



Estratégias de Escalonamento de Produção Inteligente na Indústria 4.0

FÁBIO MIGUEL MARTINS COELHO

Outubro de 2020

Estratégias de Escalonamento de Produção Inteligente na Indústria 4.0

Fábio Miguel Martins Coelho

**Dissertação para obtenção do Grau de Mestre em
Engenharia Informática, Área de Especialização em
Engenharia de Software**

Orientador: Goretí Marreiros

Supervisor: André Oliveira

Júri:

Presidente:

[Nome do Presidente, Categoria, Escola]

Vogais:

[Nome do Vogal1, Categoria, Escola]

[Nome do Vogal2, Categoria, Escola] (até 4 vogais)

Porto, outubro 2020

*Dedico aos meus pais,
à minha família
e aos meus amigos.*

Resumo

O processo de escalonamento da produção é complexo, sendo o eficaz escalonamento e planeamento dos recursos um problema transversal à indústria, que pode afetar diretamente a produtividade e a competitividade.

No paradigma da indústria 4.0, esta temática torna-se ainda mais relevante. Colaboradores, máquinas e *software* estão interligados, originando a necessidade da introdução de mudanças no processo de escalonamento, com o objetivo de o tornar mais eficaz. Posto isto, o escalonador da Sistrade, com as suas limitações, não é capaz de dar uma boa resposta a esta problemática. Assim, esta dissertação tem como objetivo dar resposta a esta necessidade de mudança/ inovação no escalonamento da produção.

Com o objetivo de melhorar a capacidade de resposta face aos novos paradigmas da indústria 4.0, foi desenvolvido um algoritmo totalmente novo, capaz de efetuar otimizações segundo critérios, nomeadamente, diminuição da percentagem de ordens de fabrico atrasadas, aumento da taxa de ocupação dos recursos, diminuição do tempo total de preparação e diminuição da duração total da produção (*makespan*). Para além disto, foi desenvolvida uma solução que incorpora este algoritmo. Esta, possui novos aspetos funcionais e de integração face à solução da Sistrade. Nomeadamente, é independente da plataforma (podendo ser facilmente integrada com outros sistemas), permite a utilização de recursos alternativos, de matrizes de tempos de preparação e balanceamento de carga.

Os resultados obtidos nas experiências realizadas permitiram concluir que a solução desenvolvida obtém melhores resultados do que a solução da Sistrade, traduzindo-se em ganhos significativos.

Palavras-chave: algoritmos de escalonamento, escalonamento da produção, critérios de escalonamento, balanceamento de carga, matriz de tempos de preparação.

Abstract

The production scheduling process is complex, and the efficient resource scheduling and planning is a problem that cuts across industry, which can directly affect productivity and competitiveness.

In the industry 4.0 paradigm, this theme becomes even more relevant. Employees, machines and software are interconnected, resulting in the need to introduce changes in the scheduling process, in order to make it more effective. That said, the Sistrade scheduler, with its limitations, is not able to give a good answer to this problem. Thus, this dissertation aims to respond to this need for change/innovation in the production schedule.

In order to improve the responsiveness to the new paradigms of industry 4.0, a new algorithm was developed, capable of making optimizations according to criteria, namely, decrease the percentage of delayed manufacturing orders, increase the resource occupation rate, decrease the total preparation time and decrease the makespan. In addition, a solution was developed that incorporates that algorithm. This has new functional and integration aspects compared to the Sistrade solution. Namely, it is platform independent (can be easily integrated with other systems), allows the use of alternative resources, preparation time matrices and load balancing.

The results obtained in the experiments allowed to conclude that the developed solution obtains better results than the Sistrade solution, originating significant gains.

Keywords: scheduling algorithms, production scheduling, scheduling criteria, load balancing, setup time matrix.

Agradecimentos

Em primeiro lugar, quero agradecer a quem faz parte da minha vida pessoal pelo apoio e incentivo constante. Agradeço especialmente aos meus pais, por tornarem possível o meu percurso e sempre me apoiarem em todas as decisões que tomo. Obrigado a todos os familiares e amigos que apoiam e comemoram comigo todos os momentos de vitória e superação pessoal e profissional. Àqueles amigos que são família e com os quais partilho os melhores e os piores momentos, o meu eterno agradecimento.

Agradeço ao Instituto Superior de Engenharia do Porto (ISEP) e a todas as pessoas que constituem esta organização, em especial ao Departamento de Engenharia Informática (DEI) pela excelência no conhecimento transmitido.

À professora Goreti Marreiros, um muito obrigado pelo apoio prestado e por todas as críticas construtivas fornecidas. Obrigado por ter sido a minha orientadora ao longo deste processo e por me ter ajudado a evoluir.

À Sistrade por ter disponibilizado a oportunidade de realizar o presente projeto e fornecido sempre o devido apoio e incentivo. Agradeço a todas as pessoas desta organização por todo o conhecimento transmitido. Em especial, ao meu supervisor André Oliveira por me ter transmitido inúmeros conceitos de planeamento da produção e pela ajuda ao longo da realização desta dissertação.

A todos os que referi e a quem possa, porventura, ter esquecido de referir, o meu mais sincero obrigado!

Índice

1	Introdução	1
1.1	Contexto	1
1.2	Problema.....	3
1.3	Objetivos.....	4
1.4	Resultados Esperados	4
1.5	Abordagem Preconizada.....	5
2	Enquadramento Teórico	7
2.1	Enquadramento Teórico.....	7
2.1.1	Ordem de fabrico	7
2.1.2	Operação.....	7
2.1.3	Recurso.....	9
2.1.4	Escalonamento.....	9
2.1.5	Balanceamento	10
2.1.6	Matriz de Tempos de Preparação.....	10
2.1.7	MRP	11
2.2	Escalonamento do Sistrade ERP.....	11
2.2.1	Escalonador	12
2.2.2	Gantt	13
2.2.3	Avaliação	13
3	Análise de Valor	15
3.1	Modelo NCD.....	15
3.1.1	Identificação e Análise da Oportunidade	16
3.1.2	Geração e Enriquecimento de Ideias	17
3.1.3	Seleção de Ideias.....	17

3.1.4	Definição do Conceito	18
3.2	Valor para o Cliente.....	18
3.3	Proposta de Valor	19
3.4	Modelo de Negócio	19
4	Estado da Arte	23
4.1	Soluções/Abordagens Existentes.....	23
4.2	Algoritmos de Otimização.....	26
4.2.1	Pesquisa local - <i>Tabu Search</i>	26
4.2.2	Abordagem Evolutiva - Algoritmos Genéticos	29
4.3	Avaliação de Soluções/Abordagens Existentes	30
4.3.1	Algoritmo 5	30
4.3.2	Algoritmo 6	31
4.3.3	Algoritmo 8	31
4.3.4	Algoritmo 9	32
4.3.5	Conclusão	32
4.4	Soluções/Abordagens a Adotar.....	33
4.4.1	Abordagem Principal.....	33
4.4.2	Abordagem Alternativa.....	34
4.5	Tecnologia Relevante	35
4.5.1	API	36
4.5.2	Tratamento de Dados de <i>Input</i> e <i>Output</i>	36
5	Análise e Design	39
5.1	Análise de Requisitos	39
5.1.1	Atores do Sistema	39
5.1.2	Requisitos Funcionais	39
5.1.3	Requisitos Não Funcionais	40
5.2	Modelo de Domínio	41
5.3	Design Arquitetural	43

5.3.1	Vista Lógica	43
5.3.2	Vista de Processo.....	44
6	Desenvolvimento	47
6.1	Tecnologias.....	47
6.1.1	Ambiente de Desenvolvimento.....	47
6.1.2	Framework .NET Core 2.1	48
6.1.3	Base de Dados.....	49
6.2	Algoritmo de Escalonamento	49
6.3	Dados de Input	54
6.3.1	Ordem de Fabrico	55
6.3.2	Recurso.....	57
6.3.3	Matriz de Tempos de Preparação.....	57
6.3.4	Configurações	59
6.4	Dados de Output.....	60
7	Avaliação.....	61
7.1	Indicadores e Fontes de Informação	61
7.1.1	Indicadores	61
7.1.2	Fontes de Informação	62
7.2	Hipóteses a Avaliar.....	62
7.2.1	Eficiência	63
7.2.2	Eficácia.....	63
7.3	Metodologia de Avaliação e Resultados	64
7.3.1	Dados de Input.....	64
7.3.2	Eficiência	66
7.3.3	Eficácia.....	68
7.4	Conclusões.....	70
8	Conclusões e Trabalho Futuro	71

8.1	Principais Conclusões	71
8.2	Objetivos Alcançados	72
8.3	Limitações e Trabalho Futuro	72

Lista de Figuras

Figura 1 – Escalonador Sistrade	3
Figura 2 – Abordagem preconizada	5
Figura 3 – Exemplo de Gama Operatória	8
Figura 4 – Escalonamento	9
Figura 5 – <i>Gantt</i> do planeamento do ERP Sistrade	12
Figura 6 – Modelo NCD	16
Figura 7 – Algoritmo <i>tabu search</i>	28
Figura 8 – Algoritmo genético (José Lindenberg, 2016).....	30
Figura 9 – Abordagem principal	34
Figura 10 – Abordagem alternativa	35
Figura 11 – <i>Workflow</i> de pedido a uma API.....	36
Figura 12 – Diagrama de casos de uso	40
Figura 13 – Modelo de domínio	42
Figura 14 – Vista lógica.....	43
Figura 15 – Diagrama de sequência do processo de escalonar	44
Figura 16 – Estrutura do Projeto	48
Figura 17 – Algoritmo desenvolvido	50
Figura 18 – Dados de input – Ordem de Fabrico	55
Figura 19 – Dados de input – Recurso.....	57
Figura 20 – Dados de input – Matriz de tempos de preparação	58
Figura 21 – Dados de input - Configurações	59
Figura 22 – Dados de output.....	60

Lista de Tabelas

Tabela 1 – Exemplo de matriz de tempos de preparação (minutos).....	10
Tabela 2 – modelo negócio (canvas).....	21
Tabela 3 – Comparação das abordagens existentes.....	26
Tabela 4 – Termos do algoritmo tabu search	28
Tabela 5 – Dados de input C1.....	65
Tabela 6 - Dados de input C2	65
Tabela 7 – Avaliação da eficiência.....	67
Tabela 8 – Avaliação da eficácia.....	68
Tabela 9 – Avaliação dos parâmetros da solução proposta.....	69

Acrónimos e Símbolos

Lista de Acrónimos

API	Interface de programação de aplicações (<i>Application programming interface</i>)
APS	Planeamento e Escalonamento Avançado (<i>Advanced Planning and scheduling</i>)
ERP	Planeamento de Recursos Empresariais (<i>Enterprise Resource Planning</i>)
FURPS	Funcionalidade (<i>Functionality</i>), Usabilidade (<i>Usability</i>), Confiabilidade (<i>Reliability</i>), Desempenho (<i>Performance</i>), Suportabilidade (<i>Supportability</i>)
JSON	Notação de Objetos JavaScript (<i>JavaScript Object Notation</i>)
MRP	Planeamento de Necessidades de Materiais (<i>Manufacturing Resource Planning</i>)
MTO	Produção sob encomenda (<i>Make-To-Order</i>)
MTS	Produção para stock (<i>Make-To-Stock</i>)
NCD	Novo conceito de desenvolvimento (<i>New Concept Development</i>)
OF	Ordem de Fabrico
XML	Linguagem Padronizada de Marcação Genérica (<i>Standard Generalized Markup Language</i>)

1 Introdução

Um dos objetivos deste capítulo é dar a perceber o contexto no qual se insere o problema central deste documento. Para além disso, será exposto problema, os objetivos, os resultados esperados e, por fim, a abordagem preconizada.

1.1 Contexto

O eficaz escalonamento e planeamento dos recursos em ambiente produtivo é um problema transversal à indústria, diretamente responsável pela sua produtividade e competitividade. Este problema torna-se mais relevante dentro do paradigma da indústria 4.0 (Rossit, Tohmé and Frutos, 2019), onde colaboradores, máquinas e *software* constituem Sistemas de Produção Ciber-Físicos (SPCF) (Silva *et al.*, 2017), nos quais repartem responsabilidades de decisão, colaboração e autonomia, interconectando-se entre si e introduzindo mudanças no processo de escalonamento. Neste contexto, as cadeias de fornecimento, produção, expedição e consumo encontram-se estreitamente interligadas, o que aumenta os desafios em termos de tempo de resposta e competitividade organizacional, com concreta evidência na produção.

De forma a se entender a problemática do tema em questão, é necessário compreender que, um mau planeamento da produção ou a inexistência do mesmo, pode resultar em elevados prejuízos para uma organização. Isto acontece devido a vários fatores, mas alguns dos principais são:

- Ordens de fabrico (entenda-se por Ordem de fabrico, “a instrução de trabalho a enviar para o chão de fábrica” [Sistrade Software Consulting, 2020]) em atraso: o aumento das ordens de fabrico em atraso e o conseqüente incumprimento dos prazos de entrega das mesmas, originando possíveis prejuízos e insatisfação por parte do cliente (Study, Mohamed and Coutry, 2015);

- Taxas de ocupação dos recursos reduzida: os recursos responsáveis pela produção podem estar associados a investimentos financeiros muito elevados, o que significa que é impensável a existência de recursos parados devido a um mau planeamento da produção (Shaban and Al-zubi, 2014);
- Ocupação dos recursos não otimizada: os tempos e custos de preparação dos recursos para algumas ordens de fabrico é elevado, como tal, pode não fazer sentido fazer trabalhos semelhantes em períodos não sequenciais. Ou seja, poderá ser feita uma otimização tendo por base a redução dos custos e tempos de preparação (Allahverdi and Soroush, 2006). Trovinger e Bohn (Trovinger and Bohn, 2005), num estudo sobre a indústria da montagem de placas de circuitos impressos, concluíram que 50% da capacidade das máquinas pode ser perdida em tempos de preparação e que uma redução do tempo de preparação beneficia diretamente os custos, nomeadamente, uma redução do tempo de preparação em pelo menos 80%, traduziu-se em ganhos de 1,8 milhões de dólares por ano.

Numa outra perspetiva, numa organização com muitas dezenas de ordens de fabrico ou até mesmo com centenas, a realização de um planeamento da produção de forma manual, tendo em conta todos os fatores e otimizações, torna-se demasiado complexa e morosa, sendo muito difícil obter um bom resultado em tempo útil. Assim, torna-se necessário uma ferramenta de planeamento da produção capaz de otimizar segundo as diferentes metas da organização e perante os vários cenários possíveis.

Associado ao tema em questão, surge o conceito de sistema APS (*Advanced Planning and scheduling*). Um sistema APS pode ser definido como: “Técnicas que lidam com análise e planeamento de logística e fabricação durante curtos períodos de tempo, intermédios e de longo prazo. APS descreve qualquer programa de computador que utilize algoritmos ou lógica matemática avançados para executar otimizações ou simulações no escalonamento de capacidades finitas, na cadeia de fornecimento, no planeamento de capital, no planeamento de recursos, na gestão de necessidades, entre outros. Estas técnicas consideram simultaneamente uma série de restrições e regras de negócio para fornecer planeamento e escalonamento em tempo real, apoio a decisões, datas de compromisso e capacidades disponíveis dos recursos. O APS normalmente gera e avalia vários cenários.” (APICS, 2013)

A otimização dos recursos e dos SPCF de modo a ir ao encontro dos objetivos da gestão (pela maximização de lucro, otimização dos tempos de entrega, diminuição de ordens de fabrico em espera, otimização das taxas de ocupação de recursos, entre outros) necessita de novas abordagens ao planeamento e escalonamento de produção, traduzidos na premência de algoritmos eficazes, capazes fornecer vantagens competitivas. Torna-se fundamental a existência de um sistema APS realmente inovador e capaz de dar resposta aos pressupostos descritos.

1.2 Problema

Atualmente, a Sistrade possui no seu ERP um módulo de planeamento da produção que, por sua vez, possui um escalonador (Sistrade Software Consulting, 2018). Este módulo, pode ser considerado como sendo um sistema APS, ainda que com diversas deficiências no que diz respeito ao processo de escalonamento.

Um dos problemas é o facto do escalonador ser orientado apenas aos processos existentes no seu ERP e não permitir a integração com outros sistemas/*softwares*.

Por outro lado, na maioria das vezes, a solução obtida não é a mais otimizada para os critérios definidos. Ou seja, o seu modo de funcionamento é limitado, uma vez que apenas faz uma sequenciação dos trabalhos de acordo com os critérios definidos, não existindo uma procura por uma solução ótima ou quase ótima. Tomando como exemplo a data prevista de entrega de um trabalho, o escalonador aplica uma ordenação segundo essa data e depois atribui períodos de tempo para a realização de cada operação de cada trabalho, como se pode observar na Figura 1, nunca esquecendo todas as restrições e dependências. Não existe uma complexidade maior do que esta, ou seja, não é possível, por exemplo, otimizar a data prevista de entrega e a taxa de ocupação dos recursos em simultâneo. De forma simplista, a sua forma de otimização é uma mera ordenação segundo determinados critérios.

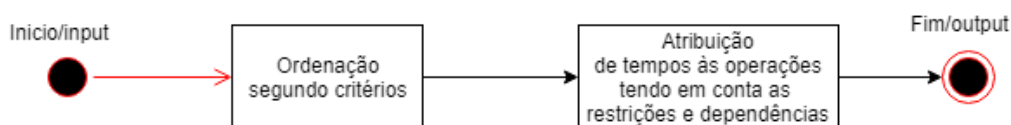


Figura 1 – Escalonador Sistrade

Para além do referido anteriormente, o escalonador em questão, também peca pela falta de algumas funcionalidades, como por exemplo, a funcionalidade de balanceamento de carga e de matrizes de tempos de preparação, tornando o resultado final aquém do esperado.

