição um determinado produto ou serviço.

Este serviço pretende remover as preocupações de distribuir tarefas permitindo ao cliente focar no mais importante.

Para o aprofundamento da proposta de valor foi utilizado o modelo de Osterwalder exemplificado pela Figura 4.

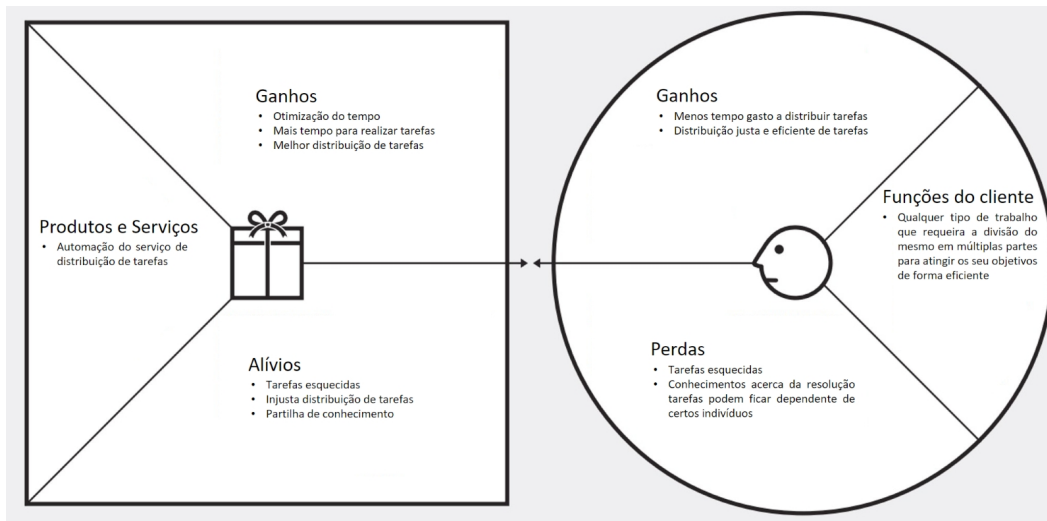


Figura 4: Modelo de Osterwalder para o projeto

3.5 Diagrama FAST

O diagrama FAST (*Function Analysis System Technique*) é uma técnica que permite a representação visual das relações lógicas entre as funções de um processo, produto ou projeto baseando-se nas questões “Como”, “Porquê” e “Quando”. Este quando é lido da esquerda para a direita realiza e responde as questões “Como”. Se for lido na direção contrária realiza e responde as questões “Porquê”. Quando se encontra um caminho vertical, este realiza e responde as questões “Quando”.

Com base no que foi referido na secção 1.2.2, a Figura 5 representa as ações referidas nessa secção sob a forma de um diagrama FAST.

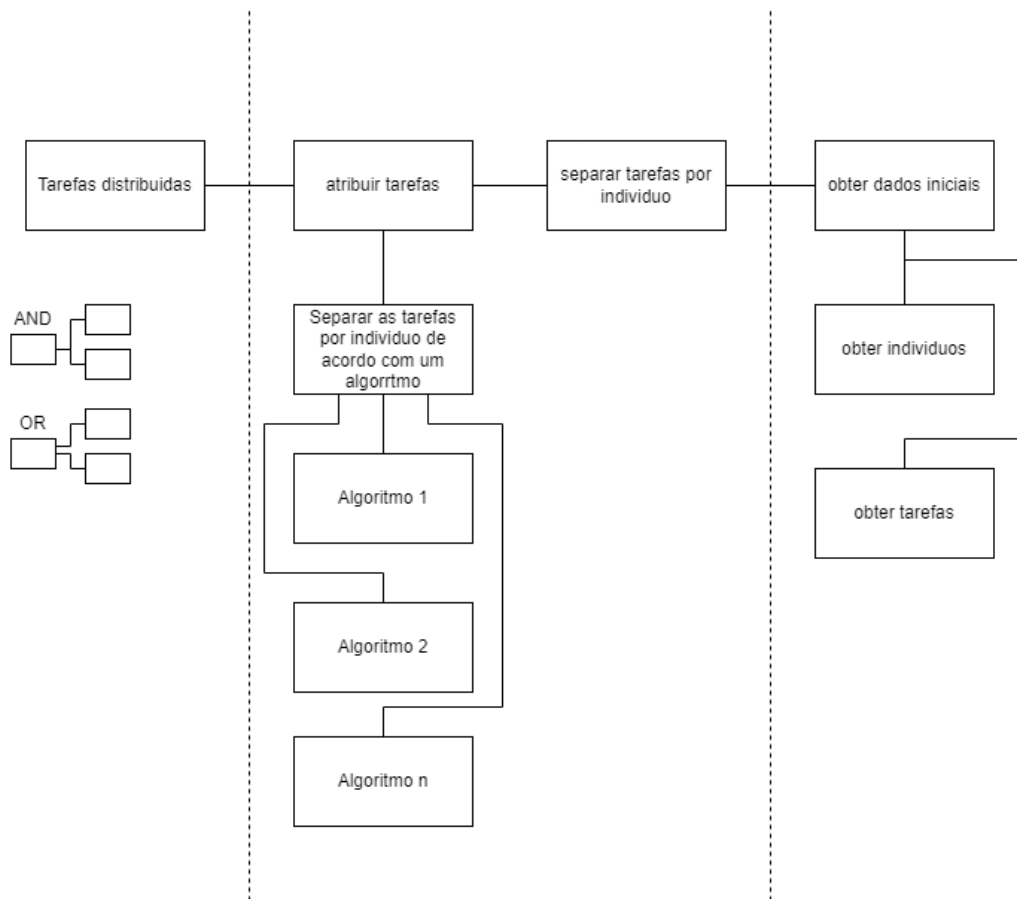


Figura 5: Modelo de FAST do projeto

3.6 Conclusão

Neste capítulo, através da análise de valor, é mostrado que este projeto é uma mais valia para qualquer produto que não tenha uma implementação capaz de responder ao problema descrito.

4 Análise e desenho da solução

Esta secção é realizada com o objetivo de indicar os resultados da análise e do desenho da solução. A informação disponibilizada neste capítulo vai justificar algumas escolhas na solução e indicar o estruturamento da mesma.

4.1 Domínio do problema

O domínio do problema é a área onde o problema irá incidir. Este vai ter toda a informação da área estudada e vai indicar como cada componente se relaciona com os restantes. Para a descrição do domínio foi criado um modelo de domínio.

O modelo de domínio é uma abstração que ajuda a ilustração de ideias, objetos ou conceitos dentro da área do domínio identificando as representações mais importantes, e as relações entre elas (Fowler, 2015).

Foi utilizada a linguagem *Unified Modeling Language* (UML) para representar o modelo de domínio deste projeto, sendo este representado na Figura 6.

Podemos notar que o modelo de domínio é constituído por três elementos:

- Tarefa
- Operário
- Critério

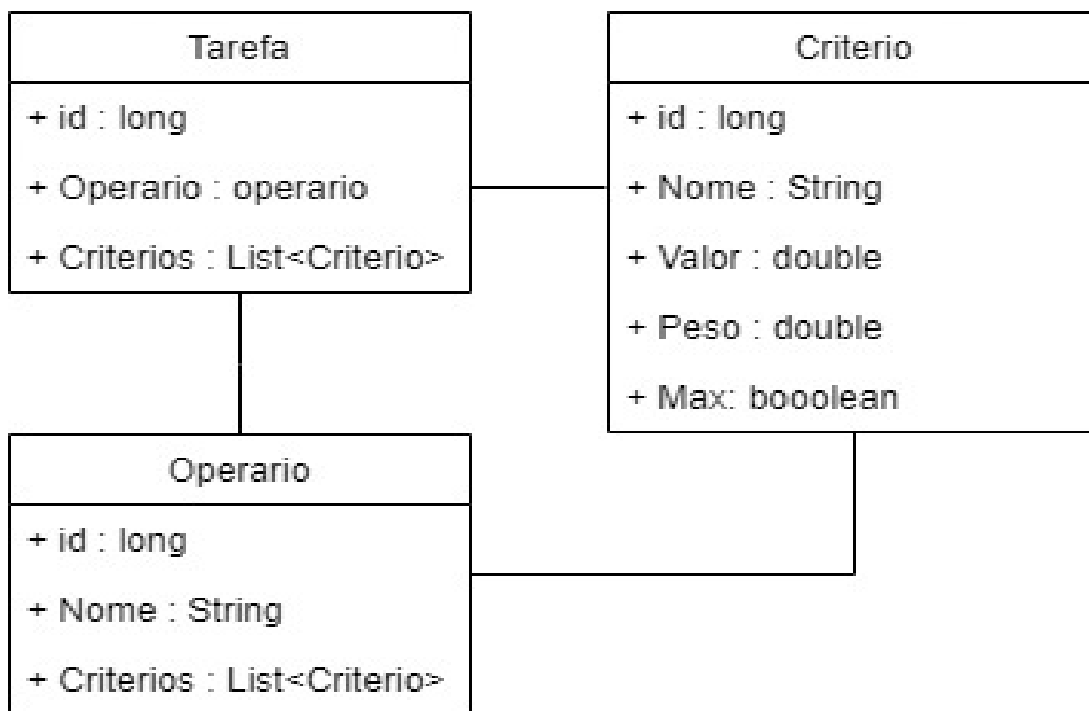


Figura 6: Modelo de domínio

É possível incorporar mais elementos ao modelo de domínio para satisfazer a construção base de um produto, no entanto, tais elementos não fazem parte do âmbito deste documento. Desta forma, este modelo de domínio deve ser assumido como algo complementar a modelos de domínio já existentes.

Tal como anteriormente referido uma tarefa representa um conjunto de ações manuais com o propósito de alcançar um determinado objetivo. Os passos necessários para chegar a um determinado fim vão variar significativamente entre tarefa, mas sabemos que elas necessitam de ser:

- identificadas

- completadas por um operário
- informativas acerca dos parâmetros da tarefa

O campo **id** permite criar um identificador único.

O campo **Operario** permite identificar qual o operário a realizar uma determinada tarefa.

O campo **Critérios** permite acumular todos os parâmetros/critérios de uma tarefa.

Um Operario identifica um utilizador ou funcionário. Neste contexto um operário tem que ter as capacidades de realizar qualquer tarefa. As informações guardadas por utilizador ou operário variam consideravelmente de acordo com a área e com o ambiente utilizado, contudo, para este desenvolvimento os utilizadores necessitam de ser:

- identificados
- Nome
- informativas acerca das qualidades de cada operário

O campo **id** permite criar um identificador único.

O campo **Nome** permite facilitar identificar o operário.

O campo **Critérios** permite identificar as qualidades/critérios de cada operário.

O Critério é utilizado para descrever qualidades e parâmetros, de forma estandardizada para com que seja possível tomar decisões acerca da informação guardada. O critério tem os seguintes campos:

- **id**
- **Nome**
- **Valor**
- **Peso**
- **Max**

O campo **id** é utilizado para criar um identificador único por cada critério criado.

O campo **Nome** é utilizado para informar que tipo de critério está a ser tratado. Este campo pode ser removido se o Critério for implementado como uma interface ou uma classe abstrata. Neste desenvolvimento todos os critérios vão ter as mesmas funcionalidades, vão ser tra-

tados de igual forma e não vão ser conceptualmente diferentes entre eles. Devido a este detalhe este campo foi criado para poder diferenciar entre critérios pois a gama de valores de cada critério pode divergir.

O campo **Valor** é utilizado para indicar qual o valor atingido num determinado critério.

O campo **Peso** é utilizado para informar a relevância de um critério. Quanto maior é o peso de um critério, mais relevante ele é.

O campo **Max** é utilizado para revelar se um determinado critério é benéfico quanto maior ou menor ele for. Um exemplo disto seria um critério custo e um critério potência de um veículo. Neste exemplo, o critério potência vai tentar ser maximizado, enquanto o critério custo vai tentar ser minimizado.

4.2 Requisitos funcionais e não funcionais

Requisitos correspondem aos objetivos e restrições que o projeto necessita de cumprir. Estes dividem-se em requisitos funcionais e não funcionais. Os requisitos funcionais determinam as funções e ações necessárias para um projeto alcançar os respetivos objetivos. Os requisitos não funcionais preocupam-se em indicar como os objetivos são alcançados.

4.2.1 Requisitos funcionais

Para indicar os requisitos funcionais vai ser utilizado um diagrama de casos de uso. Um caso de uso reporta uma situação onde um processo ou serviço pode ser utilizado. Um diagrama de casos de uso representa todos os casos de uso num sistema e indica quem tem a capacidade de executar cada caso de uso. Neste estudo, os casos de usos são possíveis de serem utilizados por qualquer elemento, indivíduo ou sistema existente.