Outro dos principais problemas é a elevada necessidade de intervenção por parte de um planeador. Isto acontece porque o escalonador atual não consegue aplicar as otimizações necessárias de forma a que o resultado vá ao encontro do resultado esperado pelo planeador. Em algumas empresas existe um objetivo no sentido de tentar diminuir ao máximo a função do planeador, esperando que o sistema seja capaz de obter um resultado igual ou superior (mais otimizado) do que o obtido quando realizado por um ser humano.

Em suma, o escalonador Sistrade não é flexível e apresenta falhas no que diz respeito à otimização de soluções. Conclui-se que existe espaço para melhoria no sentido de o otimizar, inserindo novas funcionalidades e melhorando as já existentes.

1.3 Objetivos

O principal objetivo deste projeto é o desenvolvimento de uma solução de software capaz de efetuar o escalonamento da produção através de algoritmos inovadores que otimizam as diferentes metas das organizações, representado um salto qualitativo relativamente aos algoritmos atualmente aplicados pelo sistema da Sistrade.

Adicionalmente, a possibilidade de integração com outros sistemas também é fulcral.

Pretende-se ainda adicionar novas funcionalidades, nomeadamente a criação de uma matriz de determinação de tempos de preparação e o balanceamento de carga.

Para que estes objetivos sejam cumpridos, foram delineados os seguintes objetivos complementares:

- Identificação das abordagens/algoritmos mais relevantes e apropriadas, que permitam resolver os problemas apresentados pelo algoritmo da Sistrade;
- Conceção de novos algoritmos, aplicados ao escalonamento de produção no contexto da Indústria 4.0, que tenham em conta os seus desafios e contribuam para a excelência da produção e aumento da vantagem competitiva;
- Experimentação dos algoritmos desenvolvidos;
- Criação de uma solução que permita a integração com outros sistemas e que incorpore das funcionalidades de matriz de tempos de preparação e de balanceamento de carga;
- Integração da solução desenvolvida no protótipo;
- Avaliação da solução desenvolvida.

1.4 Resultados Esperados

O resultado esperado, de uma forma simplista, é o cumprimento de todos os objetivos referidos anteriormente.

Principalmente, pretende-se que o software produzido seja capaz de escalar, em tempo útil e segundo várias metas, um conjunto de ordens de fabrico. Assim, pretende-se que seja capaz de obter resultados positivos e melhorias significativas nos seguintes indicadores:

- Diminuição da percentagem de ordens de fabrico atrasadas;

- Aumento da taxa de ocupação dos recursos;
- Diminuição do tempo total de preparação;
- Diminuição da duração total da produção (*makespan*).

Em forma de complemento e para que se obtenha um escalonador mais apelativo e diversificado, é ainda esperado que a aplicação cumpra com outros requisitos, dos quais se destacam: ser genérica o suficiente para ser utilizada por outros softwares, permita o uso de uma matriz de determinação de tempos de preparação e, por fim, permita efetuar o balanceamento de carga.

1.5 Abordagem Preconizada

De forma a cumprir com os objetivos idealizados, será criada uma aplicação capaz de receber dados de *input* de trabalhos a escalonar, num formato que a torna genérica e implementável por outros sistemas, e que irá devolver um resultado (*output*) segundo as definições e critérios selecionados. Ou seja, de uma forma simplista:

1. A aplicação recebe todos os dados necessários, como por exemplo, ordens de fabrico, operações, recursos, critérios a otimizar, etc;
2. Efetua o escalonamento das ordens de fabrico, ocorrendo as diversas otimizações, segundo os algoritmos desenvolvidos, na tentativa de otimizar os critérios definidos;
3. Devolve o resultado final.

Na Figura 2, é apresentado o *workflow* da abordagem definida e descrita anteriormente, com um nível de detalhe mais elevado.

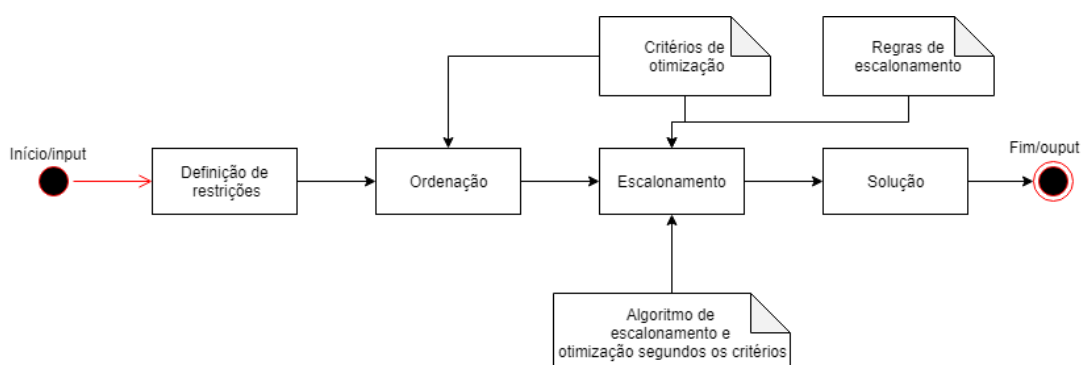


Figura 2 – Abordagem preconizada

Para o sucesso desta aplicação, é fundamental delinear muito bem uma abordagem sólida que defina quais os processos que devem ser realizados, nomeadamente:

- Identificação do problema: definição do problema que originou este projeto;
- Definição de objetivos: definição de objetivos e metas para o projeto;
- Realização do estado da arte: trabalho de investigação de algoritmos e soluções que possam ser utilizados para responder ao problema;
- Realização da análise e do design: realização da análise, e definição do design e da arquitetura da solução;
- Implementação da aplicação: desenvolvimento da solução seguindo o design definido;
- Demonstração e experimentação: realização de vários cenários com diferentes *inputs* para demonstrarem os diversos *outputs* otimizados. Experimentação através da integração com dados do Sistrade ERP;
- Avaliação da solução implementada: comparação dos resultados obtidos com esta solução face à solução existente na Sistrade, através da comparação de diversos indicadores.

2 Enquadramento Teórico

Neste capítulo, será, inicialmente, apresentado o enquadramento teórico e, posteriormente, o processo de escalonamento do Sistrade ERP.

2.1 Enquadramento Teórico

No presente subcapítulo, serão apresentados os conceitos teóricos mais relevantes relacionados com o tema do projeto e que são fundamentais para o seu desenvolvimento e entendimento.

2.1.1 Ordem de fabrico

Neste âmbito, tal como referido anteriormente, uma ordem de fabrico é considerada como uma ordem de trabalho que será realizada no chão de fábrica (Sistrade Software Consulting, 2020). É constituída por um conjunto de operações que serão realizadas para se obter o trabalho final.

Normalmente, quando a produção não é em série, está associada a um cliente específico, tendo resultado de uma encomenda por parte do mesmo. Assim, está também associada a uma data de entrega previamente definida a pedido do cliente.

2.1.2 Operação

Como referido, uma ordem de fabrico possui um conjunto de operações. Cada operação representa uma ação/etapa que tem de ser realizada para se conseguir obter o produto final.

As operações têm que ser realizadas de acordo com uma determinada ordem, ordem essa que é considerada como uma dependência para poder ser realizada. A este conjunto de sequências de operações e das suas dependências, pode-se dar o nome de gama operatória (Fonseca, 2013).

Na Figura 3, é apresentado um exemplo de um processo simples, com três operações. A primeira operação a ser realizada é a “Pré-impressão”, seguindo-se a “Impressão” e, por fim, a operação de “Acabamentos”.

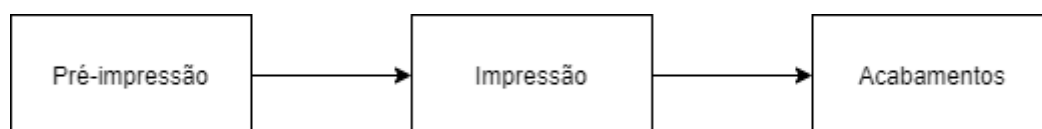


Figura 3 – Exemplo de Gama Operatória

Uma operação é realizada por um recurso e possui diversas características/atributos, como por exemplo, a cor e o tamanho do papel. Para além disto, está associada a três tempos distintos, descritos de seguida.

2.1.2.1 Tempo de preparação

O tempo de preparação é classificado como sendo o tempo necessário para efetuar a preparação para que a operação possa ser realmente produzida, aplicado, normalmente, quando existe uma troca de artigo (Dinis-Carvalho, 2008). Estas trocas de operações, estão na origem do conceito de matrizes de tempos de preparação.

2.1.2.2 Tempo de produção

Corresponde ao tempo necessário para realizar a produção de uma determinada operação, excluindo os tempos de preparação e de repouso. Ou seja, após a preparação da operação, é feita a ação de realmente produzir e este tempo em que realmente se está a produzir, é denominado de tempo de produção (Dinis-Carvalho, 2008).

2.1.2.3 Tempo de repouso

Quando uma operação termina, pode ser necessário ficar em repouso antes de poder passar à próxima etapa (operação) definida na gama operatória, a este tempo, é dado o nome de tempo de repouso.

Este tempo é necessário devido a inúmeras razões, mas um exemplo simples é o da temperatura. Ou seja, após a conclusão de uma determinada operação, o produzido pode encontrar-se a uma temperatura elevada e que é incompatível com a próxima etapa (ou

porque existe o risco de o produto ficar danificado), necessitando de ficar em repouso até atingir a temperatura ideal para passar à próxima operação.

2.1.3 Recurso

Um recurso é, por norma, o artefacto que permite o processamento das operações. Ou seja, é normalmente considerado como uma máquina (por exemplo, uma impressora), que possui um determinado horário de funcionamento (calendário).

Um recurso pode estar associado a um ou mais recursos alternativos, ou seja, a operação que está alocada para um determinado recurso também poderá ser feita num dos seus recursos alternativos. Entenda-se por recurso alternativo, recurso semelhante e compatível com outro recurso.

2.1.4 Escalonamento

Escalonar é um processo de tomada de decisão, que é utilizado frequentemente na indústria de produção. Este processo consiste em alocar recursos para as tarefas (operações) em determinados períodos de tempo e o seu objetivo é otimizar uma ou mais metas (critérios) (Pinedo, 2008).

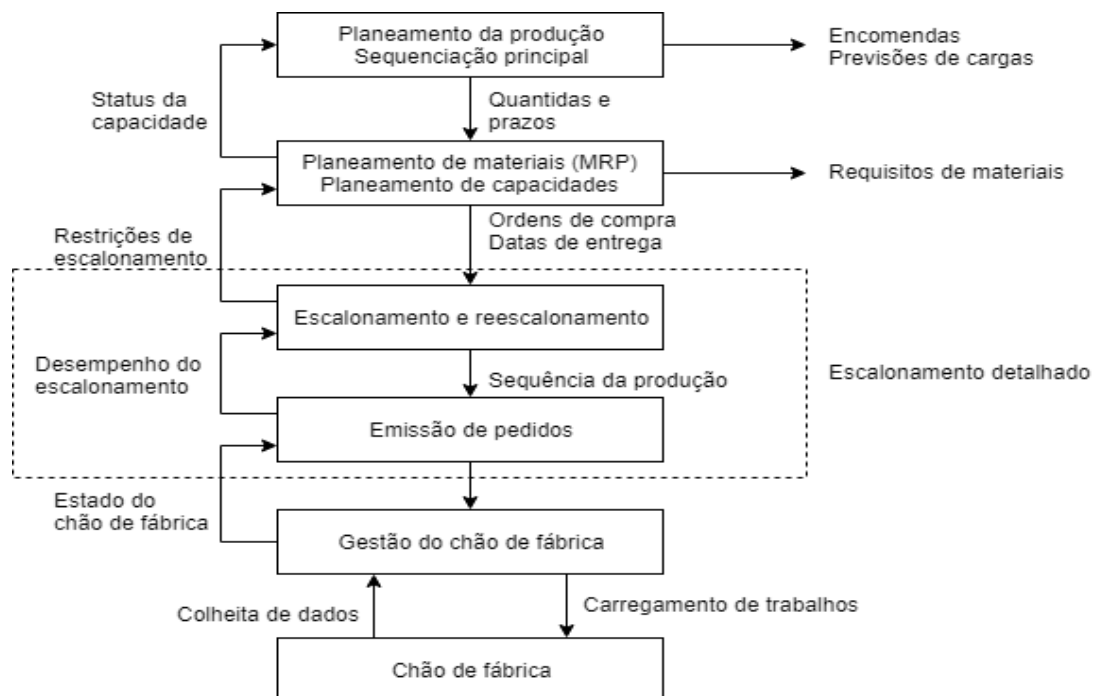


Figura 4 – Escalonamento
 Fonte: Adaptado de Pinedo, 2008

Na Figura 4, está ilustrado um exemplo de funcionamento de uma organização onde se pode verificar o papel do planeamento e os processos que afetam o mesmo. Inicialmente, observa-se a chegada de encomendas e a consequente tradução das mesmas em ordens de fabrico, sendo também geradas as restrições. De seguida, são definidos os objetivos/critérios e é realizado o planeamento e replaneamento tendo em conta esses critérios e as diferentes restrições, com a intenção de otimizar o resultado. Por fim, esses resultados são fornecidos ao chão de fábrica para que possa ser seguida a ordem do resultado otimizado na realização das ordens de fabrico/operações.

2.1.5 Balanceamento

O balanceamento de operações está incorporado no escalonamento e é responsável por alocar a operação a um dos seus possíveis recursos. Ou seja, verifica todos os recursos alternativos para a operação e decide qual o melhor recurso para a realização do trabalho. Este processo de decisão é afetado pelos critérios/objetivos definidos.

De forma a melhor se entender este processo de decisão, são apresentados, de seguida, dois exemplos simples com objetivos diferentes:

- Diminuir tempos de preparação: quando o objetivo é diminuir tempos de preparação, o balanceamento, irá tentar alocar a operação a um recurso que tenha uma operação com características semelhantes de forma a diminuir o tempo de preparação entre essas duas operações.
- Diminuir tempos de paragem: quando o objetivo é diminuir tempos de paragem, então o balanceamento irá procurar manter os recursos sempre ocupados, diminuindo assim os tempos de paragem dos recursos.

2.1.6 Matriz de Tempos de Preparação

A matriz de tempos de preparação permite determinar para cada transição qual é o respetivo tempo de preparação necessário.

No âmbito deste projeto, serão consideradas matrizes de transição para atributos de operações. Assim, tomando como exemplo a cor, poderá ser realizada uma matriz que defina o tempo de preparação associado à troca de cada cor, tal como se pode observar na Tabela 1.

Tabela 1 – Exemplo de matriz de tempos de preparação (minutos)

Cor	Preto	Branco	Vermelho	Azul	Amarelo
Preto	10	60	20	25	30

Cor	Preto	Branco	Vermelho	Azul	Amarelo
Branco	60	5	10	10	10
Vermelho	10	20	5	20	25
Azul	15	15	25	5	15
Amarelo	20	10	30	15	5

Pela análise do exemplo anterior, lê-se, por exemplo, que a troca da cor Preto para Branco, demora 60 minutos.

Devido ao elevado número de combinações existentes na matriz, torna-se possível efetuar uma otimização aos tempos de preparação, reduzindo custos.

2.1.7 MRP

É um sistema de controlo de materiais que procura assegurar que os materiais necessários para a produção das diferentes ordens de fabrico estão disponíveis e existem nas quantidades necessárias (Gallego, 2003). É um conceito que surge normalmente associado ao planeamento da produção, pois assegura a existência de um planeamento que é válido e que é coerente com a realidade.

2.2 Escalonamento do Sistrade ERP

O módulo de escalonamento da produção faz parte do Sistrade ERP e é um sistema que permite programar os trabalhos (sobretudo ordens de fabrico) de uma empresa. Este sistema em termos de apresentação gráfica, é baseado num gráfico de *Gantt* onde é possível visualizar as ordens de fabrico, os recursos e o horizonte temporal. “A programação dos trabalhos a realizar, quer sejam, produtos ou projetos, obedece a critérios de sequenciação lógicos e critérios relacionados com bom senso de quem planeia, por esse motivo é fundamental que a tarefa de planear seja feita com o auxílio de um sistema de apoio à decisão no planeamento da produção e não numa ferramenta automática de planeamento” (Sistrade Software Consulting, 2018). Como referido, o sistema de planeamento da Sistrade requer um elevado papel por parte de um planeador, com o objetivo de efetuar uma melhor otimização.

Na Figura 5 é apresentado o diagrama de *Gantt* do escalonamento do Sistrade ERP, onde é possível observar as diferentes operações das diferentes ordens de fabrico (a vermelho).

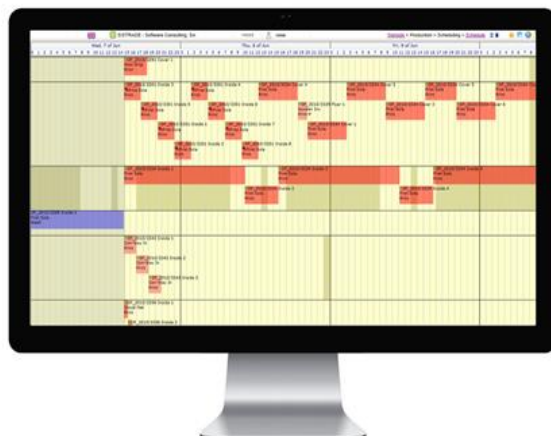


Figura 5 – *Gantt* do planeamento do ERP Sistrade¹

O resultado apresentado na figura anterior é, inicialmente, otimizado por parte do escalonador do planeamento, segundo determinados critérios, e posteriormente, refinado por parte de um planeador. De uma forma generalizada, o planeamento divide-se em dois principais componentes, que podem ser denominados de Escalonador e *Gantt*.