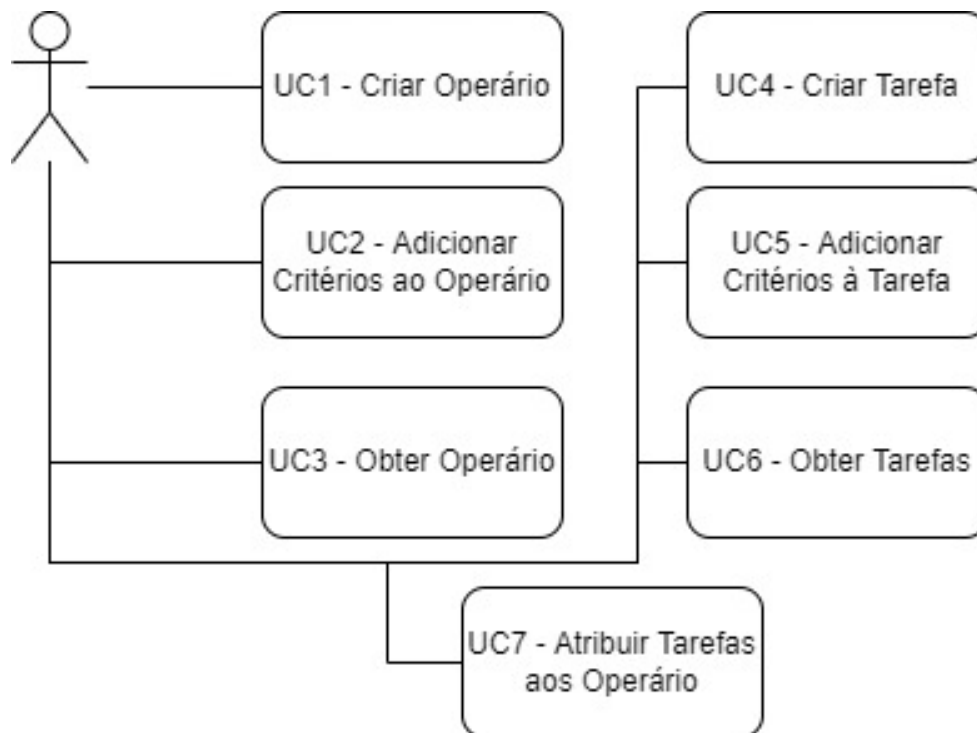


Figura 7: Diagrama de casos de uso

De seguida vão ser detalhados todos os casos de uso presentes na Figura 7.

4.2.1.1 UC1 – Criar Operário

Descrição: Para poder realizar uma atribuição de tarefas a operários é necessário que os operários existam. Este caso de uso refere à criação dos operários e à persistência da informação dos mesmos.

Pré-condições:

1. Base de dados tem que estar operacional.

Cenário principal de sucesso:

1. Criar operário.
2. Guardar operário.
3. Retornar operário.

Alternativas:

1. Não foi possível criar o operário.
 - 1.a. Acaba o caso de uso.
2. Não foi possível guardar o operário.
 - 2.a. Acaba o caso de uso.

4.2.1.2 UC2 – Adicionar Critérios ao Operário

Descrição: Para poder realizar uma atribuição de tarefas a operários é necessário que os operários tenham características que os permitam diferenciar entre eles. Este caso de uso refere à criação de critérios/características e a persistência dos mesmos em relação a um operário.

Pré-condições:

1. Base de dados tem que estar operacional.
2. O operário tem que existir.

Cenário principal de sucesso:

1. Obter operário.
2. Criar critério do operário.
3. Guardar critério.
4. Retornar operário.

Alternativas:

1. Não foi possível obter o operário.
 - 1.a. Acaba o caso de uso.
2. Não foi possível criar o critério do operário.
 - 2.a. Acaba o caso de uso.
3. Não foi possível gravar o critério do operário.
 - 3.a. Acaba o caso de uso.

4.2.1.3 UC3 – Obter Operários

Descrição: Para poder realizar uma atribuição de tarefas a operários é necessário obter os operários existentes e as respectivas características dos mesmos. Este caso de uso refere à obtenção dos operários e dos critérios/características deles.

Pré-condições:

1. Base de dados tem que estar operacional.

Cenário principal de sucesso:

1. Obter todos os operários com os respectivos critérios/características.

Alternativas:

4.2.1.4 UC4 – Criar Tarefa

Descrição: Para poder realizar uma atribuição de tarefas a operários é necessário que as tarefas existam. Este caso de uso refere à criação das tarefas e à persistência da informação das mesmas.

Pré-condições:

1. Base de dados tem que estar operacional.

Cenário principal de sucesso:

1. Criar tarefa.
2. Guardar tarefa.
3. Retornar tarefa.

Alternativas:

1. Não foi possível criar a tarefa.
 - 1.a. Acaba o caso de uso.
2. Não foi possível guardar a tarefa.
 - 2.a. Acaba o caso de uso.

4.2.1.5 UC5 – Adicionar Critérios à Tarefa

Descrição: Para poder realizar uma atribuição de tarefas a operários é necessário que as tarefas tenham parâmetros que permitam diferenciar entre elas. Este caso de uso refere à criação de critérios/parâmetros e à persistência dos mesmos em relação a uma tarefa.

Pré-condições:

1. Base de dados tem que estar operacional.
2. A tarefa tem que existir.

Cenário principal de sucesso:

1. Obter tarefa.
2. Criar critério da tarefa.
3. Guardar critério.
4. Retornar tarefa.

Alternativas:

1. Não foi possível obter a tarefa.
 - 1.a. Acaba o caso de uso.
2. Não foi possível criar o critério da tarefa.
 - 2.a. Acaba o caso de uso.
3. Não foi possível gravar o critério da tarefa.
 - 3.a. Acaba o caso de uso.

4.2.1.6 UC6 – Obter Tarefas

Descrição: Para poder realizar uma atribuição de tarefas a operários é necessário obter as tarefas existentes e os respectivos parâmetros das mesmas. Este caso de uso refere à obtenção das tarefas e dos critérios/parâmetros delas.

Pré-condições:

1. Base de dados tem que estar operacional.

Cenário principal de sucesso:

1. Obter todas as tarefas com os respectivos critérios/ parâmetros.

Alternativas:**4.2.1.7 UC7 – Atribuir Tarefas aos Operários**

Descrição: Este caso de uso refere à atribuição das tarefas e aos respectivos operários.

Pré-condições:

1. Base de dados tem que estar operacional.
2. Tem que existir pelo menos uma tarefa.
3. Tem que existir pelo menos um operário.

Cenário principal de sucesso:

1. Obter todas as tarefas (UC6 – Obter Tarefas).
2. Obter todos os operários (UC3 – Obter Operários).
3. Atribuir tarefas aos operários.
4. Guardar atribuições.
5. Retorna tarefas.

Alternativas:

1. Não foi possível obter tarefas.
 - 1.a. Acaba o caso de uso.
2. Não foi possível obter operários.

- 2.a. Acaba o caso de uso.
- 3. Não foi possível atribuir tarefas aos operários.
 - 3.a. Acaba o caso de uso.
- 4. Não foi possível guardar atribuições.
 - 4.a. Acaba o caso de uso.

4.2.2 Requisitos não funcionais

Os requisitos não funcionais indicam as exigências de qualidade necessárias relativas às funcionalidades criadas. Para indicar estes requisitos foi utilizado o modelo FURPS (Baker and Kennett, 1999). FURPS é um acrónimo que representa um modelo utilizado para classificar e informar atributos de qualidade, identificado através das seguintes categorias:

- funcionalidade
- usabilidade
- fiabilidade
- desempenho
- suportabilidade

4.2.2.1 Funcionalidade

A parte Funcionalidade define as capacidades do programa, a generalidade das funções entregues e a segurança do sistema.

Este projeto não pode criar vulnerabilidades num sistema, e todos os casos de uso têm de ser cumpridos. O caso de uso “UC7 – Atribuir Tarefas aos Operários” tem que ser realizado de forma a permitir a fácil substituição do algoritmo.

4.2.2.2 Usabilidade

A parte Usabilidade considera a estética de um programa, a consistência e o fator humano.

Dado que este projeto é utilizado para explorar uma solução de forma analítica, o fator humano e a estética não vão ser considerados. No entanto, na criação dos diferentes algoritmos

presentes no caso de uso “UC7 – Atribuir Tarefas aos Operários”, é obrigatório enviar e receber a informação da mesma forma.

4.2.2.3 Fiabilidade

A parte fiabilidade avalia as falhas, a recuperabilidade, a previsibilidade e a veracidade dos resultados obtidos.

Para este projeto, os casos de usos não podem ter falhas e em condições iniciais iguais tem que obter resultados iguais. Os resultados devem ser contestados com tabelas realizadas manualmente.

4.2.2.4 Desempenho

A parte Desempenho supervisiona a velocidade, tempo de resposta, consumo de recursos, escalabilidade e a eficiência de um programa.

Este projeto é utilizado para avaliar os resultados obtidos pelas atribuições, logo a performance vai ser vista como uma parte não crítica.

4.2.2.5 Suportabilidade

A parte Suportabilidade indica a testabilidade, a flexibilidade, a compatibilidade e a configurabilidade de um programa.

O projeto necessita de suportar mudanças rápidas e fáceis de algoritmos para a realização do caso de uso “UC7 – Atribuir Tarefas aos Operários”

4.3 Arquitetura de Software

A arquitetura de software refere à estrutura e organização de um sistema informático. A arquitetura define como os componentes de um determinado software vão ser construídos, como eles se relacionam e como eles comunicam. A arquitetura serve como uma guia para o desenvolvimento de um software.

Este projeto só se apresenta na qualidade de complementar, ou seja, ele é realizado para complementar um determinado produto ou serviço. Tendo isto em conta, é necessário uma arquitetura relativamente conhecida e usada, versátil, fácil de adaptar, baixo impacto de desenvolvimento e com um bom nível de desacoplamento nos respectivos componentes. Micro-serviços e *Model-View-Controller* (MVC) cumprem estes requerimentos, no entanto, para facilidade de desenvolvimento, a mesma base de dados vai ser utilizada para múltiplos serviços e vai ser promovida a reutilização de código, resultando na preferência da arquitetura *Model-View-Controller*(Fowler, 2015). A arquitetura *Model-View-Controller* normalmente decompõe-se nos três componentes que o próprio nome indica, no entanto, foi adicionado mais um para facilitar a adaptação da arquitetura ao problema, exemplificado pela Figura 1. Os componentes são:

- *Model*
- *View*
- *Controller*
- *Algorithms*

O componente *Model* representa a lógica da informação do negócio.

O componente *View* indica como a informação é apresentada.

O componente *Controller* indica toda a lógica do negócio e as interações possíveis.

Finalmente, o componente *Algorithms* serve de apoio ao *Controller* encapsulando e separando a lógica do negócio da forma como os algoritmos se comportam.

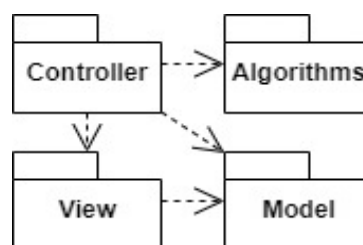


Figura 8: Vista de implementação com granularidade de aplicação

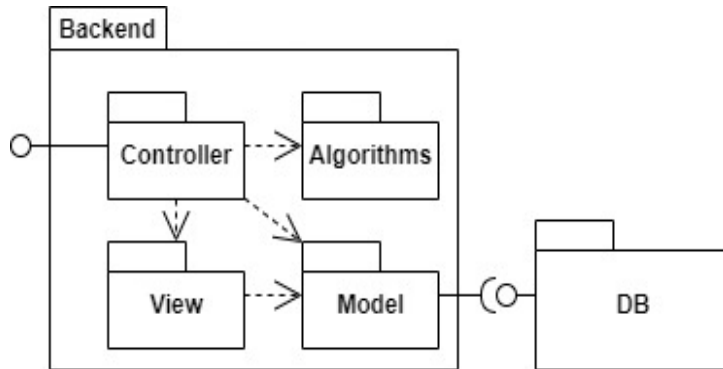


Figura 9: Vista de lógica com granularidade de aplicação

Devido à característica complementar deste projeto e possibilidade da incorporação deste projeto noutros, não vão ser apresentadas vista de implementação nem vista lógica com granularidade de sistema.

4.4 Modelo de Dados

O modelo de dados demonstra a estrutura lógica de uma base de dados e indica como os dados vão ser guardados, geridos e processado dentro da base de dados. A Figura 10 representa o modelo de dados para este projeto.