2.2.1 Escalonador

O escalonador é responsável por toda a lógica de escalonamento. Aplica os critérios definidos pelo utilizador e sequencia as operações das ordens de fabrico de acordo com esses critérios e com as diferentes restrições existentes.

Os critérios de planeamento são a forma de otimizar o resultado final, podendo ser considerados como objetivos. Estes critérios são parametrizáveis de acordo com a necessidade do cliente. Contudo, existem critérios que são comuns à maior parte dos clientes, como por exemplo, a data de entrega do trabalho.

Ao nível das restrições, estas garantem as regras de negócio, de forma a que o resultado final faça sentido e não existam casos de impossibilidade lógica. As restrições consideradas são:

- Garantir a gama operatória/sequência das operações;
- Garantir a dependência entre ordens de fabrico;
- Garantir que o calendário do recurso é tido em conta;

¹ <https://www.sistrade.com/pt/solucoes/gestao-producao/manufacturing-40/>

2.2.2 Gantt

Este componente é responsável por apresentar visualmente o resultado (camada gráfica) calculado pelo escalonador. Apresenta outras funcionalidades para além da apresentação visual do resultado, algumas das quais interferem diretamente com o escalonador e são relevantes para o problema em questão:

- Fixar operações num determinado momento;
- Recalcular o planeamento a partir de uma determinada data;
- Atribuir uma percentagem de conclusão à operação anterior para que a operação imediatamente a seguir possa começar (esta percentagem é de 100% por defeito);
- Alterar o tempo de preparação, produção e repouso de uma operação.

2.2.3 Avaliação

O escalonamento do Sistrade ERP não utiliza nenhum algoritmo heurístico ou similar. De uma forma simplista, o processo consiste numa sequenciação, tendo em conta todas as restrições, após a ordenação segundo os critérios definidos.

Vantagens

- Muito rápido, cerca de 3 segundos de cálculo para cerca de 2000 operações;
- Considera a capacidade dos recursos como finita;
- Critérios de otimização configuráveis;

Desvantagens

- Não considera matrizes de tempo de preparação;
- Os critérios servem apenas como ordenação, tornando a otimização por mais do que um critério ineficaz;
- Não é independente da plataforma, está dependente do Sistrade ERP;
- Não possui o conceito de otimização, mas sim apenas o de critério. Ou seja, não permite, por exemplo, otimizar tempos de preparação e, ao mesmo tempo, cumprir as datas de entrega.

- O balanceamento de carga entre recursos é “cego”, ou seja, não permite otimizar, por exemplo, por ferramenta (com o objetivo de diminuir o tempo de preparação). Apenas distribui a carga entre os recursos alternativos, sem olhar a outras possíveis otimizações.

3 Análise de Valor

Neste capítulo, será apresentada a análise de valor do projeto, com o objetivo de demonstrar o seu valor para a organização, para os clientes e para o mercado em geral.

Nesta análise, será utilizado o modelo NCD (*New Concept Development*) (Koen *et al.*, 2001), para suportar a demonstração do valor do projeto. Como suplemento e para demonstrar de uma forma simples e intuitiva o negócio, será também utilizado o modelo CANVAS (Osterwalder *et al.*, 2010).

3.1 Modelo NCD

O modelo NCD (*New Concept Development*) permite definir a fase inicial do processo de negócio e de inovação. É constituído por três componentes (Figura 6):

- Fatores de influência: são diversos fatores caracterizadores de uma organização, nomeadamente, nas suas capacidades organizacionais, na sua estratégia empresarial, e no exterior (por exemplo, canais de distribuição, clientes e concorrentes). Estes fatores afetam diretamente o processo de inovação (Koen *et al.*, 2001).
- Elementos chave: existem cinco elementos chave, são eles, identificação da oportunidade, análise da oportunidade, geração e enriquecimento de ideias, seleção de ideias e definição do conceito
- Motor: responsável por colocar em movimento/execução os cinco elementos chave, sob a liderança e cultura da organização (Koen *et al.*, 2001).

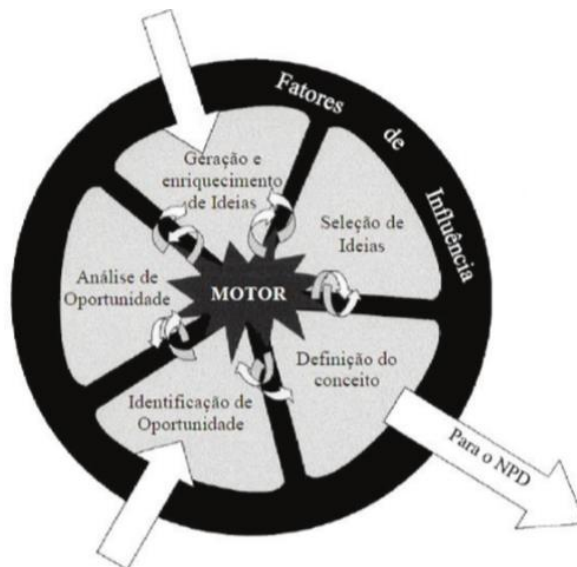


Figura 6 – Modelo NCD

Adaptação de Koen et al., 2001

3.1.1 Identificação e Análise da Oportunidade

O problema do escalonamento e planeamento dos diferentes trabalhos pelos recursos é um problema comum na indústria, que afeta diretamente a produtividade e competitividade de uma organização.

Existem dois tipos de estratégias de produção, *make-to-order* (MTO) e *make-to-stock* (MTS) (Li, 1988), respetivamente, produzir por encomenda ou produzir para stock. O problema em questão é aplica-se em ambos os casos, mas é mais notável no primeiro, no cenário MTO. Isto acontece porque existe um maior número de produções para realizar e essas produções são mais diversificadas, pois dependem de cliente para cliente. Assim, as organizações deparam-se com problemas provenientes desta abordagem, como por exemplo, o incumprimento de datas de entrega ou os elevados custos de preparação dos recursos. É importante referir que quanto maior é número de encomendas, maior será, por norma, o problema originado e maior será também a dificuldade de planear de forma eficaz e em tempo útil todas as respetivas ordens de fabrico das encomendas.

Tendo como base o problema referido anteriormente e as suas consequências, é recorrente a procura, por parte das organizações, de tentar eliminar ou reduzir essas consequências. Neste contexto, surgem os *softwares* de escalonamento e planeamento da produção. Estes *softwares* são capazes de planear e escalonar a produção segundo vários critérios, ajudando as organizações a obter uma maior produtividade e uma maior competitividade.

Posto isto, surgiu a ideia de criar um novo escalonador que seja mais eficaz e inovador e que forneça diversos ganhos às organizações. É pretendido que este escalonador seja capaz de dar resposta aos diversos problemas do escalonamento da produção e que possa ser utilizado de forma completamente isolada (sem interligação com o *software* Sistrade) por outros sistemas, pois existem vários potenciais clientes que utilizam outros sistemas.

3.1.2 Geração e Enriquecimento de Ideias

Ao longo do tempo, quando começaram a surgir as necessidades dos clientes de um escalonador mais eficaz, e quando surgiram potenciais clientes apenas para o escalonador, surgiram também algumas ideias, nomeadamente:

- Ideia 1: algoritmo inovador capaz de utilizar vários critérios e que procure as melhores soluções possíveis, ao invés de fazer apenas uma sequenciação segundo critérios;
- Ideia 2: escalonador multiplataforma, usável sem a necessidade do SistradeERP (*standalone*);
- Ideia 3: algoritmo que suporta matrizes de tempos de preparação;
- Ideia 4: algoritmo com balanceamento de carga eficaz;
- Ideia 5: algoritmo que suporte o conceito de MRP;
- Ideia 6: mecanismo de autoaprendizagem de forma a tornar as melhorias automáticas.

O conjunto de ideias apresentado anteriormente, sugere a criação de uma aplicação que seja capaz de funcionar em modo *standalone* e que o seu algoritmo seja capaz de dar resposta aos maiores problemas de planeamento da produção, através de funcionalidades que otimizam o mesmo. O processo de seleção de ideias será explicado no capítulo seguinte.

3.1.3 Seleção de Ideias

As ideias apresentadas no capítulo anterior, foram identificadas pois representam uma mais valida para o produto e para o cliente. Desta forma, as ideias 1 a 4 devem ser incluídas na solução final pois são fundamentais para o cumprimento do objetivo definido anteriormente.

A ideia 5 foi excluída para este projeto, pois recorrendo à experiência recente com diversos clientes diretos, o conceito de MRP como restrição de planeamento, não é um fator de grande importância. O que os clientes em questão pretendem, é o conceito de serem notificados caso

não tenham material disponível para a data planeada, mas, o planeamento deverá ser efetuado normalmente mesmo que (ainda) não exista material disponível.

A ideia 6 também deverá ser aplicada, mas, devido ao seu grau de complexidade, não foi identificada como uma prioridade a curto prazo.

3.1.4 Definição do Conceito

Tendo por base a análise efetuada nos capítulos anteriores, torna-se possível formular um conceito para o problema do escalonamento da produção. Assim, irá ser desenvolvida uma solução que permita cumprir os diferentes objetivos e as diferentes ideias identificadas e aprovadas, de forma a ser um conceito inovador e com valor para a organização e para os clientes. Desta forma, a solução passa pela criação de um *software* que funcione de forma completamente autónoma, independentemente do sistema onde irá ser integrado. Este *software* tem que cumprir os seguintes requisitos:

- Todos os conceitos que já existem no *software* Sistrade devem ser aplicados, à exceção de conceitos que sejam demasiado específicos ou que não se considerem como uma mais valia;
- Algoritmo totalmente novo e inovador, capaz de otimizar diferentes critérios/objetivos com uma maior percentagem de sucesso do que o atual;
- Implementação do conceito de matrizes de tempos de preparação e de balanceamento de carga.

3.2 Valor para o Cliente

Tendo por base o conceito de negócio definido anteriormente, é possível concluir que existem diversos ganhos e vantagens, numa perspetiva do cliente, com a produção deste *software*. Assim, os benefícios para o cliente provenientes desta solução são:

- Não existe a necessidade de adquirir *software* adicional pertencente ao Sistrade ERP;
- Maior número de funcionalidades e maior flexibilidade no escalonamento, a pensar nos diversos tipos de negócio existentes;
- Maior taxa de otimização, segundo os diversos critérios/objetivos, que contribui para um aumento da competitividade e para uma melhoria no cumprimento de metas definidas pela organização.

Por outro lado, devido à maior complexidade do algoritmo e conseqüentemente à obtenção de melhores resultados, existe um possível sacrifício que terá que ser feito. Esse sacrifício encontra-se no tempo de execução necessário pelo *software* para escalonar as diversas ordens de fabrico. Embora o objetivo é que esse tempo não seja elevado, deverá ser sempre um pouco superior ao escalonador atual do Sistrade ERP.

3.3 Proposta de Valor

O presente capítulo, tem como objetivo apresentar de uma forma sumarizada a proposta de valor da solução. Esta proposta de valor é baseada em toda a informação apresentada anteriormente, enumerada em diversos capítulos.

Com o objetivo de alargar os potenciais clientes, a solução a desenvolver será do tipo *standalone*, ou seja, poderá ser utilizada independentemente do ERP que o cliente utilize. Esta solução irá manter todos os conceitos com valor provenientes da solução atual do Sistrade ERP. Será produzido um novo algoritmo, mais apelativo, que permite uma melhor otimização de critérios/objetivos, bem como, uma maior flexibilidade na aplicação de cada um desses critérios. Por fim, a solução proposta terá uma maior diversidade de funcionalidades, nomeadamente, matrizes de tempos de preparação e o conceito de balanceamento.

Esta solução tem como principal objetivo permitir aos clientes uma melhoria significativa no seu processo de escalonamento/planeamento da produção, de forma a conseguirem cumprir os objetivos por si propostos.

3.4 Modelo de Negócio

O modelo de negócio pode ser representado com recurso ao modelo *Canvas* (Osterwalder *et al.*, 2010), com o objetivo de tornar a sua perceção mais fácil e intuitiva. Este modelo possui nove diferentes tópicos que permitem fazer o planeamento estratégico de um negócio. Cada um destes tópicos está apresentado na Tabela 2.

Como parceiros principais, identificam-se clientes da Sistrade e potenciais clientes (*prospects*), uma vez que, são os potenciais utilizadores do resultado deste projeto. Para além disso, poderão também ser úteis na fase de teste do projeto.

O escalonamento da produção segundo critérios e com funcionalidades de balanceamento e de matriz de tempos de preparação, são consideradas as atividades chave. Desenvolveres,

serão responsáveis pela tarefa de implementação das atividades chave. Por outro lado, de forma a suportar o processo, são necessários servidores e uma equipa de apoio ao cliente.

O principal componente da proposta de valor é um novo escalonador, totalmente independente da plataforma, que efetua otimizações segundo critérios e que incorpora novas funcionalidades apelativas. Associado a isto, existirá ainda uma equipa de apoio ao cliente para esclarecimento de dúvidas e resolução de problemas.

A relação com o cliente ficará a cargo, em primeira instância, do departamento comercial. Posteriormente, ficará a cargo de um gestor de projeto, que será responsável não só por gerir a relação com o cliente, mas também todo o decorrer do mesmo. Como complemento, a equipa de apoio ao cliente também terá um certo impacto na temática em questão, uma vez que, são a linha de contacto direto para esclarecimentos e resolução de problemas.

Relativamente a clientes, podem ser considerados todos aqueles que foram identificados para as parcerias principais, com especial atenção para a captação de organizações que necessitam de ferramentas de escalonamento da produção. Como canais de propagação, foram identificados o website da Sistrade, demonstrações e feiras, e por fim, emails de *newsletter*.

Em termos de custos, destacam-se, em primeiro lugar, o custo de desenvolvimento e os custos de hospedagem do serviço. Posteriormente, existe também o custo relacionado com a equipa de apoio ao cliente. Por outro lado, a receita seria obtida através da venda de licenças para uso do *software* produzido.

Tabela 2 – modelo negócio (canvas)

<p>Parcerias principais (<i>Key partners</i>)</p> <p>Cientes Sistrade; <i>Prospects</i>; Organizações que necessitam de planejar a produção.</p>	<p>Atividades chave (<i>key activities</i>)</p> <p>Escalonamento da produção segundo critérios e com diversas funcionalidades, nomeadamente, balanceamento de cargas, matrizes de preparação.</p>	<p>Proposta de valor (<i>key propositions</i>)</p> <p>Escalonamento da produção, otimizado segundo critérios, que permite aumentar a competitividade da organização, recorrendo ainda a novas funcionalidades; Software independente do sistema; Apoio ao cliente para resolução de problemas.</p>	<p>Relacionamento com clientes (<i>customer relationships</i>)</p> <p>Gestor de projeto; Apoio ao cliente; Departamento comercial.</p>	<p>Segmento de clientes (<i>customer segments</i>)</p> <p>Cientes da Sistrade; <i>Prospects</i>; Organizações que necessitam de planejar a produção.</p>
	<p>Recursos principais (<i>key resources</i>)</p> <p>Desenvolvedores; Apoio ao cliente; Servidores.</p>		<p>Canais (<i>channels</i>)</p> <p>Website Sistrade; Demonstrações; feiras; Email.</p>	
<p>Estrutura de custos (<i>cost structure</i>)</p> <p>Custo de hospedagem do serviço; Equipa de desenvolvimento e de apoio ao cliente.</p>			<p>Fontes de receita (<i>revenue streams</i>)</p> <p>Venda da licença do <i>software</i> a clientes já existentes ou a novos clientes.</p>	

4 Estado da Arte

Este capítulo pretende descrever de forma objetiva quais as principais abordagens existentes sobre algoritmos de escalonamento, sendo também efetuada uma avaliação e comparação destas abordagens e selecionadas as que podem/devem ser adotadas. São ainda apresentadas as tecnologias mais relevantes para a resolução do problema global. Irá incidir, principalmente, sobre a temática de algoritmos de escalonamento pois é a principal vertente deste projeto.

4.1 Soluções/Abordagens Existentes

Neste subcapítulo, serão apresentadas algumas das soluções mais relevantes para o problema em questão. É importante notar que os algoritmos de escalonamento são uma “caixa preta” para as organizações, tornando o processo de adquirir informação sobre os mesmos uma tarefa praticamente inexecutável. Assim, para contornar este problema, serão apresentadas várias soluções teóricas formuladas por diversos autores que dão uma possível resposta ao problema.

É ainda importante reter que a solução obtida por cada um dos algoritmos tem que ser determinada em tempo útil. Ou seja, não serão analisados algoritmos que obtêm soluções ótimas num período de tempo demasiado longo e que tornam a sua utilização inútil para uma organização de forma produtiva.

Por forma a ser mais fácil a referência a cada algoritmo, são apresentados sequencialmente e devidamente numerados.

Algoritmo 1, 2 e 3, respetivamente (Kolisch and Heß, 1998), o tema do planeamento foi abordado tendo em conta múltiplas máquinas, e em larga escala, com restrições de materiais e de disponibilidade de recursos em organizações que produzem por encomenda. Foram

introduzidos três algoritmos heurísticos e feita a sua experimentação e avaliação. Os algoritmos introduzidos foram:

- “Biased Random Sampling”(Kolisch and Heß, 1998): de uma forma generalizada, este algoritmo gera de forma probabilística, para obter soluções diferentes, um conjunto de soluções para o problema. Para cada uma das soluções vai aplicar uma função objetivo, que determina um valor para a solução, com o objetivo determinar a melhor solução.
- “API-Neighborhood” e “Critical Neighborhood”(Kolisch and Heß, 1998): ambos os algoritmos são baseados no algoritmo *tabu search* (capítulo 4.2.1). No primeiro, e mais simples, a vizinhança é resultante de uma troca adjacente par a par. Quanto ao segundo algoritmo, assumindo que existe uma operação que está a ser impedida de ser planeada por um conjunto de operações, a vizinhança é obtida através da troca dessa operação com a primeira operação da lista de dependentes (em termos de ordem).

Algoritmo 4 (Dellaert, Jeunet and Jonard, 2000), os autores propuseram um algoritmo baseado em algoritmos genéticos, que integra ainda o conceito MRP. Neste algoritmo, foram criados cinco operadores genéticos que asseguram que a pesquisa ocorre num conjunto de soluções viáveis.

Algoritmo 5 (Moon *et al.*, 2006), foi proposto um algoritmo que responde ao problema do escalonamento da produção baseado numa solução híbrida entre algoritmos genéticos e algoritmos de pesquisa local. Quanto ao primeiro tipo, foi utilizado para pesquisar globalmente todo o espaço pesquisa. O segundo tipo, foi utilizado para pesquisar localmente aquando da geração de novos descendentes e antes de estes fazerem parte da nova população. Este algoritmo híbrido contempla vários conceitos do escalonamento da produção, destacando-se: tempos de preparação, capacidade finita dos recursos e recursos alternativos.

Algoritmo 6 (Madureira, 2013), é composto por um sistema multiagente que utiliza algoritmos genéticos e algoritmos do tipo *tabu search*. Nesta abordagem, é possível escolher um dos dois tipos de algoritmos referidos anteriormente. A solução contempla os conceitos de recursos alternativos, dependências entre operações e de datas de entrega. Ainda nesta abordagem, é introduzido o tema de escalonamento com base em soluções híbridas, nomeadamente, da possibilidade de utilização de uma combinação entre algoritmos genéticos e algoritmos do tipo *tabu search*.

Algoritmo 7 (Kljajić, Bernik and Breskvar, 2014), os autores definiram uma metodologia um pouco diferente das apresentadas anteriormente, recorrendo a um algoritmo que suporta múltiplos critérios/objetivos que é baseado em algoritmos genéticos e em simulações visuais. O algoritmo baseado nos algoritmos genéticos, é responsável por gerar várias soluções para o

problema e cada uma dessas soluções é apresentada visualmente, juntamente com alguns indicadores, para o planejador decidir qual a que melhor se adequa ao pretendido.

Algoritmo 8 (Palacios *et al.*, 2015), foi desenvolvido um algoritmo para responder ao problema do escalonamento da produção em que o grande objetivo/critério é diminuir o tempo total necessário para finalizar todas as ordens de fabrico. Este algoritmo é composto por três outros algoritmos:

- Algoritmo heurístico: responsável por gerar a solução inicial;
- Algoritmo genético: é aplicado para gerar várias soluções através da criação de várias gerações;
- *Tabu search*: é aplicado a cada solução gerada, com o objetivo de encontrar uma solução melhor, sendo que, quando é encontrada uma solução melhor, é efetuada uma reconstrução da solução de forma a que essas características sejam transmitidas a gerações futuras.

Algoritmo 9 (Moses and Sangplung, 2017), o autor definiu um algoritmo para o problema do escalonamento da produção em ambientes onde são feitas várias encomendas específicas e com características próprias. Este algoritmo tem a vantagem de olhar para a capacidade finita dos recursos, de planejar as operações tendo em conta a data de entrega da ordem de fabrico e de estar preparado para cenários mais exigentes, com um elevado número de operações e restrições. É baseado no algoritmo *tabu search* e consiste em duas fases distintas:

- Primeira fase: inicialmente, calcula uma possível solução inviável/inválida, mas que seria ótima se não existissem as restrições da capacidade. De seguida, modifica a solução encontrada anteriormente de forma a que esta seja válida em termos de capacidade dos recursos.
- Segunda fase: a partir da solução encontrada na primeira fase, é efetuada a procura por uma solução melhor, tendo em conta as restrições de capacidade. Nesta procura, o algoritmo tenta alocar as operações o mais perto possível (nem mais cedo, nem mais tarde) da sua data de entrega.

Como forma de resumo para literatura apresentada anteriormente, é disponibilizada a Tabela 3. Nesta tabela, pode-se observar quais os algoritmos utilizados em cada abordagem. Os algoritmos encontrados que apareceram de forma isolada, foram considerados como “Outro Algoritmo” por não serem relevantes para a estatística apresentada.

Tabela 3 – Comparação das abordagens existentes

Abordagem	Algoritmo Tabu search	Algoritmo Genético	Outro Algoritmo
Algoritmo 1			X
Algoritmo 2	X		
Algoritmo 3	X		
Algoritmo 4		X	
Algoritmo 5		X	X
Algoritmo 6	X	X	
Algoritmo 7		X	
Algoritmo 8	X	X	
Algoritmo 9	X		

4.2 Algoritmos de Otimização

Existem diferentes tipos de algoritmos, de entre os quais se destacam, algoritmos baseados em pesquisas locais e algoritmos baseados na evolução biológica dos organismos. Estes dois tipos de algoritmos possuem algoritmos que são adequados para o problema do escalonamento da produção.

Com base no levantamento realizado na secção anterior (4.1) e na Tabela 3, foram selecionados dois tipos de algoritmos, de entre os mais utilizados, nomeadamente o *tabu search* e os algoritmos genéticos.

4.2.1 Pesquisa local - *Tabu Search*

Os métodos de pesquisa local são algoritmos iterativos que exploram as diversas soluções possíveis através da passagem, passo a passo, de uma solução para outra. São iniciados a partir de uma solução que é escolhida de forma arbitrária e a transição de uma solução para outra solução é feita de acordo com algumas modificações, que dependem de caso para caso (Widmer, Hertz and Costa, 2010).

O *tabu search* é um algoritmo que por ser considerado como “uma meta-heurística sobreposta noutra heurística” (Gendreau, 2006).

A eficiência dos processos de pesquisa é muito importante para que se possa encontrar a melhor solução e em tempo útil. Assim, de forma a melhorar esta eficiência, o *tabu search*, monitoriza não só a informação local (o melhor valor atual encontrado), mas também a informação relacionada com o processo de pesquisa/exploração (detalhe das últimas soluções encontradas). Esta informação será utilizada na passagem para a solução seguinte (Hertz *et al.*, 2000). Esta é a principal característica deste método de pesquisa, sendo também o grande diferenciador entre este método e os restantes métodos de pesquisa. De uma forma simples, este algoritmo guarda a cada iteração uma lista de soluções tabu e fica proibido de visitar essa lista de soluções (de considerar uma solução dessa lista).

De forma a se entender melhor o algoritmo, é fundamental perceber os seguintes conceitos (Widmer, Hertz and Costa, 2010):

- **Função de aspiração:** durante o processo de escolha da melhor solução, pode ser necessário escolher entre vários candidatos que possuem o mesmo valor. Se o vizinho escolhido não conduzir a uma boa região do espaço de soluções, pode então ser necessário voltar a uma solução visitada, mesmo que agora ela faça parte da lista tabu, com o objetivo de explorar uma nova região vizinha. Assim e por este motivo, é necessária a função de aspiração. Quando uma solução vizinha pertence à lista de tabu e é válida face aos critérios de aspiração, então o seu estado de tabu é removido e a solução passa a ser considerada uma possível solução válida aquando do processo de seleção.
- **Vizinhança:** dado um conjunto de objetos e um conjunto de recursos e tendo em conta que exatamente um recurso deve ser atribuído a cada objeto, pode-se definir a vizinhança de uma solução como o conjunto de soluções que podem ser obtidas a partir da solução atual, alterando o recurso atribuído de um objeto.
- **Condição de paragem:** é a condição que faz com que o algoritmo pare.

De uma forma genérica, na Figura 7 (Widmer, Hertz and Costa, 2010), está representado um exemplo do algoritmo do tipo *tabu search*.

```

Seleção da solução inicial  $s \in X$ ;
atribuir  $s^* \leftarrow s$ ;
atribuir  $nbiter \leftarrow 0$ ;
atribuir  $T \leftarrow \emptyset$ ;
inicializar a função de aspiração  $A$ ;
atribuir  $best\_iter \leftarrow 0$ ;
ciclo enquanto  $(F(s) > f)$  e  $(nbiter - best\_iter < nbmax)$  então
    atribuir  $nbiter \leftarrow nbiter + 1$ ;
    gerar um subconjunto  $N' \subseteq N(s)$  de soluções vizinhas de  $s$ ;
    escolher a melhor solução  $s' \in N'$  em que  $F(s') \leq A(F(s))$  ou  $s' \notin T$ ;
    atualizar a função aspiração  $A$  e a lista de tabu  $T$ ;
    atribuir  $s \leftarrow s'$ ;
    se  $F(s) < F(s^*)$  então atribuir  $s^* \leftarrow s$  e  $best\_iter \leftarrow nbiter$ 
fim do ciclo

```

Figura 7 – Algoritmo tabu search

De forma a simplificar o algoritmo anterior, foram simplificados alguns termos. Os termos para interpretação do algoritmo anterior, estão presentes na Tabela 4.

Tabela 4 – Termos do algoritmo tabu search

Termo	Descrição
s	solução atual
s^*	melhor solução
$nbiter$	contador das iterações
T	lista de tabu
$best_iter$	número da iteração onde foi encontrada melhor solução s^*
X	conjunto de soluções
F	função objetivo
$N(s)$	vizinhança de uma solução $s \in X$
A	função de aspiração
$nbmax$	número máximo de iterações entre duas melhorias de s^*
N'	subconjunto de $N(s)$

Termo	Descrição
f	limite inferior aceitável para o valor objetivo

4.2.2 Abordagem Evolutiva - Algoritmos Genéticos

Este tipo de algoritmos foi introduzido por Holland (Holland, 1975) e é baseado na evolução dos seres vivos e no fenómeno da adaptação, fundamentalmente, nas teorias de Darwin. É um algoritmo de pesquisa que se baseia em mecanismos de seleção natural, defendendo a ideia que os organismos mais aptos têm maior probabilidade de sobrevivência do que os menos aptos (Rodrigues, 2000).

Mais detalhadamente, este algoritmo consiste na passagem de uma população de cromossomas (soluções) para uma nova população, através de vários operadores, com o intuito de obter uma boa solução. Cada cromossoma é constituído por genes e cada um destes genes é uma instância de um alelo em particular. O alelo define um valor possível, como por exemplo, um simples 0 ou 1. O operador de seleção é responsável por escolher os pares de cromossomas que se vão reproduzir, tendo em conta que os cromossomas mais aptos têm uma maior tendência para produzir mais descendentes que os menos aptos. Através da reprodução (operador de cruzamento/recombinação), são trocadas duas partes de cada par de cromossomas, gerando um novo par de cromossomas (Mitchell, 1995). De forma a que diversidade da população seja assegurada, pode ser aplicado o operador de mutação que altera alguns dos valores dos alelos. Este mecanismo de mutação tem associada uma probabilidade de ocorrência, sendo normalmente baixa.

De uma forma genérica, um algoritmo genético pode ser definido como (Rodrigues, 2000):

1. Geração dos cromossomas que representam a população inicial;
2. Avaliação dos cromossomas;
3. Seleção dos pares de cromossomas;
4. Reprodução, recombinação e mutação para os pares de cromossomas selecionados;
5. Verificação dos critérios de finalização. Se os critérios forem atingidos, então o algoritmo é considerado como finalizado. Se os critérios não forem atingidos, então serão repetidos os passos a partir do passo número 2 (avaliação dos cromossomas).

O ciclo definido anteriormente está representado na Figura 8, de uma forma muito simplista com o objetivo de tornar a sua compreensão fácil e intuitiva.

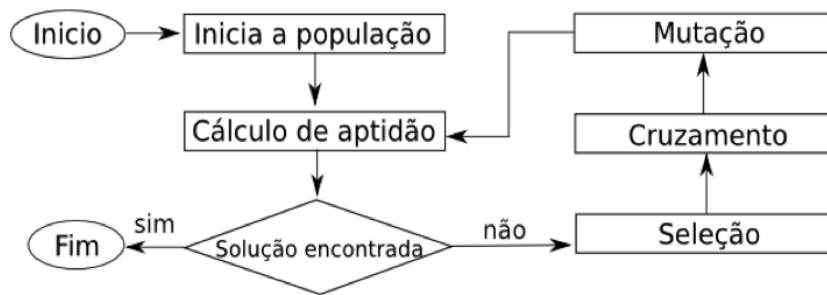


Figura 8 – Algoritmo genético (José Lindenberg, 2016)

4.3 Avaliação de Soluções/Abordagens Existentes

Neste subcapítulo, serão avaliadas as diferentes abordagens, que foram identificadas como relevantes na realização das soluções/abordagens existentes (4.1), seguindo a mesma numeração/identificação utilizada, que dão resposta ao problema do escalonamento da produção. Essa avaliação, irá servir de suporte para a seleção da abordagem a ser utilizada.

Algumas abordagens, por apresentarem características semelhantes com outras abordagens, ficaram excluídas deste comparativo, de forma a existir uma diversidade de soluções.

Para cada algoritmo, serão apresentadas as suas principais vantagens ou pontos fortes e as suas desvantagens.

4.3.1 Algoritmo 5

O algoritmo proposto é baseado numa solução híbrida entre algoritmos genéticos e algoritmos de pesquisa local.

Vantagens

- Probabilidade de obter uma solução ótima é muito elevada, entre 94% e 100%, segundo o demonstrado no artigo;
- Tem em conta tempos de preparação e tempos de transporte;
- Capacidade finita dos recursos.

Desvantagens

- Necessita de algum tempo de processamento, segundo o estudo contido no artigo, cerca de 76 segundos para 80 operações. Contudo, é importante referir que este estudo é de 2006 e o hardware evoluiu.

- O seu único critério é o de minimizar o tempo total necessário para efetuar toda a produção (*makespan*).

4.3.2 Algoritmo 6

Esta solução utiliza dois algoritmos distintos. Um do tipo *tabu search* e outro baseado em algoritmos genéticos. É possível selecionar qual dos algoritmos se pretende utilizar ou uma solução híbrida constituída por estes dois algoritmos.

Vantagens

- Permite planejar vários recursos;
- Utiliza o conceito de restrições entre operações;
- Permite seleção do algoritmo a aplicar;
- Utiliza o conceito de datas de entrega;
- Utiliza o conceito de recursos alternativos.

Desvantagens

- Este algoritmo encontra-se presente numa patente, que não contém experimentações nem resultados e, desta forma, não é possível avaliar a solução tendo por base os resultados obtidos.

4.3.3 Algoritmo 8

Este algoritmo é constituído por outros três algoritmos, um algoritmo heurístico, um algoritmo genético e um algoritmo do tipo *tabu search*.

Vantagens

- Tem em conta as restrições de capacidade dos recursos;
- Conceito de recursos alternativos aplicado;
- Utiliza critérios de otimização, nomeadamente, diminuir o *makespan*.

Desvantagens

- O seu tempo de cálculo é moroso, para uma instância com 10 ordens de fabrico, 10 recursos e 80 operações, demora entre 51 segundos a 53 segundos. Contudo, é importante referir que todos os recursos são considerados como alternativos e alterações nos parâmetros dos algoritmos poderiam reduzir estes tempos de cálculo.

4.3.4 Algoritmo 9

Este algoritmo é baseado no tipo *tabu search* e é constituído por duas fases. Na primeira é calculada uma solução ótima onde as restrições de capacidade não são tidas em conta. Na segunda, são aplicadas as restrições e é feita uma otimização para se tentar obter a melhor solução dentro do tempo computacional definido.

Vantagens

- Utiliza o conceito de capacidade finita dos recursos;
- Utiliza o conceito de prazo/data de entrega;
- Permite definir um tempo máximo de execução.

Desvantagens

- O seu tempo de cálculo é consideravelmente afetado à medida que as restrições entre operações aumentam;
- O conceito de capacidade finita dos recursos é definido como uma espécie de *slots*, ou seja, um intervalo de tempo em que o recurso está disponível é considerado um *slot*. O problema deste conceito é que o algoritmo não permite que uma operação seja planeada em dois *slots* ao mesmo tempo, ou seja, por exemplo, começar no *slot* A e depois ser acabada no *slot* B (porque entre eles existe um período em que o recurso não está disponível).

4.3.5 Conclusão

Nos subcapítulos anteriores, foram apresentadas as principais vantagens e desvantagens dos algoritmos mais relevantes previamente identificados.

Os vários algoritmos estudados apresentam como principal critério de otimização a minimização do *makespan*. Todos os algoritmos têm em conta o conceito de restrições de operações. Em termos de restrição de capacidade dos recursos, praticamente todos os algoritmos fazem uso deste conceito.

Os algoritmos analisados, conseguem bons resultados no capítulo dos indicadores e das otimizações, de acordo com os estudos efetuados pelos autores.

De uma forma geral, o Algoritmo 8 e o Algoritmo 9, conseguiram obter resultados muito satisfatórios, de acordo com os estudos realizados pelos autores, e aplicam diversos conceitos importantes e referenciados para um bom escalonamento da produção.

4.4 Soluções/Abordagens a Adotar

Com base na avaliação das abordagens, pode-se verificar que cada algoritmo tem vantagens apelativas e que são soluções robustas para o problema. Assim, neste capítulo, serão definidas as duas principais abordagens que poderão servir de ponto de partida para se obter a solução final.