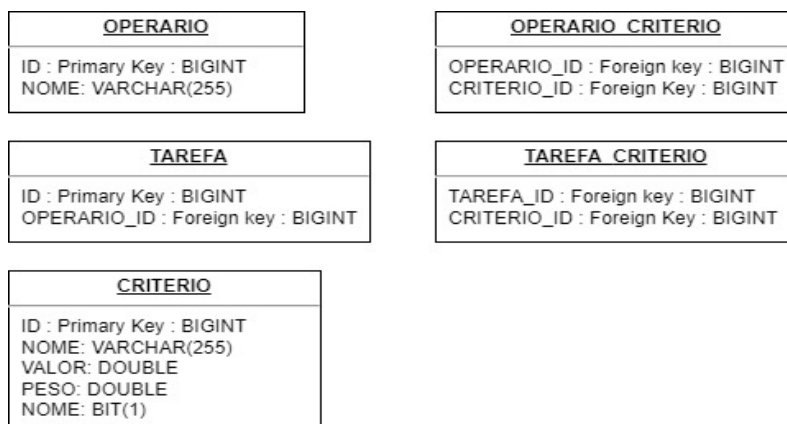


Figura 10: Modelo de dados

Podemos observar que as tabelas OPERARIO, TAREFA e CRITERIO seguem o modelo de domínio com a exceção do parâmetro Critérios na tabela TAREFA e OPERARIO. Deste modo, todos os operários criados devem ser guardados na tabela OPERARIO, todas as tarefas criadas devem ser guardadas na tabela TAREFA, e todos os critérios criados são guardados na tabela CRITERIO, independentemente se são dos operários ou das tarefas

Dado que uma tarefa e um operário tem uma lista de critérios, foram criadas as tabelas OPERARIO_CRITERIO e TAREFA_CRITERIO para registrar as respectivas relações entre eles.

4.5 Desenho

O desenho é um processo que planeia ou converte os requerimentos de software em funções, objetos e/ou métodos que consigam alcançar os objetivos pretendidos.

Para realizar o desenho do sistema, são utilizados os casos de uso referidos anteriormente para dividir o mesmo, onde cada caso de uso é representado através de um diagrama de sistema. Diagramas de sistema permitem modular o sistema e especificar os métodos e funções necessárias para este atingir um determinado objetivo. Um diagrama de sistema é uma representação visual que indica o fluxo do sistema.

4.5.1 UC1 – Criar Operário

O desenho deste caso de uso está representado na Figura 11. É de notar a necessidade de introduzir os parâmetros necessários para a criação de um operário.

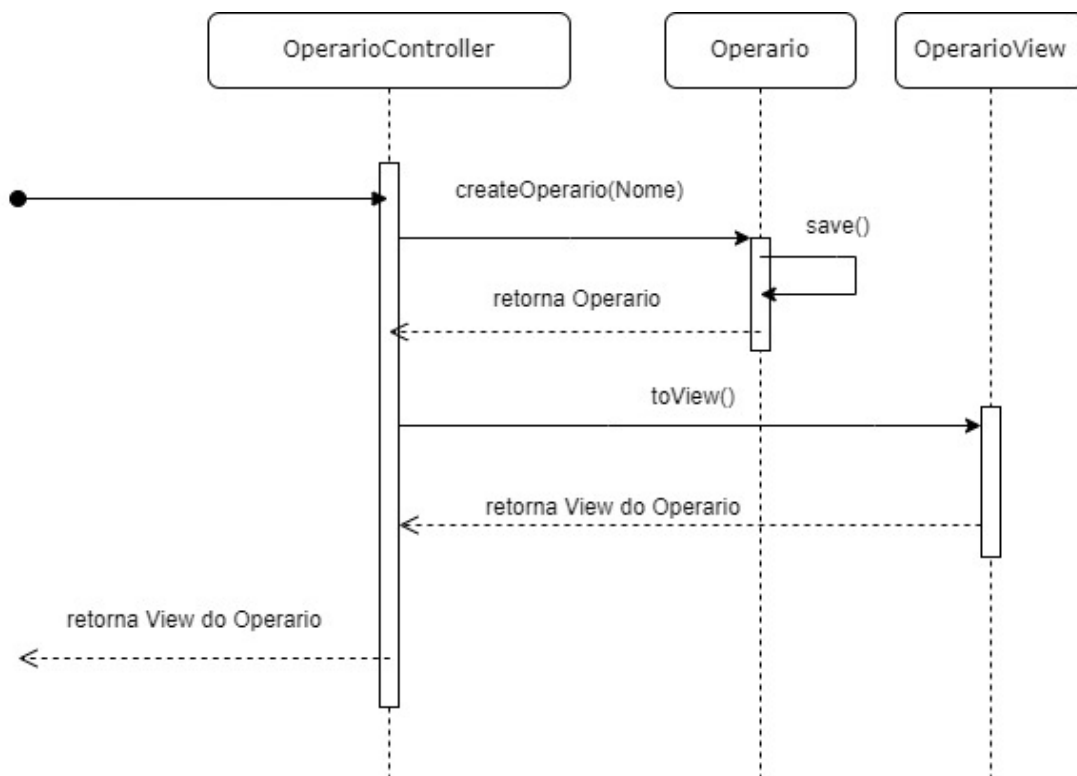


Figura 11: Diagrama de sequência do UC1

4.5.2 UC2 – Adicionar Critérios ao Operário

O desenho deste caso de uso está representado na Figura 12. É de notar a necessidade de introduzir os parâmetros necessários para a criação de um Critério.

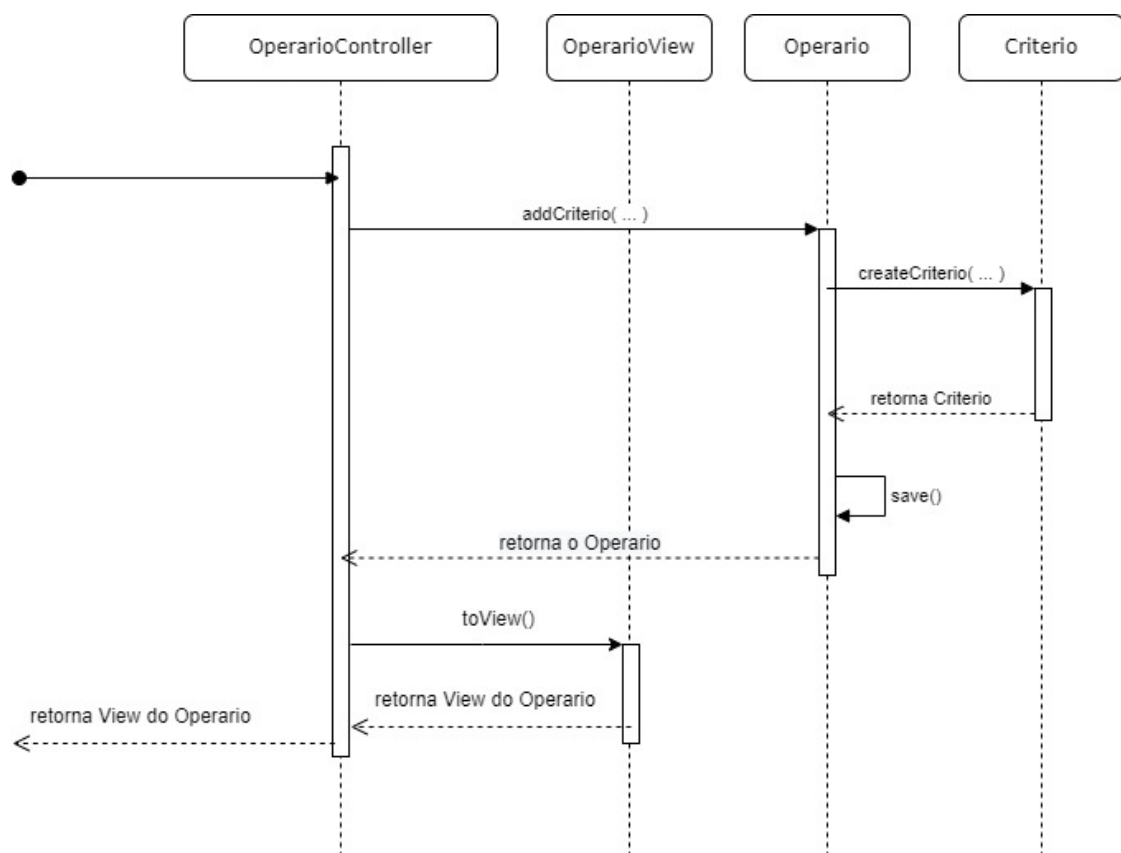


Figura 12: Diagrama de sequência do UC2

4.5.3 UC3 – Obter Operários

O desenho deste caso de uso está representado na Figura 13.

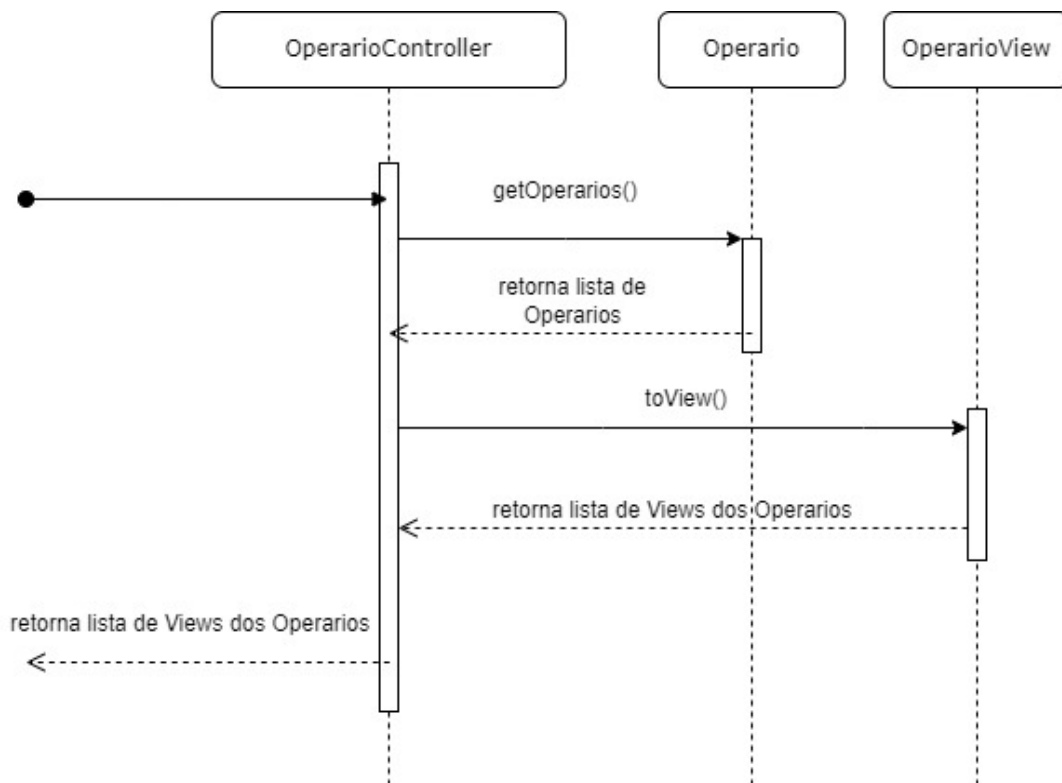


Figura 13: Diagrama de seqüência do UC3

4.5.4 UC4 – Criar Tarefa

O desenho deste caso de uso está representado na Figura 14.

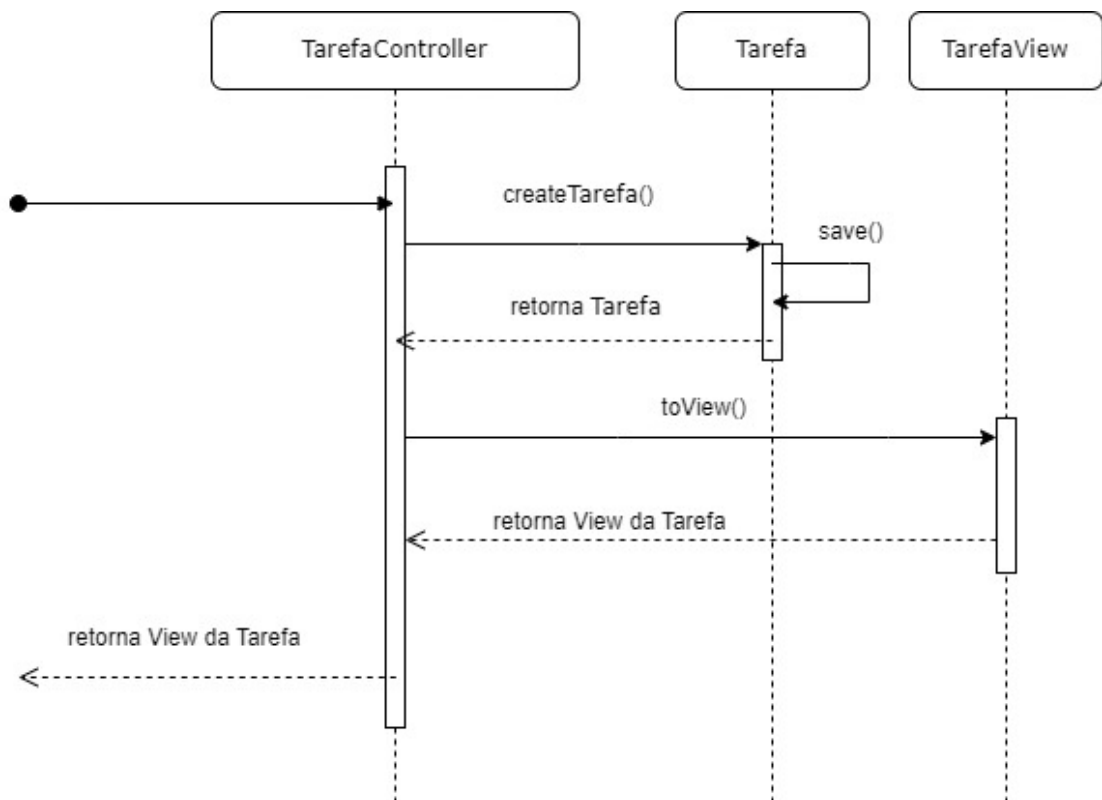


Figura 14: Diagrama de sequência do UC4

4.5.5 UC5 – Adicionar Critérios à Tarefa

O desenho deste caso de uso está representado na Figura 15. É de notar a necessidade de introduzir os parâmetros necessários para a criação de um Critério.

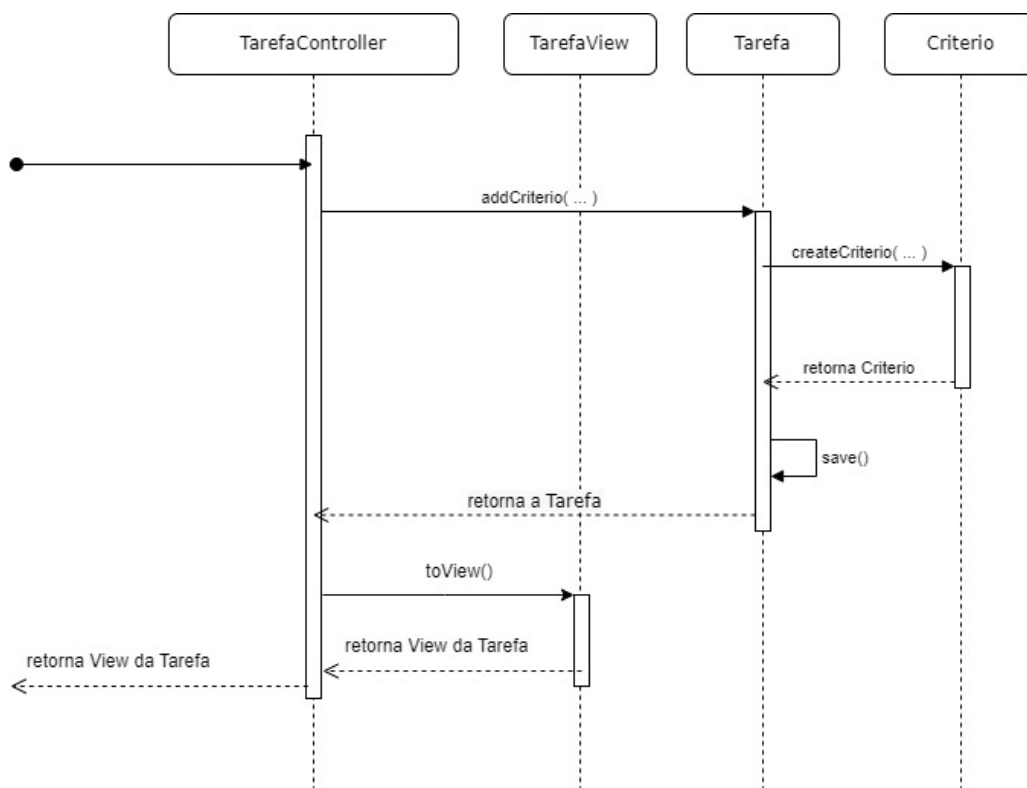


Figura 15: Diagrama de seqüência do UC5

4.5.6 UC6 – Obter Tarefas

O desenho deste caso de uso está representado na Figura 16.

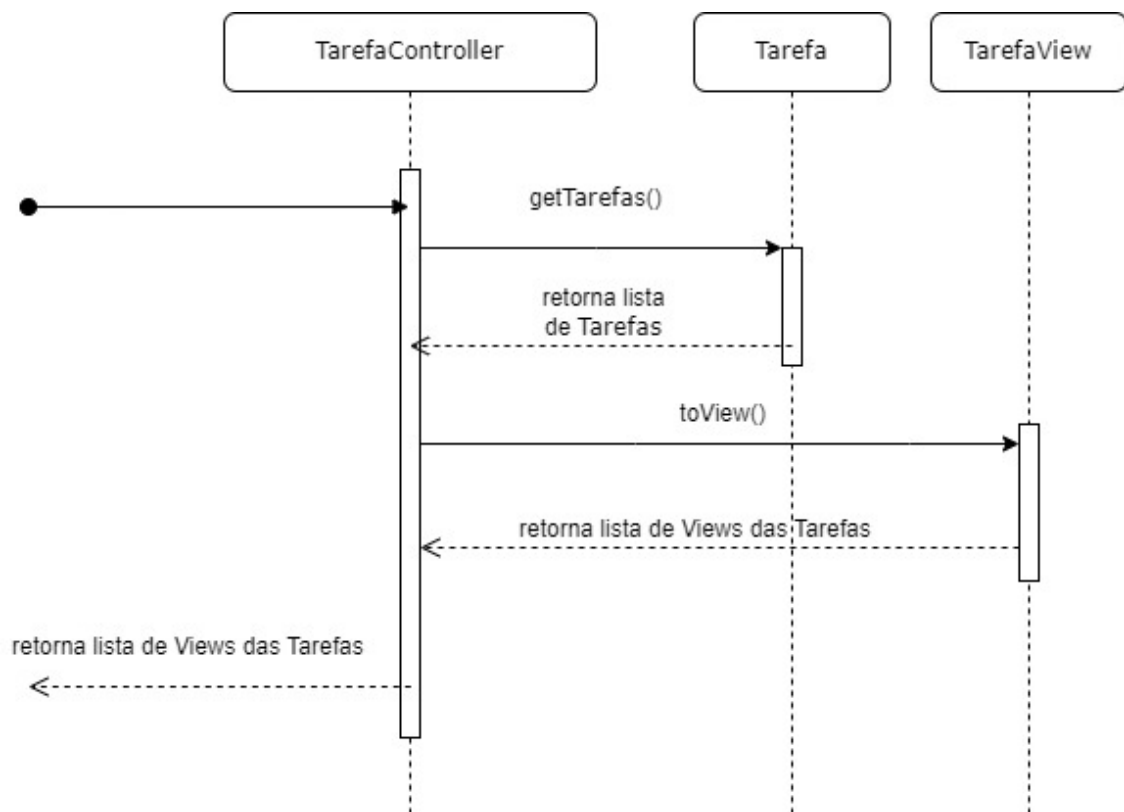


Figura 16: Diagrama de seqüência do UC6

4.5.7 UC7 – Atribuir Tarefas aos Operários

O desenho deste caso de uso está representado na Figura 17. É de notar que este caso de uso utiliza os casos de uso “UC3 – Obter Operários ” e “UC6 – Obter Tarefas” para obter as tarefas e os operários. Algorithms é uma interface onde é aplicada um *strategy pattern* para facilitar a mudança de algoritmos e encapsular cada um deles, facilitando o desenvolvimento e o debugging. O *strategy pattern* é um padrão de desenho que permite definir uma família de algoritmos e por os algoritmos em classes separadas (Refactoring.guru, 2014). Os algoritmos utilizados são explicados na implementação.

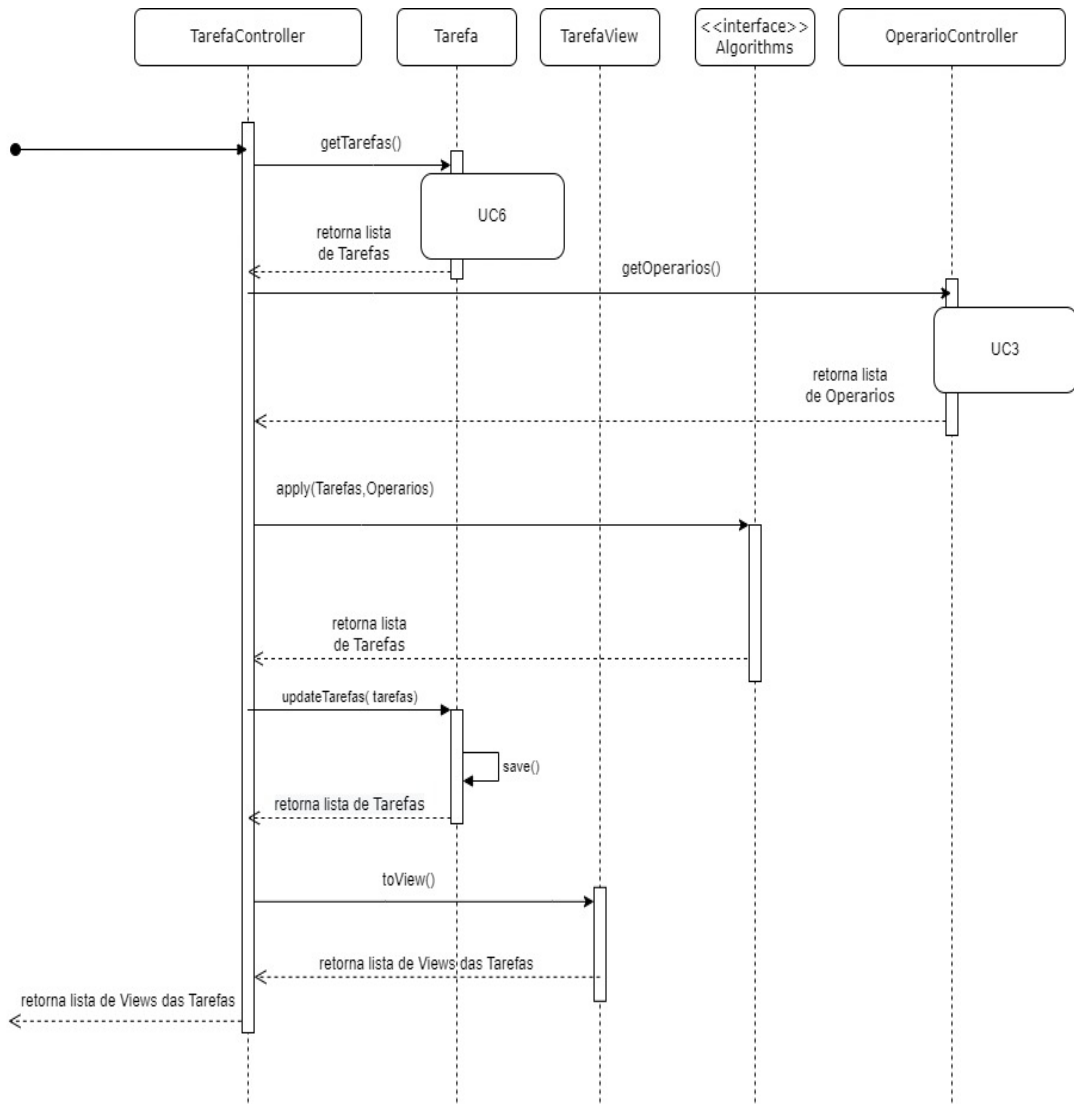


Figura 17: Diagrama de seqüência do UC7

4.6 Conclusão

Neste capítulo, analisado todos os requisitos funcionais e não funcionais necessários para atingir o objetivo pretendido. Foi verificado que é necessário sete casos de uso e o caso de uso “UC7 – Atribuir Tarefas aos Operários” é responsável por atingir a solução pretendida. Os restantes casos de uso apoiam o caso de uso “UC7 – Atribuir Tarefas aos Operários”. Também é decidido a utilização de uma arquitetura MVC, é definido o modelo de dados e é realizado o desenho de todos os casos de uso, onde é aplicado um strategy pattern no caso de uso “UC7 – Atribuir Tarefas aos Operários” para responder aos requisitos mencionados.

5 Implementação da solução

Esta secção indica como o desenho e a análise se traduzem na implementação. Também é referido como os algoritmos foram construídos, a linha de pensamento deles e ainda que testes foram realizados. Esta secção é dividida de acordo com os casos de uso e o caso de uso “UC7 – Atribuir Tarefas aos Operários” é subdividido de acordo com as diferentes alternativas.