4.4.1 Abordagem Principal

Esta primeira abordagem é baseada no Algoritmo 8 (Palacios *et al.*, 2015). Este algoritmo é constituído por três algoritmos, descritos de seguida e representados na Figura 9.

No primeiro algoritmo (algoritmo 1), é efetuada a geração da solução/população inicial.

No segundo algoritmo (algoritmo 2), são geradas novas soluções/populações fazendo uso de um algoritmo genético.

No terceiro algoritmo (algoritmo 3), é baseado em *tabu search* e é aplicado sempre que é gerada uma nova população. O objetivo é encontrar uma melhor solução do que aquela que foi gerada pelo segundo algoritmo. Quando é encontrada uma solução melhor, então existe uma reconstrução da solução para que as características que tornam a solução melhor sejam passadas às gerações futuras.

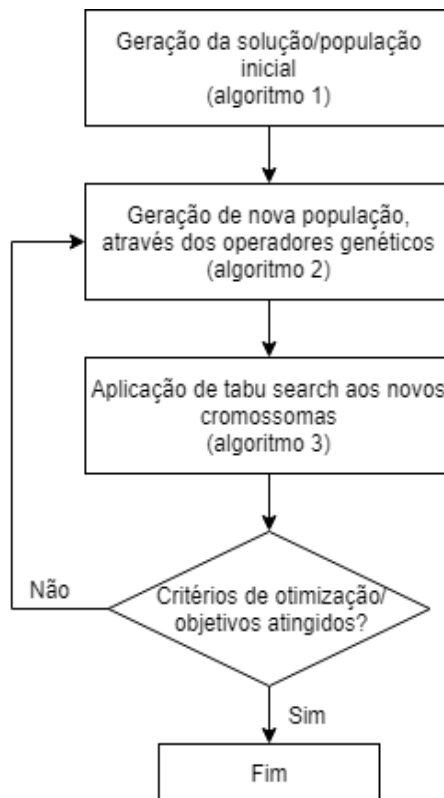


Figura 9 – Abordagem principal
 Fonte: Adaptado de Palacios et al., 2015

4.4.2 Abordagem Alternativa

Esta segunda abordagem, é baseada no Algoritmo 9 do tipo *tabu search* de duas fases (Moses and Sangplung, 2017).

Na primeira fase, inicialmente, é efetuado o cálculo de uma possível solução inválida, mas que seria ótima se não existissem as restrições da capacidade. De seguida, a solução encontrada é modificada de forma torná-la válida em termos de capacidade dos recursos.

Na segunda fase, partir da solução encontrada na primeira fase, é efetuada a procura por uma solução melhor, tendo em conta as restrições de capacidade e os critérios definidos.

Na Figura 10, é apresentado, na forma de gráfico, o algoritmo descrito anteriormente.

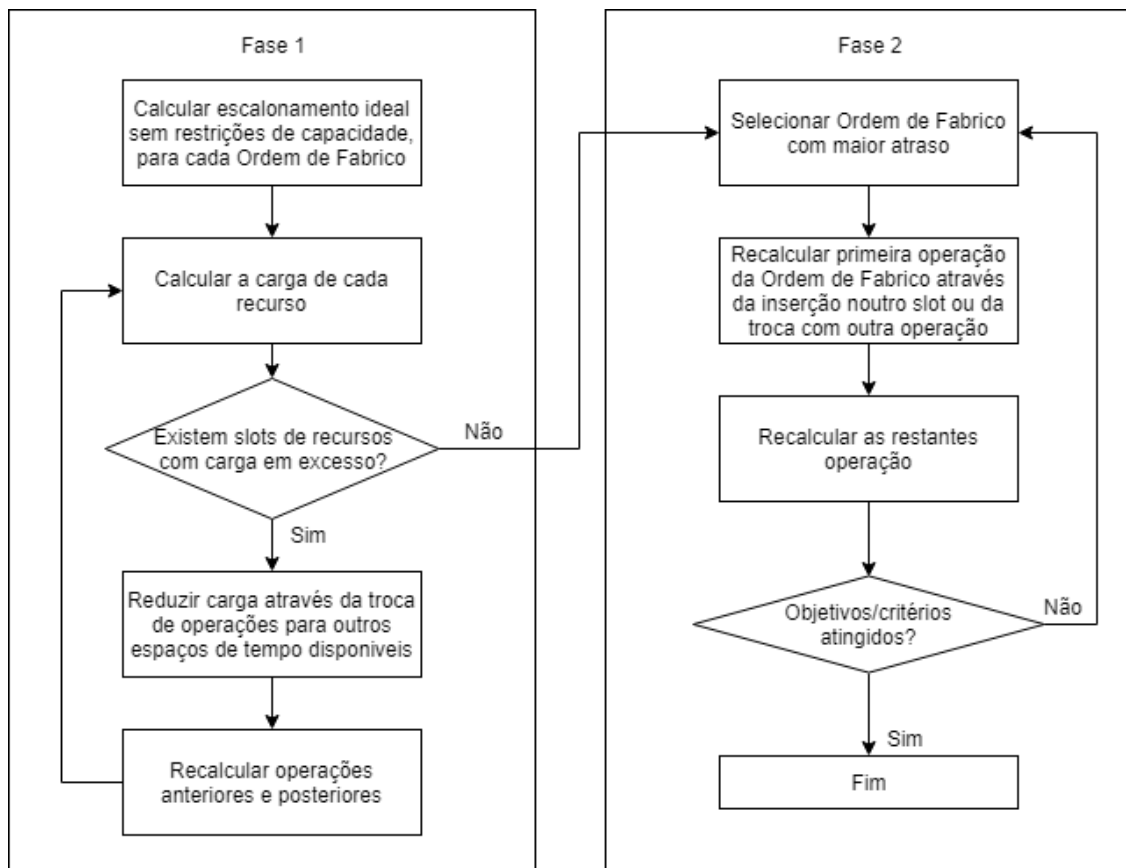


Figura 10 – Abordagem alternativa
 Fonte: Adaptado de Moses e Sangplung, 2017

Relembrando que uma das grandes desvantagens deste algoritmo era o conceito de *slots*, este conceito tem de ser “eliminado” ou adaptado, com o objetivo de permitir construir uma solução mais viável.

4.5 Tecnologia Relevante

Um dos objetivos deste projeto é possa ser utilizado por diferentes softwares, com a finalidade de poder ser integrado em ambientes ERP. Para tal, existem diversas tecnologias que podem ser utilizadas para realizar esta tarefa. Neste subcapítulo serão apresentadas algumas dessas tecnologias mais relevantes.

As tecnologias identificadas baseiam-se nas normas estipuladas pela Sistrade e, como tal, sempre que possível, são selecionadas tecnologias disponibilizadas pela Microsoft.

4.5.1 API

Como referido, a solução adotada tem que ser capaz de ser integrada por outros softwares. Esta integração deve ser simples e rápida, reduzindo ao máximo a necessidade de produção de mais software para a integração ser bem-sucedida. Tendo em conta esta premissa, uma das formas de fazer esta comunicação é através da internet, através de pedidos do tipo HTTP(s). Desta forma, evita-se a necessidade de instalações específicas por software, pois uma instalação poderá ser utilizada por vários softwares. Esta solução evita ainda a complicação do processo de integração pois não é embutida no próprio software, ou seja, o software que a pretende integrar apenas precisa de realizar invocações (pedidos) à mesma. Este processo de invocação ocorre de forma muito simples (Figura 11): os dados são enviados para a API, que efetua o seu processamento e devolve o resultado final.

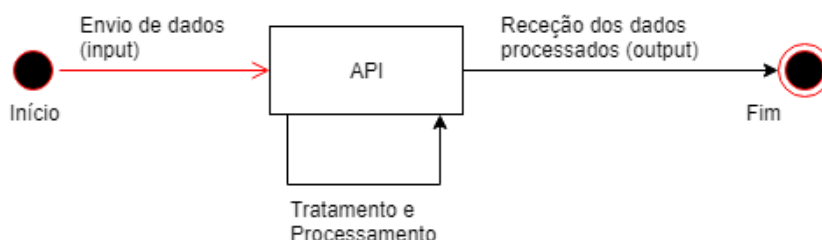


Figura 11 – *Workflow* de pedido a uma API

Uma *framework* que suporta este tipo de pedidos, é o ASP.NET Core da Microsoft em conjunto com a sua capacidade para produzir aplicações do tipo WEB API (Roth, Rick and Luttin, 2019). As aplicações do tipo WEB API permitem a integração entre sistemas e permitem a criação de processos para realizar determinada ação, neste caso, a ação de escalonar os dados recebidos. Estas aplicações depois de criadas, são publicadas num determinado endereço que é acedível pela internet.

A tarefa de escalonar é uma tarefa exigente em termos de processamento de máquina e como tal, é fundamental a utilização de tecnologias vantajosas e que permitam reduzir os tempos de processamento. A *framework* ASP.NET Core permite, através de alguns mecanismos, diminuir esses tempos de processamento. Um dos seus pontos fortes, é a possibilidade de criar *Threads* que são executadas de forma paralela, podendo assim ser feita mais do que uma tarefa ao mesmo tempo. Por consequência, origina uma redução dos tempos de processamento.

4.5.2 Tratamento de Dados de *Input* e *Output*

Os dados de input recebidos por uma API podem ser de vários formatos. Dependendo da arquitetura a ser utilizada e dependendo do software que irá utilizar essa API, devem ser definidos e suportados esses formatos.

Dois dos principais formatos de dados utilizados são o JSON e o XML (Adam DuVander, 2013). Olhando a esta premissa, e de forma a permitir uma maior flexibilidade no que diz respeito ao formato dos dados, é necessária a existência de um serviço capaz de receber vários formatos de dados, nomeadamente, JSON e XML e que os interprete e trate para depois poderem ser processados. No fim do processamento, também é necessário formatar a resposta para corresponder ao tipo de dados de input.

O problema descrito anteriormente, pode ser contornado com recurso à tecnologia ASP.NET CORE. Esta tecnologia permite definir o formato de dados de input e de output para cada pedido de forma simples e rápida. Através desta tecnologia, é evitada a necessidade do uso de uma outra solução mais robusta para um processo que é simples.

5 Análise e Design

Neste capítulo, serão apresentados os requisitos funcionais e não funcionais da solução e será apresentado o design arquitetural do *software* a ser implementado.

5.1 Análise de Requisitos

No presente subcapítulo, serão apresentados os atores e partes interessadas, os requisitos funcionais e não funcionais identificados e inerentes ao projeto.

5.1.1 Atores do Sistema

Foram identificadas as seguintes partes interessadas:

- Planeador: pretende que o software seja capaz de calcular uma solução em tempo útil e com um elevado nível de otimização;
- Administração: pretende que o *software* permita obter ganhos e uma maior competitividade.

Embora todas as partes interessadas possam ser atores, no âmbito deste projeto, foram agrupadas num único conceito, o conceito de planeador. Assim, o planeador é definido como qualquer utilizador que utilize o sistema desenvolvido.

5.1.2 Requisitos Funcionais

A partir do problema e do estado da arte, foram definidos os seguintes requisitos funcionais:

1. Efetuar o escalonamento da produção

2. Efetuar o escalonamento da produção a partir de uma determinada data
3. Consultar um histórico dos últimos escalonamentos efetuados, através da apresentação de indicadores para cada escalonamento
4. Disponibilizar o resultado de um escalonamento do histórico
5. Simular vários escalonamentos, e respetiva apresentação de indicadores para cada escalonamento

Na Figura 12 está representado o diagrama de casos de uso, elaborado a partir dos requisitos funcionais definidos.

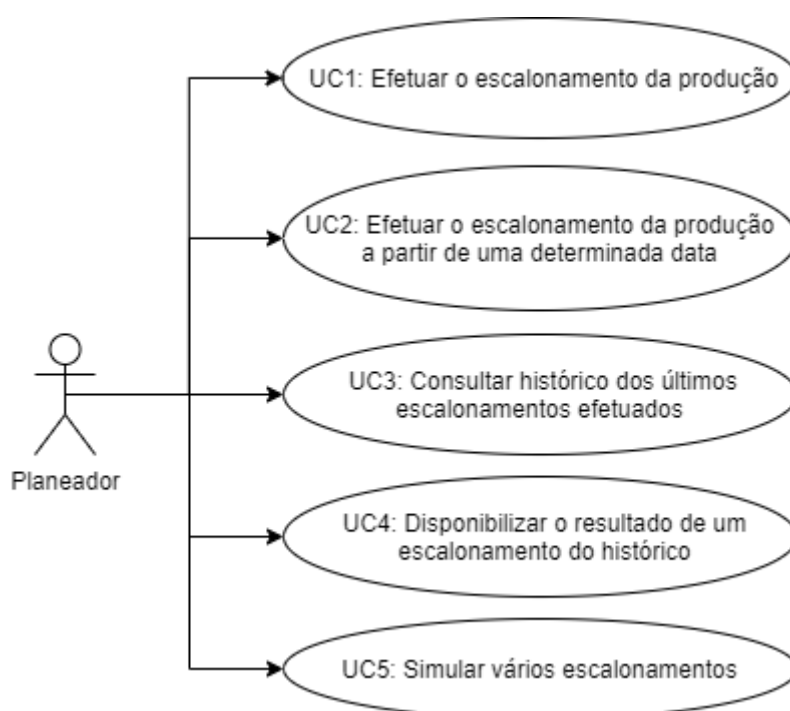


Figura 12 – Diagrama de casos de uso

5.1.3 Requisitos Não Funcionais

Existem requisitos que devem ser tidos em conta independentemente da abordagem seguida. Estas funcionalidades permitem obter uma solução mais diversificada e com maior qualidade, sendo deste modo uma contribuição direta para o aumento do valor para o cliente. Assim, de seguida, são apresentados os requisitos mais relevantes que devem ser aplicados:

- Restrições entre operações;
- Recursos com capacidade finita;

- Matriz de tempos de preparação;
- Critérios de otimização;
- Recursos alternativos;
- Balanceamento de carga.

Por outro lado, existem ainda outros requisitos para garantir que o software produzido deve respeitar as regras, as restrições e os atributos de qualidade definidos. Baseado no modelo FURPS+ (Eeles, 2004), os requisitos definidos são:

- Permitir pedidos e respostas tanto em JSON como em XML;
- Apresentar uma taxa de disponibilidade elevada;
- Apresentar uma taxa de falha reduzida;
- Apresentar uma elevada eficácia;
- Apresentar uma elevada eficiência;
- Apresentar um elevado desempenho mesmo nos cenários mais exigentes;
- Garantir uma fácil manutenção;
- Garantir tratamento de falhas.

5.2 Modelo de Domínio

O presente capítulo tem como objetivo apresentar e explicar os conceitos chave presentes no domínio do projeto. Na Figura 13 está representado um diagrama de domínio com os conceitos mais relevantes e as diversas ligações entre os conceitos.

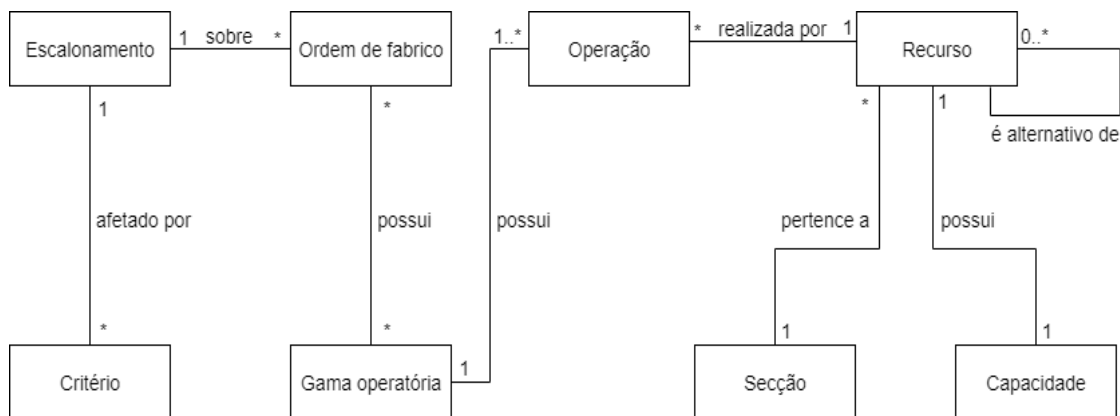


Figura 13 – Modelo de domínio

Com o objetivo de facilitar a compreensão da Figura 13, será feita uma breve descrição a cada um dos conceitos nela presentes:

- Escalonamento: entidade que é responsável por desencadear o processo de escalonamento e de agrupar os conceitos do negócio.
- Critério: o critério determina a forma como deve ser otimizado o escalonamento. Exemplo: diminuição dos tempos de preparação.
- Ordem de fabrico: representa um trabalho que é necessário realizar para produzir determinado artigo. Exemplo: produzir um livro.
- Gama operatória: representa um conjunto de operações, bem como, as suas dependências, que são necessárias para realizar uma ordem de fabrico. Exemplo: cortar e depois imprimir.
- Operação: é uma ação que é necessária realizar de forma a se obter o artigo final ou um artigo intermédio que é necessário para se obter o artigo final. Exemplo: imprimir.
- Secção: agrupa um ou mais recursos de forma a categorizar os mesmos. Exemplo: Impressoras.
- Recurso: responsável por realizar uma operação. Neste âmbito, é, normalmente, uma máquina. Exemplo: impressora A.
- Capacidade: representa a capacidade disponível do recurso, em termos temporais.

5.3 Design Arquitetural

Neste subcapítulo, será apresentado o design arquitetural do sistema a desenvolver, nomeadamente, as tecnologias, a vista lógica e a vista de processo. As vistas referidas anteriormente, fazem parte do modelo 4+1 (Kruchten and Rational Software, 1995) e permitem definir partes da arquitetura que irá ser seguida.