Na implementação deste projeto aplicou-se as práticas de *single responsibility, open-closed, Liskov substitution, interface segregation* e *dependency inversion* (SOLID) e *General Responsibility Assignment Software Patterns* (GRASP) quando possível. Os princípios do SOLID são descritos pelo próprio acrónimo. E o GRASP é constituído por princípios como *information expert, controller, high cohesion, low coupling, polymorphism, creator, protected variation* e *indirection*.

Os casos de uso descritos anteriormente foram realizados de forma a facilitar a integração deste projeto com outros já existentes com o mínimo de esforço, fazendo com que este sistema possa existir independentemente do restante sistema.

Devido à complexidade e ao foco deste projeto, o caso de uso “UC7 – Atribuir Tarefas aos Operários” é mais detalhado que os restantes. No entanto, os casos de uso, com a exceção do

caso de uso “UC7 – Atribuir Tarefas aos Operários”, correspondem aos passos necessários para preparar e criar condições aceitáveis para o estudo deste projeto.

Nos processos de criação/gravar na base de dados, a base de dados é responsável por criar um identificador único para cada operário, tarefa ou critério.

5.1 UC1 – Criar Operário

Ao chamar o serviço associado a este caso de uso é necessário indicar o nome do operário. Depois é criado o operário com o respetivo nome e gravado na base de dados.

5.2 UC2 – Adicionar Critérios ao Operário

Ao chamar o serviço associado a este caso de uso é necessário indicar o operário, o nome, o valor e o peso do critério e informar se o critério em questão é melhor quanto maior ou menor for o valor. De seguida é criado o critério com as informações recebidas, com a exceção do operário, e gravado na base de dados. Finalmente o critério é associado ao operário indicado e a relação também é gravada na base de dados.

5.3 UC3 – Obter Operários

Ao chamar o serviço associado a este caso de uso é realizada uma busca na base de dados para obter todos os operários e os respetivos critérios.

5.4 UC4 – Criar Tarefa

Ao chamar este caso de uso a tarefa é imediatamente criada. Esta é depois gravada na base de dados.

5.5 UC5 – Adicionar Critérios à Tarefa

Este caso de uso é muito semelhante ao caso de uso “UC2 – Adicionar Critérios ao Operário”, mas é indicada e associada uma tarefa em vez de um operário.

5.6 UC6 – Obter Tarefas

Ao chamar o serviço associado a este caso de uso é realizada uma busca na base de dados para obter todas as tarefas e os respectivos critérios.

5.7 UC7 – Atribuir Tarefas aos Operários

Ao chamar o serviço associado a este caso de uso, as tarefas e os operários são obtidos através dos casos de uso “UC6 – Obter Tarefas” e “UC3 – Obter Operários”. Depois é utilizado um algoritmo para realizar a atribuição das tarefas aos operários seguido da gravação da atribuição na base de dados retornando as tarefas com as respectivas associações aos operários.

Para alcançar um algoritmo que seja capaz de satisfazer as condições necessárias foi decomposto o problema nas seguintes partes:

1. Identificar o que diferencia cada tarefa e cada operário.
2. Identificar a posição de uma tarefa em relação às restantes e um operário em relação aos restantes, ou seja, classificá-los numa lista de *rankings* de acordo com os critérios anteriormente identificados. Deste modo, é possível indicar quais as tarefas mais críticas a serem realizadas e quais os operários potencialmente mais aptos para as realizar.
3. Atribuir as tarefas aos operários de forma equilibrada.

A solução para a primeira parte é obtida através do desenho da solução. Nele foi adicionado o conceito *Criterio*. Tal como foi dito anteriormente, através dos critérios é possível diferenciar

tarefas e operários. Todas as tarefas têm o mesmo tipo de critérios e todos os operários têm o mesmo tipo de critérios.

A solução da segunda parte começa pela transformação das tarefas e os respectivos critérios numa matriz e a transformação dos operários e os respectivos critérios noutra matriz. Desta forma podemos aplicar um algoritmo de decisão multi-critério para realizar a classificação das tarefas e dos operários.

A última parte pode ser solucionada através de um algoritmo de escalonamento.

As soluções anteriores permitem realizar uma ideia para o algoritmo, no entanto, existem múltiplos algoritmos de decisão multi-critério e de escalonamento que podem influenciar os resultados. Tendo isto em conta foram realizadas múltiplas soluções, variando somente os algoritmos de decisão multi-critério, e gerados dados para as tarefas e para os operários, com o objetivo de comparar as soluções em questão. Foram criadas quatro versões do algoritmo e vai ser explicado como foram alcançadas.

5.7.1 Primeira parte da solução do Algoritmo

Tal como referido, foram geradas tarefas e operários e através dos critérios foram traduzidos para matrizes. As matrizes são apresentadas sob a forma de tabela para facilitar a representação das mesmas, onde cada linha representa uma tarefa e cada coluna com exceção da primeira representam critérios.

As tarefas geradas são representadas pela Tabela 2.

Tabela 2: Tarefas geradas

Tarefa	Prioridade	Complexidade
tarefa01	91	3
tarefa02	43	2
tarefa03	80	3
tarefa04	58	0
tarefa05	88	2
tarefa06	67	4
tarefa07	93	2
tarefa08	40	2
tarefa09	31	2
tarefa10	24	1

A coluna Tarefa representa um identificador único para cada tarefa.

A coluna Prioridade informa a urgência de realizar uma determinada tarefa. Esta coluna varia de 0 a 100. Quanto mais elevado for o valor Prioridade mais urgente é a tarefa. O peso deste critério é de 70% e é um critério benéfico.

A coluna Complexidade indica o quão complicado é realizar uma determinada tarefa. Esta coluna varia de 0 a 5. Quanto mais elevado for o valor Complexidade mais difícil é a tarefa. O peso deste critério é de 30% e é um critério benéfico.

Os operários gerados são representados pela Tabela 3.

Tabela 3: Operários gerados

Operário	Rácio Temporal	Quantidade de tarefas	Qualidade	Rácio de tarefas completadas	Nível de experiência
Operário01	1.11	153	0.03	0.8	1
Operário02	1.16	162	0.26	0.78	4
Operário03	0.96	158	0.33	0.34	2
Operário04	1.15	161	0.36	0.27	0
Operário05	1.09	154	0.5	0	4

A coluna Operário representa um identificador único para cada operário.

A coluna Rácio Temporal indica o tempo médio de cada tarefa completada por mês. Quanto mais elevado for o valor Rácio Temporal mais tempo demora o operário a realizar uma tarefa. O peso deste critério é de 30% e é um critério prejudicial.

A coluna Quantidade de tarefas é a quantidade de tarefas completadas por mês. Esta coluna varia de 152 a 164. Quanto mais elevado for o valor Quantidade maior é o número de tarefas completadas pelo operário. O peso deste critério é de 30% e é um critério benéfico.

A coluna Qualidade informa a quantidade de tarefas que não foram reabertas sob a quantidade de tarefas completadas. Quanto mais elevado for o valor Qualidade maior é a qualidade do trabalho realizado pelo operário. O peso deste critério é de 10% e é um critério benéfico.

A coluna Rácio de tarefas completadas indica as tarefas completas fora do tempo limite sob a quantidade de tarefas completadas. Quanto mais elevado for o valor Rácio de tarefas completadas maior é o tempo ocupado por cada tarefa pelo operário. O peso deste critério é de 25% e é um critério prejudicial.

A coluna Nível de experiência representa o nível de experiência. Quanto mais elevado for o valor maior é a experiência do operário. O peso deste critério é de 5% e é um critério benéfico.

5.7.2 Segunda parte da solução do Algoritmo

É nesta parte onde existe variedade na solução, devido à utilização de algoritmos de decisão multi-critério. Existem múltiplos algoritmos desta categoria e podem ter finalidades e/ou necessidade de condições iniciais diferentes. Como condição final está identificada a necessidade de ordenar as tarefas e os operários de acordo com os critérios e a condição inicial é que cada critério tem um peso associado (Wątróbski et al., 2019). Deste modo é possível diminuir significativamente a quantidade de algoritmos de decisão multi-critério para a seguinte lista:

- TOPSIS
- VIKOR
- SMART
- SAW
- PROMETHEE II
- MAUT
- MAVT
- etc.

Desta lista foram selecionados quatro algoritmos para o estudo:

- MAUT
- PROMETHEE II
- TOPSIS
- VIKOR

MAUT foi selecionado por ser um dos algoritmos mais simples existentes e por outros algoritmos se basearem nele (Shanmuganathan et al., 2018).

PROMETHEE II foi escolhido por ter uma abordagem relativamente única e complexa em relação aos restantes algoritmos (Brans and Mareschal, 2020).

TOPSIS foi eleito devido à aparente popularidade e presença deste (Yadav et al., 2019).

VIKOR foi designado por o método ser próximo do MAUT (Opricovic and Tzeng, 2007).

Embora estes algoritmos sejam diferentes, eles necessitam de utilizar os dados na mesma escala/unidades. Para este efeito, o primeiro passo em todos eles é a normalização dos valores que resulta na Tabela 4 para as tarefas e a Tabela 5 para os operários. No entanto, estas tabelas não se aplicam ao algoritmo TOPSIS, devido a este utilizar um cálculo diferente para normalização. A normalização é aplicada a cada coluna devido às escalas serem dependentes de cada critério. Se o critério em questão for benéfico é utilizada a fórmula 1, caso contrário é utilizada a fórmula 2. Para todas as fórmulas a letra i é utilizada para referir às linhas da tabela e a letra j é utilizada para referir às colunas da tabela.

$$f(x) = \frac{j_x - \text{Min}(j)}{\text{Max}(j) - \text{Min}(j)} \quad (1)$$

$$f(x) = \frac{\text{Max}(j) - j_x}{\text{Max}(j) - \text{Min}(j)} \quad (2)$$

Tabela 4: Tarefas normalizadas

Tarefa	Prioridade	Complexidade
tarefa01	0.97	0.75
tarefa02	0.28	0.5
tarefa03	0.81	0.75
tarefa04	0.49	0
tarefa05	0.93	0.5
tarefa06	0.62	1
tarefa07	1	0.5
tarefa08	0.23	0.5
tarefa09	0.1	0.5
tarefa10	0	0.25

Tabela 5: Operários normalizados

Operário	Rácio Temporal	Quantidade de tarefas	Qualidade	Rácio de tarefas completadas	Nível de experiência
Operário01	0.25	0	0	0	0.25
Operário02	0	1	0.49	0.03	1
Operário03	1	0.56	0.64	0.58	0.5
Operário04	0.05	0.89	0.7	0.66	0
Operário05	0.35	0.11	1	1	1

De seguida vão ser apresentados todos os passos e os resultados finais da aplicação de cada um dos algoritmos.

5.7.2.1 MAUT

Após a normalização é necessário aplicar os pesos de cada critério à matriz através de multiplicação resultando nas tabelas 6 e 7 (Shanmuganathan et al., 2018).

Tabela 6: Tarefas normalizadas com os pesos aplicados

Tarefa	Prioridade	Complexidade
tarefa01	0.68	0.23
tarefa02	0.19	0.15
tarefa03	0.57	0.23
tarefa04	0.34	0
tarefa05	0.65	0.15
tarefa06	0.44	0.3
tarefa07	0.7	0.15
tarefa08	0.16	0.15
tarefa09	0.07	0.15
tarefa10	0	0.08

Tabela 7: Operários normalizados com os pesos aplicados

Operário	Rácio Temporal	Quantidade de tarefas	Qualidade	Rácio de tarefas completadas	Nível de experiência
Operário01	0.08	0	0	0	0.01
Operário02	0	0.3	0.05	0.01	0.05
Operário03	0.3	0.17	0.06	0.14	0.03
Operário04	0.02	0.27	0.07	0.17	0
Operário05	0.11	0.03	0.1	0.25	0.05

Finalmente, para obter o resultado MAUT cada linha é somada e quanto maior for a soma, mais relevante é a tarefa ou o operário. Esta soma resulta nas tabelas 8 e 9, que já se apresentam ordenadas de acordo com a soma (Shanmuganathan et al., 2018).

Tabela 8: Resultado MAUT das tarefas

Tarefa	MAUT	Classificação
tarefa01	0.9	1º
tarefa07	0.85	2º
tarefa05	0.8	3º
tarefa03	0.79	4º
tarefa06	0.74	5º
tarefa04	0.34	6º
tarefa02	0.34	7º
tarefa08	0.31	8º
tarefa09	0.22	9º
tarefa10	0.08	10º

Tabela 9: Resultado MAUT dos operários

Operário	MAUT	Classificação
Operário03	0.7	1º
Operário05	0.54	2º
Operário04	0.52	3º
Operário02	0.41	4º
Operário01	0.09	5º

5.7.2.2 PROMETHEE II

Após a normalização é calculado a diferença avaliativa de cada alternativa em relação às restantes, ou seja, subtrair uma linha a outra linha de acordo com cada critério para cada combinação possível de linhas (Brans and Mareschal, 2020). Isto resulta nas tabelas 10 e 33 (a Tabela 33 está presente nos anexos devido ao tamanho). A primeira coluna destas tabelas representa a relação entre cada linha (exemplo: Operário01 – Operário02 representa a relação Operário01 para Operário02 e Operário02 – Operário01 representa a relação Operário02 para Operário01). A fórmula 3 representa o cálculo efetuado, onde a letra m representa uma linha sempre diferente de i .

$$f(x_{ij}) = x_{ij} - x_{mj} \quad (3)$$

Tabela 10: Diferença avaliativa dos operários

Operário	Rácio Temporal	Quantidade de tarefas	Qualidade	Rácio de tarefas completadas	Nível de experiência
Operário01 -Operário02	0.24	-1	-0.48	-0.03	-0.75
Operário01 -Operário03	-0.76	-0.56	-0.63	-0.58	-0.25

Operário	Rácio Temporal	Quantidade de tarefas	Qualidade	Rácio de tarefas completadas	Nível de experiência
Operário01 -Operário04	0.2	-0.89	-0.7	-0.67	0.25
Operário01 -Operário05	-0.11	-0.11	-1	-1	-0.75
Operário02 -Operário01	-0.24	1	0.48	0.03	0.75
Operário02 -Operário03	-1	0.44	-0.15	-0.55	0.5
Operário02 -Operário04	-0.03	0.11	-0.22	-0.64	1
Operário02 -Operário05	-0.35	0.89	-0.52	-0.97	0
Operário03 -Operário01	0.76	0.56	0.63	0.58	0.25
Operário03 -Operário02	1	-0.44	0.15	0.55	-0.5
Operário03 -Operário04	0.97	-0.33	-0.07	-0.09	0.5
Operário03 -Operário05	0.65	0.44	-0.37	-0.42	-0.5
Operário04 -Operário01	-0.2	0.89	0.7	0.67	-0.25
Operário04 -Operário02	0.03	-0.11	0.22	0.64	-1

Operário	Rácio Temporal	Quantidade de tarefas	Qualidade	Rácio de tarefas completadas	Nível de experiência
Operário04 -Operário03	-0.97	0.33	0.07	0.09	-0.5
Operário04 -Operário05	-0.32	0.78	-0.3	-0.33	-1
Operário05 -Operário01	0.11	0.11	1	1	0.75
Operário05 -Operário02	0.35	-0.89	0.52	0.97	0
Operário05 -Operário03	-0.65	-0.44	0.37	0.42	0.5
Operário05 -Operário04	0.32	-0.78	0.3	0.33	1

A seguir é aplicada a função de preferência, onde se um elemento das tabelas 10 e 33 for inferior a 0 ele é substituído por 0 (Brans and Mareschal, 2020), resultando nas tabelas 11 e 34 (a Tabela 34 está presente nos anexos devido ao tamanho).

Tabela 11: Preferências dos operários

Operário	Rácio Temporal	Quantidade de tarefas	Qualidade	Rácio de tarefas completadas	Nível de experiência
Operário01 - Operário02	0.24	0	0	0	0

Operário	Rácio Temporal	Quantidade de tarefas	Qualidade	Rácio de tarefas completadas	Nível de experiência
Operário01 - Operário03	0	0	0	0	0
Operário01 - Operário04	0.2	0	0	0	0.25
Operário01 - Operário05	0	0	0	0	0
Operário02 - Operário01	0	1	0.48	0.03	0.75
Operário02 - Operário03	0	0.44	0	0	0.5
Operário02 - Operário04	0	0.11	0	0	1
Operário02 - Operário05	0	0.89	0	0	0
Operário03 - Operário01	0.76	0.56	0.63	0.58	0.25
Operário03 - Operário02	1	0	0.15	0.55	0
Operário03 - Operário04	0.97	0	0	0	0.5
Operário03 - Operário05	0.65	0.44	0	0	0
Operário04 - Operário01	0	0.89	0.7	0.67	0

Operário	Rácio Temporal	Quantidade de tarefas	Qualidade	Rácio de tarefas completadas	Nível de experiência
Operário04 - Operário02	0.03	0	0.22	0.64	0
Operário04 - Operário03	0	0.33	0.07	0.09	0
Operário04 - Operário05	0	0.78	0	0	0
Operário05 - Operário01	0.11	0.11	1	1	0.75
Operário05 - Operário02	0.35	0	0.52	0.97	0
Operário05 - Operário03	0	0	0.37	0.42	0.5
Operário05 - Operário04	0.32	0	0.3	0.33	1

Depois é calculado o agregado das preferências e colocadas numa matriz de relação, ou seja, uma matriz que permita ver como os operários se relacionam com outros operários e outra matriz com o mesmo efeito, mas para as tarefas. O agregado das preferências é calculado através da soma de todos os critérios, sendo que cada um dos valores dos critérios é multiplicado pelo respetivo peso associado (Brans and Mareschal, 2020). Isto culmina nas tabelas 12 e 13.

Tabela 12: Matriz agregado de preferência das tarefas

	tare- fa01	tare- fa02	tare- fa03	tare- fa04	tare- fa05	tare- fa06	tare- fa07	tare- fa08	tare- fa09	tare- fa10
tare- fa01	-	0.56	0.11	0.56	0.11	0.24	0.08	0.59	0.68	0.83
tare- fa02	0	-	0	0.15	0	0	0	0.03	0.12	0.27
tare- fa03	0	0.45	-	0.45	0.08	0.13	0.08	0.48	0.57	0.72
tare- fa04	0	0.15	0	-	0	0	0	0.18	0.27	0.34
tare- fa05	0	0.46	0.08	0.45	-	0.21	0	0.49	0.58	0.72
tare- fa06	0.08	0.39	0.08	0.39	0.15	-	J139	0.42	0.52	0.66
tare- fa07	0.02	0.51	0.13	0.51	0.05	0.26	-	0.54	0.63	0.78
tare- fa08	0	0	0	0.15	0	0	0	-	0.09	0.24
tare- fa09	0	0	0	0.15	0	0	0	0	-	0.15
tare- fa10	0	0	0	0.08	0	0	0	0	0	-

Tabela 13: Matriz agregado de preferência dos operários

	Operário01	Operário02	Operário03	Operário04	Operário05
Operário01	-	0.07	0	0.07	0
Operário02	0.39	-	0.16	0.08	0.27
Operário03	0.62	0.45	-	0.31	0.33
Operário04	0.5	0.19	0.13	-	0.23
Operário05	0.46	0.4	0.17	0.26	-

Após as tabelas de agregado de preferência é necessário calcular o fluxo de entrada e o fluxo de saída. O fluxo de entrada é obtido através da média da soma dos valores de uma coluna e o fluxo de saída é alcançado através da média da soma dos valores de uma linha (Brans and Mareschal, 2020). Isto resulta nas tabelas 14 e 15.

Tabela 14: Fluxos das tarefas

Tarefa	Fluxo de entrada	Fluxo de saída
tarefa01	0.01	0.42
tarefa02	0.28	0.06
tarefa03	0.04	0.33
tarefa04	0.32	0.11
tarefa05	0.04	0.33
tarefa06	0.09	0.3
tarefa07	0.02	0.38
tarefa08	0.3	0.05
tarefa09	0.39	0.03
tarefa10	0.52	0.01

Tabela 15: Fluxos dos operários

Operário	Fluxo de entrada	Fluxo de saída
Operário01	0.49	0.04
Operário02	0.28	0.23
Operário03	0.11	0.43
Operário04	0.18	0.27
Operário05	0.21	0.32

Finalmente, podemos obter o rendimento do fluxo, que é usado para classificar cada alternativa, resultando nas tabelas 16 e 17. O rendimento do fluxo é calculado subtraindo o fluxo de entrada ao fluxo de saída. Quanto melhor for o rendimento do fluxo mais relevante é a linha (Brans and Mareschal, 2020).

Tabela 16: Resultado PROMETHEE II das tarefas

Tarefa	Rendimento do fluxo	Classificação
tarefa01	0.41	1º
tarefa07	0.36	2º
tarefa05	0.29	3º
tarefa03	0.28	4º
tarefa06	0.2	5º
tarefa04	-0.21	6º
tarefa02	-0.22	7º
tarefa08	-0.25	8º
tarefa09	-0.35	9º
Tarefa10	-0.51	10º

Tabela 17: Resultado PROMETHEE II dos operários

Operário	Rendimento do fluxo	Classificação
Operário03	0.31	1º
Operário05	0.11	2º
Operário04	0.08	3º
Operário02	-0.05	4º
Operário01	-0.46	5º

5.7.2.3 TOPSIS

Na normalização das tabelas para o algoritmo TOPSIS é utilizada a fórmula 4. As tabelas 18 e 19 resultam da normalização com a fórmula referida multiplicada pelos pesos de cada critério

correspondente (Yadav et al., 2019). Na fórmula referida, z representa o número total de linhas numa tabela.

$$f(x) = \frac{x_{ij}}{\sqrt{\sum_{k=1}^z x_{kj}^2}} \quad (4)$$

Tabela 18: Normalização TOPSIS das tarefas multiplicadas pelo respetivo peso de cada critério

Tarefa	Prioridade	Complexidade
tarefa01	0.3	0.12
tarefa02	0.14	0.08
tarefa03	0.27	0.12
tarefa04	0.19	0
tarefa05	0.29	0.08
tarefa06	0.22	0.16
tarefa07	0.31	0.08
tarefa08	0.13	0.08
tarefa09	0.1	0.08
tarefa10	0.08	0.04

Tabela 19: Normalização TOPSIS dos operários multiplicadas pelo respetivo peso de cada critério

Operário	Rácio Temporal	Quantidade de tarefas	Qualidade	Rácio de tarefas completadas	Nível de experiência
Operário01	0.14	0.13	0	0.17	0.01
Operário02	0.14	0.14	0.03	0.16	0.03
Operário03	0.12	0.13	0.04	0.07	0.02
Operário04	0.14	0.14	0.05	0.06	0
Operário05	0.13	0.13	0.07	0	0.03

Depois é calculada a distância euclidiana à melhor hipótese e à pior hipótese. A distância euclidiana permite indicar a distância entre dois pontos e é representada pela fórmula 5. A melhor hipótese é obtida através dos máximos de cada coluna, se a coluna representar um critério benéfico. Se o critério for prejudicial, é procurado o mínimo da coluna. A pior hipótese é o reverso da melhor hipótese, ou seja, o máximo é substituído pelo mínimo e o mínimo é trocado pelo máximo. Na fórmula 5, h representa uma das hipóteses referidas e w é utilizado para representar o número máximo de colunas existentes (Yadav et al., 2019).

$$f(x) = \sqrt{\sum_{k=1}^w (x_{ik} - h_{ik})^2} \quad (5)$$

Tabela 20: Distâncias às hipóteses das tarefas

Tarefa	Distância à melhor hipótese	Distância à pior hipótese
tarefa01	0.04	0.25
tarefa02	0.19	0.1
tarefa03	0.06	0.22
tarefa04	0.2	0.11
tarefa05	0.08	0.23
tarefa06	0.09	0.22
tarefa07	0.08	0.24
tarefa08	0.19	0.1
tarefa09	0.22	0.08
tarefa10	0.26	0.04

Tabela 21: Distâncias às hipóteses dos operários

Operário	Distância à melhor hipótese	Distância à pior hipótese
Operário01	0.18	0.01
Operário02	0.17	0.05
Operário03	0.08	0.11
Operário04	0.07	0.12
Operário05	0.02	0.18

Finalmente, é calculado o valor TOPSIS que permite classificar as tarefas e os operários. O valor TOPSIS é obtido através da fórmula 6, onde b representa a distância à melhor hipótese e p a distância à pior hipótese (Yadav et al., 2019).

$$f(x) = \frac{p}{b+p} \quad (6)$$

Tabela 22: Resultado TOPSIS das tarefas

Tarefa	TOPSIS	Classificação
tarefa01	0.86	1º
tarefa03	0.79	2º
tarefa07	0.75	3º
tarefa05	0.73	4º
tarefa06	0.71	5º
tarefa04	0.36	6º
tarefa02	0.36	7º
tarefa08	0.33	8º
tarefa09	0.27	9º
tarefa10	0.13	10º

Tabela 23: Resultado TOPSIS dos operários

Operário	TOPSIS	Classificação
Operário05	0.91	1º
Operário04	0.63	2º
Operário03	0.59	3º
Operário02	0.21	4º
Operário01	0.05	5º

5.7.2.4 VIKOR

Após a normalização da tabela, ela é multiplicada pelos pesos de cada critério, seguido pelo cálculo da medida de utilidade e da medida de arrependimento. A medida de utilidade é obtida através da soma de todos os critérios. Isto resulta na medida de utilidade e no resultado MAUT representado pelas tabelas 8 e 9 serem iguais. A medida de arrependimento resulta da obtenção do maior valor de cada linha (Opricovic and Tzeng, 2007). As tabelas 24 e 25 representam o resultado destes cálculos.