5.3.1 Vista Lógica

A aplicação a desenvolver constituirá uma aplicação independente de todas as outras aplicações que poderão fazer uso da mesma. Assim, na Figura 14, é possível observar esta aplicação, denominada “API Escalonador” que irá comunicar com uma outra qualquer aplicação, denominada de “Cliente”. A aplicação “Cliente” é apresentada apenas para efeitos demonstrativos, uma vez que a mesma não será desenvolvida no âmbito deste projeto.

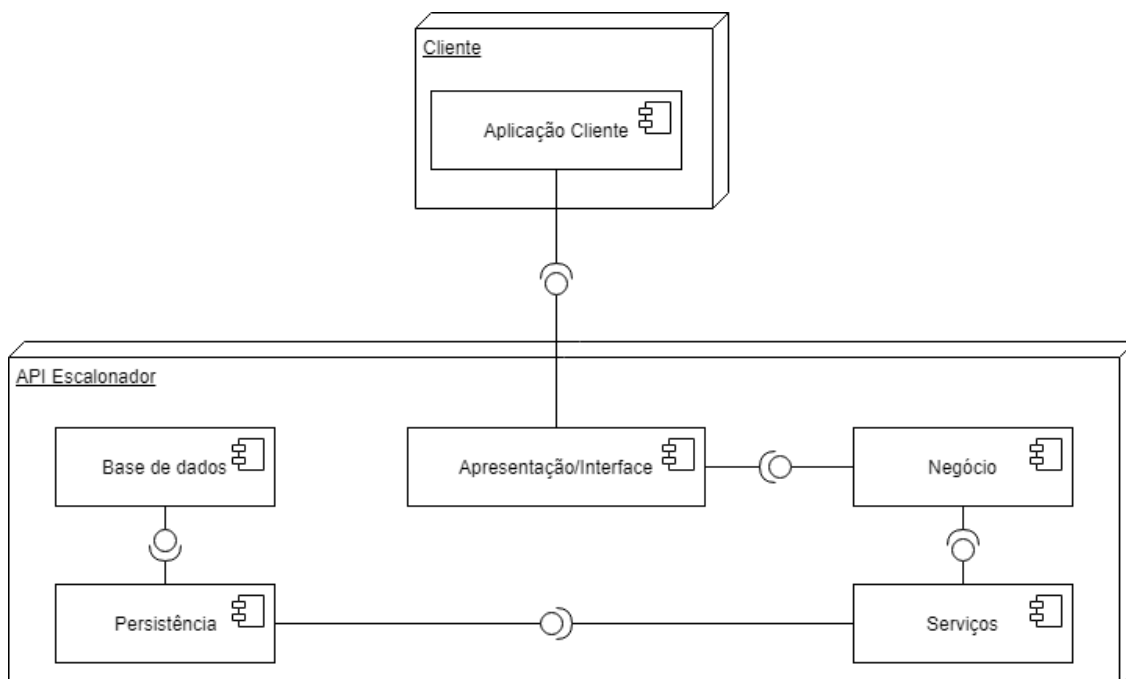


Figura 14 – Vista lógica

Ainda na Figura 14, são apresentados diversos componentes que irão constituir a aplicação a desenvolver, são eles:

- Apresentação/Interface: responsável por efetuar as comunicações com a aplicação “Cliente” e por assegurar que os dados são obtidos e enviados no formato correto.

- Negócio: responsável por definir toda a lógica e regras de negócio. É também responsável por efetuar o escalonamento das ordens de fabrico.
- Serviços: responsável por realizar a comunicação com a camada de persistência.
- Persistência: responsável por realizar a comunicação com a base de dados.
- Base de dados: responsável por armazenar os dados.

5.3.2 Vista de Processo

A vista de processo tem como objetivo apresentar as interações entre as diversas partes do sistema, quando é realizada uma determinada ação.

O processo principal do sistema será o de efetuar o escalonamento da produção. Como tal, no diagrama de sequência da Figura 15 está representado esse processo num formato de alto nível, com a intenção não só de demonstrar o processo, mas também de mostrar as interações entre os diversos componentes e aplicações.

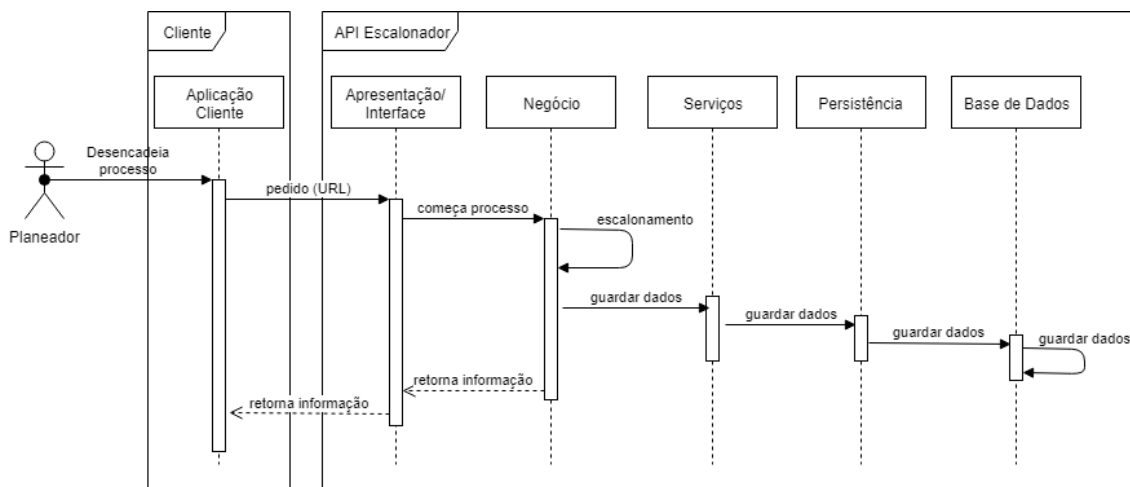


Figura 15 – Diagrama de sequência do processo de escalonar

A partir da Figura 15, verifica-se que o processo de escalonamento, desde a ação do utilizador até ao momento em que este obtém a resposta, consiste nas seguintes etapas:

1. O planeador inicia o processo de escalonar.
2. A aplicação “Cliente” recebe esse pedido e processa-o, efetuando um pedido à aplicação “API Escalonador”, com todos os dados necessários para a realização do escalonamento.

3. A aplicação "API Escalonador" recebe o pedido, trata-o e inicia o processo de escalonamento.
4. A camada de negócio efetua o escalonamento e solicita à camada de serviços que esse resultado seja guardado em base de dados.
5. A camada de serviços, por sua vez, solicita à camada de persistência a gravação do resultado do escalonamento.
6. A camada de persistência recebe o resultado do escalonamento e guarda o mesmo na base de dados.
7. É enviada uma resposta à camada de apresentação/interface com o resultado do escalonamento para que esta o processe para o formato correto.
8. A camada de apresentação/interface retorna o resultado processado para a aplicação cliente, para que esta possa apresentar os dados o planeador.

6 Desenvolvimento

Este capítulo, apresenta o processo de implementação do projeto, tendo em conta a análise e o design apresentados no capítulo anterior.

Será descrito, de forma sucinta, as tecnologias utilizadas e a estrutura do projeto. Será, por fim, apresentado o algoritmo desenvolvido que serve de base para o processo de escalonamento.

6.1 Tecnologias

De acordo com o estado da arte e com os requisitos/restrições do projeto provenientes da análise e do design, foram selecionadas as tecnologias apresentadas nos próximos subcapítulos.

Com esta combinação de tecnologias, é possível realizar uma solução *standalone*, e que utiliza algumas das mais recentes tecnologias disponíveis, com vantagens sobretudo em termos de desempenho.

6.1.1 Ambiente de Desenvolvimento

O desenvolvimento do projeto foi realizado sobre o ambiente de desenvolvimento *Microsoft Visual Studio 2017*, pois, para além de ser da Microsoft permite a criação de soluções que incorporam a *framework .NET Core 2.1*, identificada no estado da arte.

6.1.1.1 Estrutura do Projeto

A estrutura do projeto foi definida de acordo com uma arquitetura em camadas (Genero, Jiménez and Piattini, 2000), existindo uma separação de responsabilidades entre cada camada.

Na Figura 16, é apresentada a estrutura do projeto definida no ambiente de desenvolvimento explicado anteriormente.

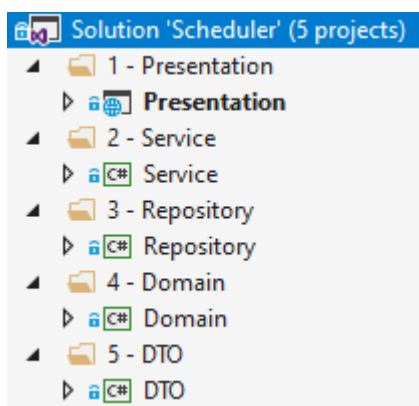


Figura 16 – Estrutura do Projeto

O projeto foi dividido em 5 camadas distintas:

- Presentation: responsável pela interação com as aplicações cliente, ou seja, responsável por receber pedidos (de escalonamento ou dos restantes casos de uso), delegar para a camada Service e, por fim, devolver a informação necessária;
- Service: responsável por tratar a informação recebida (em parte, transformar objetos da camada DTO em objetos da camada Domain) e, desencadear os processos necessários para a obtenção de resultados;
- Repository: responsável pela comunicação com a Base de Dados;
- Domain: contém a informação do negócio, sendo responsável por efetuar todas as transações, dependências e restrições entre as várias entidades do modelo de domínio;
- DTO: contém objetos de transferência de dados, ou seja, é responsável por possuir objetos de input e de output dos dados, de acordo com as necessidades da camada Presentation e da camada Repository.

Todas as camadas referidas, apesar de possuírem diferentes responsabilidades, funcionam em conjunto para permitir dar resposta aos casos de uso identificados anteriormente.

6.1.2 Framework .NET Core 2.1

Tal como referido no subcapítulo anterior, o projeto foi criado com base na tecnologia .NET Core 2.1. O uso desta tecnologia justifica-se nas premissas:

- Possibilidade de criação de aplicações *standalone*: com a opção de WEB API (Roth, Rick and Luttin, 2019), permite a criação de uma solução que pode receber pedidos de diversos clientes, tornando a solução implementável por outros sistemas, tal como pretendido;
- Permite dados do tipo JSON e XML (Anderson and Smith, 2020): desta forma, a solução torna-se ainda mais dinâmica, permitindo o uso de dados de input e output nos dois formatos descritos;
- Elevado desempenho: face às versões anteriores, nomeadamente à *.NET framework*, possui melhor desempenho (Toub, 2017), o que é fundamental para a solução em questão;

6.1.3 Base de Dados

Para a persistência de dados, foi utilizada a tecnologia Microsoft SQL Server 2017, fornecida pela Microsoft.

Esta tecnologia permite a criação de base de dados que permitem armazenar todos os dados necessários, por exemplo, histórico de escalonamentos, de forma estruturada através de tabelas.

Por fazer parte das tecnologias da Microsoft, a sua integração com as restantes tecnologias é simples, tornando-se numa escolha óbvia para este projeto.

6.2 Algoritmo de Escalonamento

Um dos objetivos deste projeto é a produção de um novo algoritmo de escalonamento. Este algoritmo deve respeitar e seguir determinadas regras e restrições previamente identificadas.

Com base no estado da arte efetuado e nas abordagens a adotar definidas, foi desenvolvido um algoritmo que procura ser robusto e com melhorias significativas face ao algoritmo atualmente utilizado pela Sistrade.

Este algoritmo baseia-se no conceito da abordagem principal definida (Palacios *et al.*, 2015), ainda que seja substancialmente diferente, sendo caracterizado por dois algoritmos distintos. Na Figura 17, observa-se a relação entre os dois algoritmos.

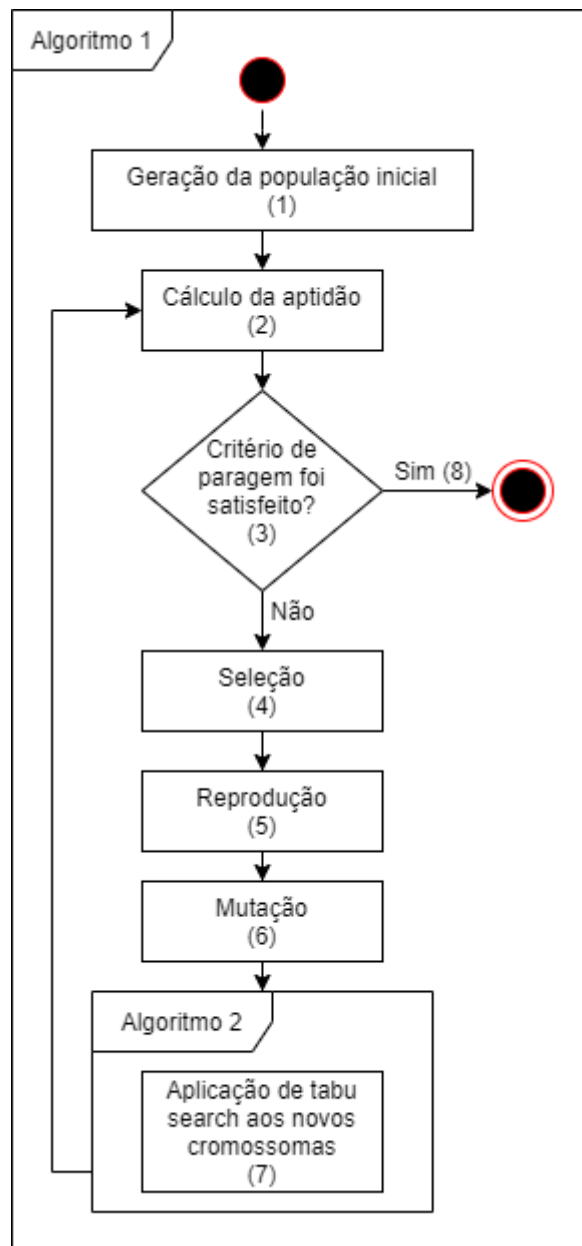


Figura 17 – Algoritmo desenvolvido

Antes da explicação do modo de funcionamento dos algoritmos, é fundamental a perceção da relação entre os conceitos de um algoritmo genético e os conceitos do problema em questão. Nas descrições dos algoritmos, serão utilizados os termos relacionados com algoritmos genéticos, e correspondem aos seguintes termos da temática em questão:

- População: corresponde a um conjunto de cromossomas, ou seja, a todos os conjuntos de Operações para o problema;
- Cromossoma: corresponde a uma solução, ou seja, a um conjunto de Operações;

- Gene: corresponde a uma Operação.

De igual forma, relativamente ao algoritmo do tipo *tabu search*, o conceito de vizinhança, refere-se um conjunto de operações.

O primeiro algoritmo (algoritmo 1) da figura anterior, é baseado na utilização de algoritmos genéticos. De acordo com a Figura 17, é constituído pelas seguintes fases:

- **(1) Início - Geração da população inicial**

Primeira fase, início do processo.

Nesta fase, é gerada a população inicial, recorrendo a um algoritmo de geração aleatória proposto por Fisher-Yates (Fisher and Yates, 1938). Este algoritmo foi selecionado pois apresenta uma elevada eficiência computacional.

- **(2) Cálculo da aptidão**

Fase responsável por calcular qual a aptidão da população. Este cálculo é efetuado tendo por base os critérios previamente definidos:

- Percentagem de ordens de fabrico atrasadas;
- Taxa de ocupação dos recursos;
- Percentagem do tempo de preparação;
- Duração total da produção (*makespan*).

Tendo em conta que podem ser selecionados 1 ou mais critérios, o cálculo é efetuado com a aplicação de um peso (percentagem) a cada critério, baseando-se no modelo de soma aplicada de Fishburn (Fishburn, 1967).

Após este cálculo, são identificados alguns cromossomas elites. Ou seja, os melhores X % (número configurável) são selecionados como sendo elites. Este conceito aplica-se com o objetivo de garantir que nas populações seguintes, pelo menos X % dos cromossomas se mantêm, evitando a possível perda de melhores soluções (Villalobos-Arias, Coello and Hernández-Lerma, 2006).

- **(3) Critério de paragem**

Nesta fase, é feita a verificação do(s) critério(s) de paragem. Caso este(s) critério(s) tenha(m) sido satisfeito(s), então a execução do algoritmo termina, passando para a fase (8).

Foram definidos dois critérios de paragem:

- Número máximo de gerações (Safe *et al.*, 2004): o algoritmo termina quando um determinado número de gerações foi atingido, entenda-se por geração a criação de uma nova população, portanto, uma iteração do algoritmo;
- Número máximo de melhor solução repetida (Bhandari, Murthy and Pal, 2012): o algoritmo termina quando o número máximo aceitável de gerações com a mesma melhor solução é atingido, ou seja, após X gerações com a mesma solução, a execução termina e aceita-se que já não irá ser encontrada uma solução melhor do que a atual.

Caso o(s) critério(s) de paragem não seja(m) satisfeito(s), então o algoritmo continua a sua execução, passando para a fase (4).

- **(4) Seleção**

Na fase da Seleção, são escolhidos dois cromossomas pais, com o objetivo de, seguidamente, se reproduzirem na fase (5).

O mecanismo de seleção utilizado é o método da roleta (Holland, 1975). Neste método, cada gene do cromossoma é representado na roleta proporcionalmente à sua aptidão, calculado em (2), ou seja, os genes com uma maior aptidão, têm uma maior probabilidade de serem selecionados. Seguidamente, a roleta é girada duas vezes, ocorrendo a seleção dos dois pais sorteados pela roleta.