Tabela 24: Medidas das tarefas

Tarefa	Medida de utilidade	Medida de arrependimento
tarefa01	0.9	0.68
tarefa03	0.34	0.19
tarefa07	0.79	0.57
tarefa05	0.34	0.34
tarefa06	0.8	0.65
tarefa04	0.74	0.44
tarefa02	0.85	0.7
tarefa08	0.31	0.16
tarefa09	0.22	0.15
tarefa10	0.08	0.08

Tabela 25: Medidas dos operários

Operário	Medida de utilidade	Medida de arrependimento
Operário01	0.08	0.07
Operário02	0.41	0.3
Operário03	0.7	0.3
Operário04	0.52	0.27
Operário05	0.54	0.25

Finalmente, é calculado o resultado que irá permitir realizar a classificação, através da fórmula 7. Na fórmula, S representa a medida de utilidade, R representa a medida de arrependimento, max o valor máximo de uma coluna, min o valor mínimo de uma coluna e V permite ajustar a prioridade das medidas. É assumido que as medidas são de igual importância embutindo V

com o valor de 0,5. A utilização da fórmula culmina nas tabelas 26 e 27 (Opricovic and Tzeng, 2007).

$$f(x) = \left(\frac{S_i - S_{min}}{S_{max} - S_{min}}\right)V + \left(\frac{R_i - R_{min}}{R_{max} - R_{min}}\right)(1 - V) \quad (7)$$

Tabela 26: Resultado VIKOR das tarefas

Tarefa	VIKOR	Classificação
tarefa01	0.98	1º
tarefa07	0.97	2º
tarefa05	0.9	3º
tarefa03	0.83	4º
tarefa06	0.69	5º
tarefa04	0.38	6º
tarefa02	0.26	7º
tarefa08	0.21	8º
tarefa09	0.15	9º
tarefa10	0	10º

Tabela 27: Resultado VIKOR dos operários

Operário	VIKOR	Classificação
Operário03	1	1º
Operário04	0.78	2º
Operário02	0.76	3º
Operário05	0.76	4º
Operário01	0	5º

5.7.3 Terceira parte da solução do Algoritmo

Na parte final é necessário realizar a atribuição das tarefas aos operários de forma equilibrada. Compreende-se equilibrado como não dar só as tarefas mais relevantes ao melhor operário. Um algoritmo que tem esta característica é o round-robin, especialmente caso os elementos venham pré-ordenados. Na segunda parte da solução os operários e as tarefas foram ordenados de acordo com a relevância, sob diferentes algoritmos. Para facilitar a avaliação dos resultados o round-robin foi aplicado aos emparelhamentos dos mesmos algoritmos para as tarefas e para os operários. Por outras palavras, uma das aplicações realizadas foi a combinação MAUT tarefas e MAUT operários. O round-robin foi aplicado no mesmo estilo aos restantes algoritmos da segunda parte da solução resultando nas diferentes atribuições representadas pelas tabelas 28, 29, 30 e 31.

Tabela 28: Atribuição MAUT

Tarefa	Operário
tarefa01	Operário03
tarefa07	Operário05
tarefa05	Operário04
tarefa03	Operário02
tarefa06	Operário01
tarefa04	Operário03
tarefa02	Operário05
tarefa08	Operário04
tarefa09	Operário02
tarefa10	Operário01

Tabela 29: Atribuição PROMETHEE II

Tarefa	Operário
tarefa01	Operário03
tarefa07	Operário05
tarefa05	Operário04
tarefa03	Operário02
tarefa06	Operário01
tarefa04	Operário03
tarefa02	Operário05
tarefa08	Operário04
tarefa09	Operário02
tarefa10	Operário01

Tabela 30: Atribuição TOPSIS

Tarefa	Operário
tarefa01	Operário05
tarefa03	Operário04
tarefa07	Operário03
tarefa05	Operário02
tarefa06	Operário01
tarefa04	Operário05
tarefa02	Operário04
tarefa08	Operário03
tarefa09	Operário02
tarefa10	Operário01

Tabela 31: Atribuição VIKOR

Tarefa	Operário
tarefa01	Operário03
tarefa07	Operário04
tarefa05	Operário02
tarefa03	Operário05
tarefa06	Operário01
tarefa04	Operário03
tarefa02	Operário04
tarefa08	Operário02
tarefa09	Operário05
tarefa10	Operário01

5.8 Testes

Este projeto atinge uma percentagem quase completa de testes unitários. Para a primeira parte da solução, os testes unitários são relativamente diretos. No entanto, para a segunda parte eles tornam-se complexos devido à natureza dos algoritmos. Tendo isto em conta, de início foi utilizado o Microsoft Office Excel para recriar tabelas existentes e comparar os resultados obtidos com os resultados apresentados. Depois estas tabelas foram traduzidas para testes unitários para garantir a consistência e o correto funcionamento dos algoritmos utilizados na segunda parte. Para a última parte foram utilizados dados gerados e através da folha de cálculo anterior, foi criado um resultado esperado para as atribuições. Este resultado foi depois utilizado nos testes unitários.

5.9 Avaliação da solução

Tal como anteriormente referido, não existe uma solução única para este problema. Para combater isto, foi realizado um questionário para encontrar a solução com maior preferência. Mas a informação necessária para analisar e poder realizar comparações entre as diferentes atribuições é complexa. Considerando isto, foi utilizada uma técnica de comparação por pares onde duas soluções eram apresentadas e depois era pedido a preferência entre elas. Esta técnica era depois repetida com diferentes combinações de atribuições até permitir indicar a ordem de preferência entre elas. Este questionário obteve 17 respostas.

5.9.1 Resultados do questionário

As comparações realizadas são apresentadas na Tabela 32, onde cada linha é uma comparação. As colunas Algoritmo1 e Algoritmo2 indicam qual a atribuição utilizada baseada nos algoritmos presentes na segunda parte da solução do problema anteriormente referido. E as colunas Preferência1 e Preferência2 indicam a preferência das pessoas numa comparação e a coluna Preferência1 indica a percentagem da preferência do algoritmo presente na coluna Algoritmo1 e a coluna Preferência2 indica a percentagem da preferência do algoritmo presente na coluna Algoritmo2.

Tabela 32: Resultados do questionário

Algoritmo1	Preferência1	Algoritmo2	Preferência2
MAUT	58.8%	PROMETHEE II	41.2%
MAUT	64.7%	TOPSIS	35.3%
MAUT	35.3%	VIKOR	64.7%
PROMETHEE II	35.3%	TOPSIS	64.7%
PROMETHEE II	41.2%	VIKOR	58.8%
TOPSIS	17.6%	VIKOR	82.4%

5.9.2 Análise dos resultados obtidos

Nos resultados dos algoritmos podemos observar que o MAUT e o PROMETHEE II obtêm o mesmo resultado. As tarefas são todas ordenadas da mesma forma com a exceção do TOPSIS que só altera a posição das tarefas tarefa03, tarefa05 e tarefa07. Os operários são ordenados sempre de maneira diferente com a exceção do MAUT e o PROMETHEE II e o operário01 é consistentemente o operário com menos relevância.

Nos resultados dos questionários, obtemos as conclusões que VIKOR é mais preferível que MAUT, PROMETHEE II e TOPSIS, que MAUT é mais preferível que PROMETHEE II e TOPSIS e que TOPSIS é mais preferível que PROMETHEE II. A relação de preferência do TOPSIS com PROMETHEE II revela ser especialmente problemática, dado que PROMETHEE II e MAUT são iguais e que MAUT é mais preferível do que TOPSIS. Isto mostra que existem dados incongruentes nos resultados das respostas obtidas.

5.10 Conclusão

Neste capítulo, foi indicado a forma de como os casos uso foram implementados e a aplicação das práticas de SOLID e GRASP. Também é explicado como a solução foi atingida. Existem quatro versões da solução e elas foram divididas em três partes. A primeira obtém toda a informação necessária. A segunda aplica um algoritmo decisão multi-critério (MAUT, PROMETHEE II, TOPSIS ou VIKOR), para ordenar a informação obtida. E a última parte utiliza o algoritmo round-robin para realizar a atribuição de tarefas aos operários. Foi também realizado um questionário a perguntar a ordem de preferência das soluções referidas. O questionário permitiu indicar o VIKOR como o mais preferível, no entanto foram encontradas algumas inconsistências nos resultados dos questionários.

6 Conclusão

Com o crescimento das redes de informação, revela-se que o tratamento da informação de forma manual torna-se incrementalmente inviável, e que um dos possíveis caminhos é recorrer a automação para realizar os respetivos processos de preparação de dados.

Este projeto permite evidenciar múltiplas formas de como conjuntos de dados complexos podem ser tratados para atingir um determinado objetivo. O projeto determina diferentes formas para atribuir tarefas a operários, no entanto, pode também ser utilizado com outras finalidades como, por exemplo, indicar o carro que se adequa melhor a uma pessoa perante o estilo de condução dela.

6.1 Objetivos Concretizados e Resultados obtidos

Dada a condição inicial podemos indicar que foram alcançadas múltiplas soluções para o problema original e que através do desenho e implementação os requisitos funcionais e não funcionais são cumpridos.

Os resultados dos algoritmos permitem evidenciar que a existência de mais critérios por cada tarefa ou operário pode provocar uma maior discrepância na ordem em que eles são classificados, o que irá afetar os resultados da atribuição. Tendo isto em conta, quantos mais critérios forem utilizados para realizar uma atribuição, mais importante é a implementação de múltiplos algoritmos de decisão multi-critério e a realização de comparação dos resultados obtidos por eles.

Os resultados dos questionários indicam uma preferência pelo algoritmo VIKOR, mas dada a quantidade de respostas obtidas e a incongruência dos resultados dos algoritmos restantes não se pode aconselhar nenhum deles baseados nos questionários. Com base no referido anteriormente, e tendo em consideração a complexidade e o processo de implementação dos algoritmos, aconselha-se a iniciar uma implementação pelo MAUT devido a este ser o mais simples, seguido pelo VIKOR por este partilhar uma parte da implementação com o MAUT, depois o TOPSIS por ser menos complexo que o PROMETHEE II, acabando no PROMETHEE II.

6.2 Limitações e trabalho futuro

Neste trabalho podemos observar que a complexidade da informação necessária de analisar para indicar que uma atribuição é melhor do que outra e que a inexistência de uma solução única são fatores limitantes. Isto implica depender da subjetividade humana para realizar a comparação dos resultados obtidos, o que pode levar a incongruências de resultados.

Neste projeto foram estudadas quatro soluções para atingir o resultado. Embora o resultado tenha sido alcançado, existem diferentes algoritmos de decisão multi-critério, diferentes combinações de algoritmos de decisão multi-critério e a aplicação de inteligência artificial por estudar, mostrando que este tema pode ser mais aprofundado.

6.3 Apreciação final

Este projeto foi realizado com o intuito de tentar aproximar a uma possível solução num problema não determinístico dentro da área da automação, e possivelmente enriquecer produtos existentes com novas funcionalidades. Embora não se tenha conseguido indicar qual o algoritmo que é mais preferível, foi possível traçar um caminho para facilitar a exploração deles e sugerir uma rota de implementação de acordo com o processo e a complexidade de implementação.