- **(5) Reprodução**

Nesta fase, é efetuada a reprodução/recombinação genética. Por outras palavras, a partir dos pais selecionados, serão gerados dois filhos, ou seja, dois novos cromossomas.

A reprodução é efetuada tendo por base um operador de reprodução ordenada (Davis, 1985). Este operador consiste em:

1. Selecionar dois pontos (posições) dos genes que serão mantidos nos pais. Por exemplo, os genes das posições 3 a 5, identificados entre o separador “|”:

Pai 1 = (1 2 | 3 4 5 | 6 7 8 9)

Pai 2 = (8 5 | 7 1 2 | 4 9 3 6)

2. Criar dois filhos de acordo com a seleção anterior, mantendo apenas os genes selecionados na etapa anterior:

Filho 1 = (- - | 3 4 5 | - - - -)

Filho 2 = (- - | 7 1 2 | - - - -)

3. Os caracteres “-“ serão substituídos por genes na ordem em que se encontram no pai oposto, ou seja, para o Filho 1, os genes serão inseridos de acordo com a ordem no Pai 2. Esta ordem inicia-se a partir da segunda posição selecionada em 1., ou seja, 4 9 3 6 8 5 7 1 2;
4. À sequência anterior, é necessário remover os genes que já fazem parte do Filho 1, ou seja, remover os genes 3 4 5, resultando em 9 6 8 7 1 2.
5. A sequência determinada em 4. é utilizada para a inserção dos genes no Filho 1, a partir da segunda posição determinada em 1., ou seja:

Filho 1 = (- - | 3 4 5 | 9 6 8 7)

6. Seguidamente, são preenchidas as restantes posições do Filho 1 com os genes restantes (1 e 2), ou seja:

Filho 1 = (1 2 | 3 4 5 | 9 6 8 7)

7. O mesmo processo é aplicado para o Filho 2, a partir da etapa 3., resultando no fim deste processo dois novos cromossomas.

Para esta etapa, é possível definir uma probabilidade de reprodução. Ou seja, quando esta não está definida para 100%, algumas das iterações nesta fase, podem não originar novos cromossomas.

- **(6) Mutação**

Após a reprodução, poderão ocorrer mutações aos cromossomas gerados. As mutações asseguram a diversidade da solução e ocorrem de acordo com uma probabilidade, à semelhança da reprodução (Daglayan and Karakaya, 2016).

A mutação utilizada consiste na troca de posição de dois genes (Bermudez *et al.*, 2010), por exemplo, trocar o gene 3 com o gene 7, tal como demonstrado de seguida:

Cromossoma original = 1 2 3 4 5 6 7 8 9

Cromossoma após mutação = 1 2 7 4 5 6 3 8 9

- **(7) Aplicação de Tabu Search**

Tal como proposto por Palacios (Palacios *et al.*, 2015), e de forma a obter melhores soluções, que não fiquem presas em ótimos locais, foi introduzido nesta solução um algoritmo baseado Tabu Search.

Este algoritmo verifica possíveis melhores soluções e evita a avaliação de soluções repetidas.

O presente algoritmo é aplicado a seguir à etapa (6).

Com esta abordagem, para cada cromossoma, o algoritmo é executado um determinado número de vezes (configurável), e com base na vizinhança do cromossoma, retorna a melhor solução encontrada até ao momento.

A vizinhança é gerada a partir do cromossoma original, ocorrendo um processo semelhante ao da etapa (6), ou seja, existe a troca de posição entre dois genes (Schneider, 2011).

A avaliação da solução é feita de acordo com os critérios definidos, tal como explicado na etapa (2).

No fim da execução deste algoritmo, volta a ser executada a etapa (2), repetindo-se o processo descrito até à etapa (8).

- **(8) Fim – Retorno da melhor solução**

Nesta última etapa, o algoritmo termina a sua execução e devolve a melhor solução encontrada para os critérios selecionados.

6.3 Dados de Input

Tal como descrito nos requisitos do projeto, os dados de input podem ser enviados em dois formatos distintos, JSON e XML.

Nas demonstrações ao longo deste subcapítulo e por forma a simplificar a apresentação dos dados, apenas serão apresentados dados no formato JSON. Para além disto, os dados de input variam de acordo com o caso de uso em questão, mas para efeitos demonstrativos, serão apresentados os dados de input para o caso de uso 1 (Efetuar o escalonamento da produção).

Nos subcapítulos seguintes, serão apresentados os formatos dos dados de input e as funcionalidades permitidas pelo escalonador.

6.3.1 Ordem de Fabrico

O elemento principal por entre os dados de input, é a Ordem de Fabrico. Estas possuem gamas operatórias, que, por sua vez, estas são compostas por operações, tal como se pode observar na Figura 18. Todas as entidades referidas, são identificadas com base num código, por exemplo, "JO_1" para uma Ordem de Fabrico.

```
"jobOrders": [{
  "code": "JO_1",
  "productionRoutes": [{
    "code": "PO_1",
    "operations": [{
      "code": "OP_1",
      "resourceCode": "R_1",
      "startDate": "2019-09-27T00:00:00.000Z",
      "endDate": "2019-09-27T00:00:00.000Z",
      "setupTime": 0,
      "productionTime": 700,
      "restTime": 0,
      "percentagePreviousOperation": 0,
      "sequence": 1,
      "operationDependencyToProductionRoutes": [{
        "productionRouteCode": "PO_2",
        "operationCode": "OP_2"
      }
    ],
    "alternativeResources": [{
      "resourceCode": "R_2",
      "setupTime": 30,
      "productionTime": 350
    }
  ],
  "setupFeatures": [{
    "feature": "F1",
    "value": "B"
  }
]
},
  "forecastedDate": "2019-09-30T00:00:00.000Z"
},
]
```

Figura 18 – Dados de input – Ordem de Fabrico

As operações são constituídas pelos atributos base:

- *resourceCode*: código do recurso onde a operação deverá ser realizada;

- *startDate*: data de início do escalonamento anterior;
- *endDate*: data de fim do escalonamento anterior;
- *setupTime*: tempo de preparação na máquina em questão;
- *productionTime*: tempo de produção na máquina em questão;
- *restTime*: tempo de repouso;
- *percentagePreviousOperation*: percentagem de conclusão da operação anterior necessária para a operação em questão poder ser planeada;
- *sequence*: ordem da operação dentro da gama operatória, ou seja, sequência no fluxo produtivo que permite a criação de restrições entre operações.

Para além das características referidas anteriormente, as operações podem possuir outras características e funcionalidades não obrigatórias. Estas funcionalidades estão de acordo com os requisitos identificados na análise da solução.

Restrições entre gamas operatórias podem ser definidas através do atributo *operationDependencyToProductionRoutes*, que é constituído por:

- *productionRouteCode*: código da gama operatória;
- *operationCode*: código da operação.

Quanto ao conceito de balanceamento de carga, pode ser definido pelo atributo *alternativeResources*, no qual é possível definir recurso alternativos para a operação, que possui os seguintes atributos:

- *resourceCode*: código do recurso alternativo;
- *setupTime*: tempo de preparação no recurso alternativo;
- *productionTime*: tempo de produção no recurso alternativo.

Por fim, de forma a suportar o conceito de matriz de tempos de preparação, é possível definir o atributo *setupFeatures*, que é constituído por:

- *feature*: código da característica;
- *value*: valor da característica.

Para o conceito anterior, um exemplo simples e que permite compreender melhor a sua utilização é: característica denominada “tamanho do papel” com o valor “A4”.

6.3.2 Recurso

Uma lista de recursos tem de ser enviada, pois, permite aplicar o conceito de capacidade finita dos recursos, contemplado na análise de requisitos. Este conceito define qual o horário no qual o recurso pode trabalhar, quer seja por restrições técnicas, quer seja por restrições de mão de obra.

Na Figura 19, apresentam-se quais os atributos necessários para a lista de recursos.

```
"resources": [{
  "code": "R_1",
  "calendar": {
    "days": [{
      "date": "2019-09-30T00:00:00.000Z",
      "shifts": [{
        "start": "2019-09-30T00:00:00.000Z",
        "end": "2019-09-30T23:59:59.999Z"
      }]
    }]
  }
}],
```

Figura 19 – Dados de input – Recurso

Para além do código do recurso, deve ser definido qual a disponibilidade/calendário do recurso, através do atributo *calendar*, que, por sua vez, é constituído por uma lista de dias (*days*), com os seguintes atributos:

- *date*: data do dia;
- *shifts*: turnos/períodos trabalháveis dentro do dia, definidos através de uma data de início (*start*) e uma data de fim (*end*).

6.3.3 Matriz de Tempos de Preparação

Tal como apresentado anteriormente, o conceito de matriz de tempos de preparação utiliza a entidade de características como base do seu funcionamento. As características podem ser, por exemplo: cor, tamanho do papel, espessura do papel, etc.

Na Figura 20, é apresentada a estrutura deste conceito aplicada na solução.

```

"setupMatrix": [{
  "resourceCode": "R_1",
  "features": [{
    "feature": "F1",
    "detail": [{
      "valueFrom": "A",
      "valueTo": "B",
      "setupTime": 10,
      "setupMatrixType": 2
    }, {
      "valueFrom": "B",
      "valueTo": "A",
      "setupTime": 20,
      "setupMatrixType": 2
    }
  ]
}
]

```

Figura 20 – Dados de input – Matriz de tempos de preparação

As características são definidas por recurso, pois, podem apresentar diferentes tempos de preparação de acordo com o recurso. Estas, são definidas por um código e por uma lista de transições, representada pelo atributo *detail*. Para cada transição, são inseridas as informações:

- *valueFrom*: valor de origem da transição;
- *valueTo*: valor de destino da transição;
- *setupTime*: tempo de preparação;
- *setupMatrixType*: tipo de transição, podendo ser definido como:
 - 0 – subtração: o tempo definido em *setupTime* será subtraído ao tempo de preparação por defeito da operação;
 - 1 – substituição: o tempo de preparação por defeito da operação será substituído pelo tempo definido em *setupTime*;
 - 2 – adição: o tempo definido em *setupTime* será adicionado ao tempo de preparação por defeito da operação.

6.3.4 Configurações

Devido às características do algoritmo desenvolvido, existem inúmeras configurações que podem ser efetuadas. Estas configurações afetam o tempo de execução do algoritmo e a qualidade das soluções.

Na Figura 21, observam-se os atributos que constituem as configurações passíveis de serem efetuadas.

```
"scheduleConfiguration": {  
  "deepOptimization": true,  
  "maxGenerations": 20,  
  "maxSameFitnessValue": 10,  
  "populationSize": 20,  
  "eliteFactor": 5,  
  "crossoverProbability": 60,  
  "mutationProbability": 10,  
  "scheduleCriteria": {  
    "opT_MAKESPAN": 0,  
    "opT_SETUP": 0,  
    "opT_LOAD": 0,  
    "opT_DELAYED": 100  
  }  
}
```

Figura 21 – Dados de input - Configurações

O escalonador desenvolvido permite controlar a partir do atributo *deepOptimization* se irá ou não ser aplicado o algoritmo de otimização criado. Ou seja, caso não esteja ativo, comporta-se como um sequenciador, respeitando as regras e restrições sem procurar qualquer otimização. Com este atributo desativado, o escalonador desenvolvido comporta-se de forma muito semelhante ao escalonador da Sistrade.

Quando o atributo *deepOptimization* está ativo, então é possível configurar:

- *maxGenerations*: número máximo de gerações, utilizado, como explicado previamente, como critério de paragem do algoritmo;
- *maxSameFitnessValue*: número máximo de melhor solução repetida, utilizado, como explicado previamente, como critério de paragem do algoritmo;
- *populationSize*: tamanho de uma população, portanto, o número de cromossomas de uma população;
- *eliteFactor*: fator de elitismo, utilizado para calcular o número de cromossomas elites;

- *crossoverProbability*: probabilidade de ocorrência de reprodução;
- *mutationProbability*: probabilidade de ocorrência de mutações;
- *scheduleCriteria*: definição dos critérios que otimização que serão utilizados pelo algoritmo, podendo ser selecionados múltiplos critérios.

6.4 Dados de Output

Relativamente aos dados de output, é retornada uma estrutura semelhante aos dados de input. Esta estrutura encontra-se representada na Figura 22.

Essencialmente, para cada operação, é apresentado o código do recurso, a data de início e de fim, o tempo de preparação e o tempo produção.

```

"jobOrders": [{
  "code": "JO_1",
  "productionRoutes": [{
    "code": "PO_1",
    "operations": [{
      "code": "OP_1",
      "resourceCode": "R_1",
      "startDate": "2019-10-09T08:00:00Z",
      "endDate": "2019-10-09T09:30:00Z",
      "setupTime": 30,
      "productionTime": 60
    }
  ]
}
]

```

Figura 22 – Dados de output

7 Avaliação

Neste capítulo, serão descritas, com base na literatura recolhida no estado da arte, as diferentes abordagens que permitem avaliar a solução obtida, bem como, os diferentes dados de input.

O método de experimentação dominante identificado, é a simulação. Na simulação, para vários cenários diferentes, determinam-se e avaliam-se os diferentes resultados obtidos. Assim, as abordagens escolhidas serão baseadas na simulação.

7.1 Indicadores e Fontes de Informação

Neste capítulo, são apresentados os indicadores e fontes de informação que serão utilizados para que seja possível fazer uma avaliação à solução. Os indicadores definem quais são os parâmetros/hipóteses que devem ser analisados. Por sua vez, as fontes de informação definem quem ou o quê irá fornecer a informação necessária para o processo de avaliação.

7.1.1 Indicadores

De forma a ser possível avaliar a solução, é fundamental a realização da identificação de indicadores. Os indicadores relevantes para a avaliação da solução são:

- Eficiência
 - Tempo: tempo em necessário para obter uma solução;
 - Memória: memória utilizada aquando da execução da solução.
- Eficácia

- *Makespan*: tempo máximo total planeado para realizar todas as ordens de fabrico;
- Percentagem de ordens de fabrico atrasadas: ordens de fabrico planeadas para uma data posterior à data de entrega face às ordens de fabrico existentes;
- Taxa de ocupação dos recursos: tempo total em que os recursos estão ocupados com ordens de fabrico face ao tempo total em que os recursos estão disponíveis;
- Percentagem do tempo de preparação: tempo de preparação face ao tempo total.

7.1.2 Fontes de Informação

Para cada indicador definido anteriormente poder ser calculado, é necessária a existência de fontes de informação que forneçam os dados necessários. Essas fontes de informação são:

- Input com cenários fictícios: cenários criados exclusivamente para a verificação dos indicadores;
- Input com cenários reais: cenários provenientes de situações reais.

Para cada fonte de informação, existe um conjunto de dados/parâmetros que deve ser definido. Estes dados/parâmetros interferem diretamente com os resultados dos indicadores e os mais relevantes são:

- Nº de ordens de fabrico;
- Nº de operações por ordem de fabrico;
- Nº de recursos alternativos;
- Tipo de escalonamento.

7.2 Hipóteses a Avaliar

Cada indicador definido no capítulo anterior, resulta numa hipótese a avaliar. Estas hipóteses, têm como objetivo validar o desempenho e qualidade da solução segundo dois fatores: a eficiência a eficácia.

Cada hipótese será especificada sob a premissa que irá existir uma comparação direta entre a solução proposta a solução do Sistrade ERP. Esta abordagem será realizada seguindo o modelo de teste de hipóteses denominado *Paired T-Test* (Mcgreedy, 2006). Neste modelo, são definidos tipos de hipóteses:

- H0: hipótese nula, considerada como a hipótese que se pretende rejeitar;
- H1: hipótese alternativa, considerada como a hipótese que determinar como válida.

7.2.1 Eficiência

Neste capítulo serão apresentadas as hipóteses para cada indicador pertencente à avaliação da eficiência.

7.2.1.1 Tempo

H0: o tempo de execução da solução do Sistrade ERP é igual ou inferior ao tempo de execução da solução proposta;

H1: o tempo de execução da solução proposta é inferior ao tempo de execução da solução do Sistrade ERP.

7.2.1.2 Memória

H0: a memória utilizada pela solução do Sistrade ERP é igual ou inferior à memória utilizada pela solução proposta;

H1: a memória utilizada pela solução proposta é inferior à memória utilizada pela solução do Sistrade ERP.

7.2.2 Eficácia

Neste capítulo serão apresentadas as hipóteses para cada indicador pertencente à avaliação da eficácia.

7.2.2.1 Makespan

H0: o *makespan* obtido pela solução Sistrade ERP é igual ou inferior ao *makespan* obtido pela solução proposta;

H1: o *makespan* obtido pela solução proposta é inferior ao *makespan* obtido pela solução do Sistrade ERP.

7.2.2.2 Percentagem de ordens de fabrico atrasadas

H0: a percentagem de ordens de fabrico atrasadas obtida pela solução do Sistrade ERP é igual ou inferior à percentagem de ordens de fabrico atrasadas obtida pela solução proposta;

H1: a percentagem de ordens de fabrico atrasadas obtida pela solução proposta é inferior à percentagem de ordens de fabrico atrasadas obtida pela solução do Sistrade ERP.

7.2.2.3 Taxa de ocupação dos recursos

H0: a taxa de ocupação dos recursos obtida pela solução do Sistrade ERP é igual ou inferior à taxa de ocupação dos recursos obtida pela solução proposta;

H1: a taxa de ocupação dos recursos obtida pela solução proposta é inferior à taxa de ocupação dos recursos obtida pela solução do Sistrade ERP.