Referências

- (Baker and Kenett, 1999) Baker, E. and Kenett, R. (1999). *Software Process Quality Management and Control*. Crc Press.
- (Brans and Mareschal, 2020) Brans, J.-P. and Mareschal, B. (2020). Promethee Methods. *International Series in Operations Research & Management Science*, pp.163–186. doi:10.1007/0-387-23081-5_5.
- (Fowler, 2015) Fowler, M. (2015). *Patterns of enterprise application architecture*. Boston, Mass. ; Munich: Addison-Wesley.
- (GeeksforGeeks, 2020) GeeksforGeeks (2020). *Automation and its Types*. Available at: <https://www.geeksforgeeks.org/automation-and-its-types/> [Accessed 14 feb. 2022].
- (IBM, 2021) *What is automation?* www.ibm.com. Available at: <https://www.ibm.com/topics/automation> [Accessed 15 feb. 2022]
- (IFTTT, n.d.) *IFTTT (n.d.) IFTTT*. ifttt.com. Available at: <https://ifttt.com> [Accessed 23 feb. 2022]
- (Industrialautomation360.com, n.d.) Industrialautomation360.com (n.d.) Available at: <https://industrialautomation360.com/types/> [Accessed 22 feb. 2022]
- (Jacek Blazewicz and Al, 2001) Jacek Blazewicz and Al, E. (2001). *Scheduling computer and manufacturing processes*. Berlin ; New York: Springer.
- (mijacobs, n.d.) mijacobs (n.d.). *What is Scrum? - Azure DevOps*. docs.microsoft.com. Available at: <https://docs.microsoft.com/en-us/devops/plan/what-is-scrum> [Accessed 21 feb. 2022]
- (Miro, n.d.) Miro (n.d.) *The Online Whiteboard for Seamless Task Management | Miro*. [online] Available at: https://miro.com/aq/capterra/task-management/?utm_source=capterra&utm_medium=cpc&utm_campaign=task_management&utm_content=visite_site&utm_source=GetApp [Accessed 25 feb. 2022].
- (monday, n.d.) monday (n.d.) *Task Management Platform*. Available at: https://monday.com/lp/mb/task/comparison?marketing_source=adwordssearch&marketing_campaign=row-s-taskgen-b-desk-monday&aw_keyword=task&aw_match_type=p&cluster=generic&subcluster=&clid=Cj0KCQiApL2QBhC8ARIsAGMm-KFo7PEZbm8-JRNmTJPCacWg-EKht2CPae-qwxQ9DELdZ1yLEr2RYO3laAirKEALw_wcB [Accessed 25 feb. 2022].
- (Opricovic and Tzeng, 2007) Opricovic, S. and Tzeng, G.-H. (2007). Extended VIKOR method in comparison with outranking methods. *European Journal of Operational Research*, 178(2), pp.514–529. doi:10.1016/j.ejor.2006.01.020.

- (PollUnit. n.d.) PollUnit. (n.d.) *Distribute tasks with PollUnit*. Available at: https://pollunit.com/en/tutorials/distribute_tasks [Accessed 26 feb. 2022].
- (Profit.co., n.d.) Profit.co. (n.d.). *OKRs | Tasks | Performance Management*. Available at: <https://www.profit.co/product/> [Accessed 26 feb. 2022].
- (Refactoring.guru, 2014) Refactoring.guru. (2014). *Strategy*. Available at: <https://refactoring.guru/design-patterns/strategy> [Accessed 19 mar. 2022].
- (Shanmuganathan et al., 2018) Shanmuganathan, M., Kajendran, K., Sasikumar, A., Mahendran, M. and Faculty (2018). Multi Attribute Utility Theory -An Over View. *International Journal of Scientific & Engineering Research*, [online] 9(3). Available at: <https://www.ijser.org/researchpaper/Multi-Attribute-Utility-Theory-An-Over-View.pdf>.
- (Wątróbski et al., 2019) Wątróbski, J., Jankowski, J., Ziemia, P., Karczmarczyk, A. and Ziolo, M. (2019). Generalised framework for multi-criteria method selection. *Omega*, [online] 86, pp.107–124. doi:10.1016/j.omega.2018.07.004.
- (Yadav et al., 2019) Yadav, V., Karmakar, S., Kalbar, P.P. and Dikshit, A.K. (2019). PyTOPS: A Python based tool for TOPSIS. *SoftwareX*, 9, pp.217–222. doi:10.1016/j.softx.2019.02.004.

Anexos

Tabela 33: Diferença avaliativa das Tarefas

Relações das Tarefas	Prioridade	Complexidade
tarefa01-tarefa02	0.7	0.25
tarefa01-tarefa03	0.16	0
tarefa01-tarefa04	0.48	0.75
tarefa01-tarefa05	0.04	0.25
tarefa01-tarefa06	0.35	-0.25
tarefa01-tarefa07	-0.03	0.25
tarefa01-tarefa08	0.74	0.25
tarefa01-tarefa09	0.87	0.25
tarefa01-tarefa10	0.97	0.5
tarefa02-tarefa01	-0.7	-0.25
tarefa02-tarefa03	-0.54	-0.25
tarefa02-tarefa04	-0.22	0.5
tarefa02-tarefa05	-0.65	0

Relações das Tarefas	Prioridade	Complexidade
tarefa02-tarefa06	-0.35	-0.5
tarefa02-tarefa07	-0.72	0
tarefa02-tarefa08	0.04	0
tarefa02-tarefa09	0.17	0
tarefa02-tarefa10	0.28	0.25
tarefa03-tarefa01	-0.16	0
tarefa03-tarefa02	0.54	0.25
tarefa03-tarefa04	0.32	0.75
tarefa03-tarefa05	-0.12	0.25
tarefa03-tarefa06	0.19	-0.25
tarefa03-tarefa07	-0.19	0.25
tarefa03-tarefa08	0.58	0.25
tarefa03-tarefa09	0.71	0.25
tarefa03-tarefa10	0.81	0.5
tarefa04-tarefa01	-0.48	-0.75

Relações das Tarefas	Prioridade	Complexidade
tarefa04-tarefa02	0.22	-0.5
tarefa04-tarefa03	-0.32	-0.75
tarefa04-tarefa05	-0.43	-0.5
tarefa04-tarefa06	-0.13	-1
tarefa04-tarefa07	-0.51	-0.5
tarefa04-tarefa08	0.26	-0.5
tarefa04-tarefa09	0.39	-0.5
tarefa04-tarefa10	0.49	-0.25
tarefa05-tarefa01	-0.04	-0.25
tarefa05-tarefa02	0.65	0
tarefa05-tarefa03	0.12	-0.25
tarefa05-tarefa04	0.43	0.5
tarefa05-tarefa06	0.3	-0.5
tarefa05-tarefa07	-0.07	0
tarefa05-tarefa08	0.7	0

Relações das Tarefas	Prioridade	Complexidade
tarefa05-tarefa09	0.83	0
tarefa05-tarefa10	0.93	0.25
tarefa06-tarefa01	-0.35	0.25
tarefa06-tarefa02	0.35	0.5
tarefa06-tarefa03	-0.19	0.25
tarefa06-tarefa04	0.13	1
tarefa06-tarefa05	-0.3	0.5
tarefa06-tarefa07	-0.38	0.5
tarefa06-tarefa08	0.39	0.5
tarefa06-tarefa09	0.52	0.5
tarefa06-tarefa10	0.62	0.75
tarefa07-tarefa01	0.03	-0.25
tarefa07-tarefa02	0.72	0
tarefa07-tarefa03	0.19	-0.25
tarefa07-tarefa04	0.51	0.5
tarefa07-tarefa05	0.07	0

Relações das Tarefas	Prioridade	Complexidade
tarefa07-tarefa06	0.38	-0.5
tarefa07-tarefa08	0.77	0
tarefa07-tarefa09	0.9	0
tarefa07-tarefa10	1	0.25
tarefa08-tarefa01	-0.74	-0.25
tarefa08-tarefa02	-0.04	0
tarefa08-tarefa03	-0.58	-0.25
tarefa08-tarefa04	-0.26	0.5
tarefa08-tarefa05	-0.7	0
tarefa08-tarefa06	-0.39	-0.5
tarefa08-tarefa07	-0.77	0
tarefa08-tarefa09	0.13	0
tarefa08-tarefa10	0.23	0.25
tarefa09-tarefa01	-0.87	-0.25
tarefa09-tarefa02	-0.17	0

Relações das Tarefas	Prioridade	Complexidade
tarefa09-tarefa03	-0.71	-0.25
tarefa09-tarefa04	-0.39	0.5
tarefa09-tarefa05	-0.83	0
tarefa09-tarefa06	-0.52	-0.5
tarefa09-tarefa07	-0.9	0
tarefa09-tarefa08	-0.13	0
tarefa09-tarefa10	0.1	0.25
tarefa10-tarefa01	-0.97	-0.5
tarefa10-tarefa02	-0.28	-0.25
tarefa10-tarefa03	-0.81	-0.5
tarefa10-tarefa04	-0.49	0.25
tarefa10-tarefa05	-0.93	-0.25
tarefa10-tarefa06	-0.62	-0.75
tarefa10-tarefa07	-1	-0.25
tarefa10-tarefa08	-0.23	-0.25
tarefa10-tarefa09	-0.1	-0.25

Tabela 34: Preferências das Tarefas

Relações das Tarefas	Prioridade	Complexidade
tarefa01-tarefa02	0.7	0.25
tarefa01-tarefa03	0.16	0
tarefa01-tarefa04	0.48	0.75
tarefa01-tarefa05	0.04	0.25
tarefa01-tarefa06	0.35	-0.25
tarefa01-tarefa07	-0.03	0.25
tarefa01-tarefa08	0.74	0.25
tarefa01-tarefa09	0.87	0.25
tarefa01-tarefa10	0.97	0.5
tarefa02-tarefa01	-0.7	-0.25
tarefa02-tarefa03	-0.54	-0.25
tarefa02-tarefa04	-0.22	0.5
tarefa02-tarefa05	-0.65	0
tarefa02-tarefa06	-0.35	-0.5
tarefa02-tarefa07	-0.72	0
tarefa02-tarefa08	0.04	0
tarefa02-tarefa09	0.17	0
tarefa02-tarefa10	0.28	0.25

Relações das Tarefas	Prioridade	Complexidade
tarefa03-tarefa01	-0.16	0
tarefa03-tarefa02	0.54	0.25
tarefa03-tarefa04	0.32	0.75
tarefa03-tarefa05	-0.12	0.25
tarefa03-tarefa06	0.19	-0.25
tarefa03-tarefa07	-0.19	0.25
tarefa03-tarefa08	0.58	0.25
tarefa03-tarefa09	0.71	0.25
tarefa03-tarefa10	0.81	0.5
tarefa04-tarefa01	-0.48	-0.75
tarefa04-tarefa02	0.22	-0.5
tarefa04-tarefa03	-0.32	-0.75
tarefa04-tarefa05	-0.43	-0.5
tarefa04-tarefa06	-0.13	-1
tarefa04-tarefa07	-0.51	-0.5
tarefa04-tarefa08	0.26	-0.5
tarefa04-tarefa09	0.39	-0.5
tarefa04-tarefa10	0.49	-0.25
tarefa05-tarefa01	-0.04	-0.25

Relações das Tarefas	Prioridade	Complexidade
tarefa05-tarefa02	0.65	0
tarefa05-tarefa03	0.12	-0.25
tarefa05-tarefa04	0.43	0.5
tarefa05-tarefa06	0.3	-0.5
tarefa05-tarefa07	-0.07	0
tarefa05-tarefa08	0.7	0
tarefa05-tarefa09	0.83	0
tarefa05-tarefa10	0.93	0.25
tarefa06-tarefa01	-0.35	0.25
tarefa06-tarefa02	0.35	0.5
tarefa06-tarefa03	-0.19	0.25
tarefa06-tarefa04	0.13	1
tarefa06-tarefa05	-0.3	0.5
tarefa06-tarefa07	-0.38	0.5
tarefa06-tarefa08	0.39	0.5
tarefa06-tarefa09	0.52	0.5
tarefa06-tarefa10	0.62	0.75
tarefa07-tarefa01	0.03	-0.25
tarefa07-tarefa02	0.72	0

Relações das Tarefas	Prioridade	Complexidade
tarefa07-tarefa03	0.19	-0.25
tarefa07-tarefa04	0.51	0.5
tarefa07-tarefa05	0.07	0
tarefa07-tarefa06	0.38	-0.5
tarefa07-tarefa08	0.77	0
tarefa07-tarefa09	0.9	0
tarefa07-tarefa10	1	0.25
tarefa08-tarefa01	-0.74	-0.25
tarefa08-tarefa02	-0.04	0
tarefa08-tarefa03	-0.58	-0.25
tarefa08-tarefa04	-0.26	0.5
tarefa08-tarefa05	-0.7	0
tarefa08-tarefa06	-0.39	-0.5
tarefa08-tarefa07	-0.77	0
tarefa08-tarefa09	0.13	0
tarefa08-tarefa10	0.23	0.25
tarefa09-tarefa01	-0.87	-0.25
tarefa09-tarefa02	-0.17	0
tarefa09-tarefa03	-0.71	-0.25
tarefa09-tarefa04	-0.39	0.5

Relações das Tarefas	Prioridade	Complexidade
tarefa09-tarefa05	-0.83	0
tarefa09-tarefa06	-0.52	-0.5
tarefa09-tarefa07	-0.9	0
tarefa09-tarefa08	-0.13	0
tarefa09-tarefa10	0.1	0.25
tarefa10-tarefa01	-0.97	-0.5
tarefa10-tarefa02	-0.28	-0.25
tarefa10-tarefa03	-0.81	-0.5
tarefa10-tarefa04	-0.49	0.25
tarefa10-tarefa05	-0.93	-0.25
tarefa10-tarefa06	-0.62	-0.75
tarefa10-tarefa07	-1	-0.25
tarefa10-tarefa08	-0.23	-0.25
tarefa10-tarefa09	-0.1	-0.25