7.2.2.4 Percentagem do tempo de preparação

H0: a percentagem do tempo de preparação obtida pela solução do Sistrade ERP é igual ou inferior à percentagem do tempo de preparação obtida pela solução proposta;

H1: a percentagem do tempo de preparação obtida pela solução proposta é inferior à percentagem do tempo de preparação obtida pela solução do Sistrade ERP.

7.3 Metodologia de Avaliação e Resultados

Neste capítulo, serão apresentados os diferentes testes e simulações efetuados, bem como, os respetivos dados de input necessários, para avaliar as diferentes hipóteses definidas anteriormente.

7.3.1 Dados de Input

Para a realização das diferentes experimentações que avaliam as hipóteses, é necessário definir quais os dados de input que irão suportar essas avaliações. Estes dados são baseados nos parâmetros identificados a partir das fontes de informação definidas anteriormente.

Serão definidos vários cenários para os diferentes dados de input. O nome de cada cenário será constituído por: C[X]_[Y] em que, [X] corresponde a um número (sequencial) para o tipo

de dados de input (cada uma das secções seguintes) e [Y] corresponde a um número sequencial para os diferentes inputs do tipo de input.

7.3.1.1 Nº de ordens de fabrico, operações e recursos alternativos – C1_[Y]

Na Tabela 5, estão representados diversos cenários que variam entre si quanto ao número de ordens de fabrico, ao número de operações e ao número de recursos alternativos por operação.

Tabela 5 – Dados de input C1

Nome do Cenário	Nº de ordens de fabrico	Nº de operações	Nº de recursos alternativos
C1_A	10	5	2
C1_B	10	10	2
C1_C	50	5	3
C1_D	50	10	3

7.3.1.2 Tipo de escalonamento – C2_[Y]

Na Tabela 6, estão representados cenários para cada um dos critérios (indicadores de eficácia) definidos.

Tabela 6 - Dados de input C2

Nome do Cenário	Diminuir <i>makespan</i>	Diminuir % ordens de fabrico atrasadas	Diminuir % tempo preparação	Aumentar taxa de ocupação dos recursos
C2_A	X			
C2_B		X		
C2_C			X	
C2_D				X

7.3.1.3 Parâmetros do escalonador Sistrade

Como critério de sequenciação, foi definida a data de entrega, uma vez que, o escalonador não permite definir critérios para otimizar os restantes cenários.

Não foi definida nenhuma matriz de tempos de preparação nem definidos recursos alternativos, uma vez que, o escalonador não o permite.

7.3.1.4 Parâmetros Solução Proposta

Foi definido um critério de otimização de acordo com cada cenário criado (C2_A, C2_B, C2_C, C2_D).

Os parâmetros do algoritmo foram definidos com os seguintes valores:

- *maxGenerations*: 20;
- *maxSameFitnessValue*: 10;
- *populationSize*: 20;
- *eliteFactor*: 5;
- *crossoverProbability*: 60;
- *mutationProbability*: 5;
- *scheduleCriteria*: C2_A ou C2_B ou C2_C ou C2_D.

Relativamente ao tempo de preparação, foi ainda definida uma matriz de tempos de preparação fictícia, uma vez que, o Escalonador Sistrade não permite esta funcionalidade.

Quanto aos recursos alternativos, foram definidos para possuírem o mesmo tempo de preparação e de produção. Ou seja, para cada operação, o tempo não varia de acordo com o recurso. Contudo, permite que seja efetuado um balanceamento de carga.

7.3.2 Eficiência

Uma das principais restrições é que o escalonador consiga determinar uma solução válida em tempo útil. Associado a este conceito, está o conceito de memória, que também é fundamental para determinar o bom funcionamento da solução.

Relativamente à eficiência, foram definidas 4 colunas, representadas na Tabela 7:

- TSP: tempo de execução da Solução Proposta;
- TES: tempo de execução do Escalonador Sistrade;
- TSP – TES: desvio entre o tempo de execução da Solução Proposta e o tempo de execução do Escalonador Sistrade;
- MSP: memória utilizada pela Solução Proposta;

- MES: memória utilizada pelo Escalonador Sistrade;
- MSP – MES: desvio entre a memória utilizada pela Solução Proposta e a memória utilizada pelo Escalonador Sistrade.

As colunas relativas a campos temporais (TSP, TES e TSP – TES) encontram-se em segundos. Por outro lado, as colunas relativas a campos de memória (MSP, MES e MSP – MES) são apresentadas em *megabytes*.

O tempo de execução corresponde à diferença entre o fim do processo de escalonamento e o recebimento do pedido. Relativamente à memória, foi calculada tendo em conta o mesmo princípio. Para determinação da memória que está a ser utilizada, foi utilizada a função *GC.GetTotalMemory* disponibilizada pela Microsoft (Microsoft, 2020).

Tabela 7 – Avaliação da eficiência

C1	C2	SPT	EST	SPT - EST	SPM	ESM	SPM - ESM
C1_A	C2_A	10	2,5	7,5	40	31	9
C1_A	C2_B	11	2,5	8,5	42	31	11
C1_A	C2_C	11	2,5	8,5	49	31	18
C1_A	C2_D	10	2,5	7,5	41	31	10
C1_B	C2_A	20	3,5	16,5	59	30	29
C1_B	C2_B	21	3,5	17,5	53	30	23
C1_B	C2_C	21	3,5	17,5	64	30	34
C1_B	C2_D	20	3,5	16,5	39	30	9
C1_C	C2_A	60	3	57	48	44	4
C1_C	C2_B	63	3	60	58	44	14
C1_C	C2_C	63	3	60	85	44	41
C1_C	C2_D	63	3	60	50	44	6
C1_D	C2_A	191	3,5	187,5	60	46	14
C1_D	C2_B	192	3,5	188,5	62	46	16
C1_D	C2_C	189	3,5	185,5	57	46	11
C1_D	C2_D	191	3,5	187,5	126	46	80

7.3.3 Eficácia

De forma a determinar se a solução encontrada satisfaz os objetivos/critérios das organizações, foram definidos os seguintes indicadores, apresentados na Tabela 8, que comparam a solução desenvolvida com o escalonador Sistrade:

- SPMK – *Makespan* da Solução Proposta;
- ESMK – *Makespan* do Escalonador Sistrade;
- SPPO - Percentagem de ordens de fabrico atrasadas na Solução Proposta;
- ESPO - Percentagem de ordens de fabrico atrasadas no Escalonador Sistrade;
- SPPR - Taxa de ocupação dos recursos na Solução Proposta;
- ESPR - Taxa de ocupação dos recursos no Escalonador Sistrade;
- SPPP - Percentagem do tempo de preparação na Solução Proposta;
- ESPP – Percentagem do tempo de preparação no Escalonador Sistrade;

Tabela 8 – Avaliação da eficácia

C1	C2	SPMK	ESMK	SPPO	ESPO	SPPR	ESPR	SPPP	ESPP
C1_A	C2_A	86	88	50	60	85	93	18	36
C1_A	C2_B	87	88	30	60	93	93	18	36
C1_A	C2_C	87	88	40	60	86	93	16	36
C1_A	C2_D	87	88	50	60	98	93	20	36
C1_B	C2_A	92	101	50	80	41	49	25	54
C1_B	C2_B	94	101	40	80	39	49	25	54
C1_B	C2_C	92	101	50	80	46	49	24	54
C1_B	C2_D	94	101	50	80	41	49	21	54
C1_C	C2_A	376	403	68	100	89	95	15	36
C1_C	C2_B	394	403	56	100	88	95	15	36
C1_C	C2_C	377	403	58	100	95	95	15	36
C1_C	C2_D	394	403	66	100	87	95	14	36
C1_D	C2_A	381	542	78	100	37	40	19	54
C1_D	C2_B	399	542	64	100	36	40	19	54

C1	C2	SPMK	ESMK	SPPO	ESPO	SPPR	ESPR	SPPP	ESPP
C1_D	C2_C	397	542	74	100	40	40	19	54
C1_D	C2_D	399	542	80	100	37	40	18	54

Os indicadores SPMK e ESMK, foram calculados de acordo com a seguinte fórmula: diferença em horas entre a data de fim da última operação e a data de início da primeira operação.

Os indicadores SPPO e ESPO, foram calculados de acordo com a seguinte fórmula: percentual entre número total de ordens de fabrico atrasadas (data de fim posterior à data de entrega) e o número total de ordens de fabrico.

Os indicadores SPPR e ESPR, foram calculados de acordo com a seguinte fórmula: percentual entre o total da duração das operações (preparação + produção) e o tempo total em que a máquina está disponível para trabalhar (considerado no máximo até à data de fim da última operação do recurso).

Os indicadores SPPP e ESPP, foram calculados de acordo com a seguinte fórmula: percentual entre o total de tempo de preparação e o total de tempo da operação (preparação + produção).

De forma a se conseguir avaliar como os parâmetros do escalonador afetam o resultado, foram criadas simulações com os seguintes parâmetros do algoritmo da solução proposta:

- P1: *maxGenerations*, *maxSameFitnessValue*, *populationSize* definidos com valor 5
- P2: *maxGenerations*, *maxSameFitnessValue*, *populationSize* definidos com valor 10
- P3: *maxGenerations*, *maxSameFitnessValue*, *populationSize* definidos com valor 20

De acordo com os parâmetros definidos, P1, P2 e P3, foram geradas simulações para o cenário C1_B. Estas simulações são apresentadas na Tabela 9.

Tabela 9 – Avaliação dos parâmetros da solução proposta

C1	C2	P	SPMK	SPPO	SPPR	SPPP
C1_B	C2_A	P1	399	90	35	19
C1_B	C2_B	P1	400	76	36	19
C1_B	C2_C	P1	397	80	37	19
C1_B	C2_D	P1	399	80	36	19

C1	C2	P	SPMK	SPPO	SPPR	SPPP
C1_B	C2_A	P2	396	88	38	19
C1_B	C2_B	P2	399	72	38	19
C1_B	C2_C	P2	401	88	39	19
C1_B	C2_D	P2	399	84	35	19
C1_B	C2_A	P3	381	78	37	19
C1_B	C2_B	P3	399	64	36	19
C1_B	C2_C	P3	397	74	40	18
C1_B	C2_D	P3	399	80	37	19

7.4 Conclusões

Após análise dos resultados obtidos, verifica-se, praticamente em todos os cenários, uma melhoria significativa da solução proposta face à solução da Sistrade, relativamente aos indicadores da eficácia. É de salientar que, mediante alteração dos parâmetros do algoritmo, pode-se obter resultados ainda melhores, ou seja, a qualidade da solução está dependente dos parâmetros do algoritmo.

Por outro lado, relativamente aos indicadores da eficiência, a solução da Sistrade obteve os melhores resultados. Verificou-se uma diferença significativa no tempo de execução do algoritmo da solução proposta para um elevado número de operações. Este resultado já era esperado, pois é uma solução mais simples e que não efetua uma otimização significativa dos resultados. Ainda assim, a solução proposta obteve resultados satisfatórios, podendo ainda ser ajustada para obter resultados de forma mais eficiente, através dos parâmetros definidos para o algoritmo. Relativamente à memória utilizada, não foram notadas diferenças significativas, que possam colocar em causa a sua execução numa máquina relativamente atual.

Pode-se ainda concluir que existe uma relação direta entre o número de operações e os indicadores de eficiência. De igual forma, existe também uma possível associação entre os parâmetros do algoritmo e os indicadores da eficácia.

Em suma, a solução proposta conseguiu otimizar os resultados de acordo com os critérios definidos, incorporando novas funcionalidades e sendo, portanto, uma solução muito mais robusta.

8 Conclusões e Trabalho Futuro

Este capítulo pretende descrever quais as conclusões determinadas, findo o processo de avaliação. São ainda apresentados quais os objetivos alcançados e o que poderá ser realizado como trabalho futuro.

8.1 Principais Conclusões

Tendo em conta os objetivos definidos, foi realizada uma extensa investigação sobre a temática do escalonamento da produção. Mais precisamente, no capítulo do estado da arte, foram apresentadas e analisadas diferentes abordagens sobre escalonamento da produção, tendo sido determinadas as principais abordagens que poderiam ser seguidas. Foram ainda investigadas tecnologias que permitem a criação de uma solução capaz de responder aos problemas apresentados e capaz de cumprir os objetivos definidos.

A solução elaborada nesta dissertação, contempla os requisitos definidos, nomeadamente, o conceito de restrições entre operações, recursos de capacidade finita, matriz de tempos de preparação, critérios de otimização, recursos alternativos e balanceamento de carga. A adição de novas funcionalidades, traduz-se numa solução mais robusta e abrangente, e consequentemente, numa solução mais aproximada do mundo real.

Garantiu-se que a solução elaborada é independente da plataforma, podendo ser utilizada e integrada facilmente por outros sistemas, utilizando JSON ou XML para os dados de *input* e de *output*.

Por fim, a eficiência e a eficácia da solução realizada foram medidas, utilizando diversos parâmetros e para diversos cenários, tendo sido obtidos melhores resultados face à solução da Sistrade em relação aos indicadores da eficácia. Os indicadores de eficiência, tal como

esperado, obtiveram piores resultados pois é uma solução mais complexa, mas ainda assim, os resultados obtidos são aceitáveis e trabalháveis.

Foi ainda avaliado o desempenho da solução proposta face à variação dos valores dos parâmetros do escalonador, tendo sido concluído que é possível obter melhores soluções mediante a modificação dos valores dos parâmetros.

8.2 Objetivos Alcançados

Inicialmente, foram traçados objetivos para a resolução do problema apresentado nesta dissertação. Estes objetivos foram cumpridos com sucesso, tendo sido elaborada uma identificação e análise das abordagens/algoritmos relevantes que permitem colmatar os problemas apresentados pelo algoritmo da Sistrade.

Foi concebido um algoritmo que incorpora não só os conceitos analisados nas abordagens estudadas, mas também os requisitos funcionais e não funcionais delineados.

Conseguiu-se efetuar a integração da solução desenvolvida no protótipo, tendo sido realizada uma avaliação à solução de acordo com diferentes cenários.

Relativamente aos resultados esperados definidos inicialmente, obtiveram-se resultados muito positivos, que se traduzem em ganhos significativos, nomeadamente, no processo de escalonamento conseguiram-se obter soluções que apresentam uma diminuição da percentagem de ordens de fabrico atrasadas, um aumento da taxa de ocupação dos recursos, uma diminuição do tempo total de preparação e uma diminuição da duração total da produção face à solução da Sistrade. Por exemplo, para o cenário testado mais complexo, possuindo 500 operações, a solução proposta reduziu o *makespan* de 542 horas para 381 horas, traduzindo-se num ganho de 161 horas.

8.3 Limitações e Trabalho Futuro

Durante a fase de elaboração do estado da arte, encontraram-se algumas limitações. Nomeadamente, existe um vasto conjunto de abordagens para esta temática e a seleção das abordagens foi um processo complexo, que exigiu a leitura e investigação de diversas soluções de forma a ser possível determinar quais as prováveis melhores soluções. O termo “prováveis”, foi propositadamente escolhido porque as soluções analisadas são apresentadas de forma meramente teórica, ou seja, não existe a possibilidade de ser realizada uma experimentação ao algoritmo definido em cada solução, ainda que, porventura, possa existir uma ou outra solução que permitam replicar em parte o algoritmo, mas não na totalidade. Em âmbito de melhoria, nomeadamente em relação aos algoritmos genéticos e do tipo *tabu search*, deve ser realizada uma investigação detalhada de quais os melhores mecanismos (por exemplo, qual o

melhor mecanismo de seleção ou de mutação) para este tipo de problemas, devendo ser aplicado tendo em conta exemplos reais concretos.

Relativamente a trabalho futuro, existem duas vertentes principais, nomeadamente, a vertente do desenvolvimento e a vertente da avaliação.

Quanto à vertente de desenvolvimento, a solução proposta tem uma grande margem para crescimento, pois a área do escalonamento da produção é vasta. De entre as possíveis implementações de melhoria, destacam-se:

- Investigação e implementação de novas funcionalidades, tanto em termos de algoritmo como em termos de escalonador;
- Implementação de critérios por recurso;
- Implementação de mecanismo para determinação dos parâmetros do algoritmo de forma automática;
- Implementação de novos critérios de otimização e/ou critérios dinâmicos;
- Implementação de funcionalidades relacionadas com inteligência artificial, com estudos de histórico, de forma a otimizar o escalonamento.

Relativamente à vertente de avaliação, destacam-se os seguintes pontos de melhoria:

- Avaliação da solução em ambientes reais, com seleção de clientes piloto;
- Avaliação no impacto de uma organização.

Referências

Adam DuVander (2013) *JSON's Eight Year Convergence With XML | ProgrammableWeb*. Available at: <https://www.programmableweb.com/news/jsons-eight-year-convergence-xml/2013/12/26> (Accessed: 26 January 2019).

Allahverdi, A. and Soroush, H. (2006) 'The significance of reducing setup times / setup costs', *The significance of reducing setup times/setup costs*. doi: 10.1016/j.ejor.2006.09.010.

Anderson, R. and Smith, S. (2020) *Format response data in ASP.NET Core Web API*. Available at: <https://docs.microsoft.com/en-us/aspnet/core/web-api/advanced/formatting?view=aspnetcore-3.1> (Accessed: 5 October 2020).

APICS (2013) *APICS Dictionary*.

Bermudez, C. et al. (2010) 'Comparison of Recombination Operators in Panmictic and Cellular GAs to Solve a Vehicle Routing Problem Genetic Algorithm for CVRP', *Inteligencia Artificial*, 46, pp. 34–44. doi: 10.4114/ia.v14i46.1520.

Bhandari, D., Murthy, C. A. and Pal, S. K. (2012) 'Variance as a Stopping Criterion for Genetic Algorithms with Elitist Model', *Fundamenta Informaticae* 120, 120, pp. 145–164. doi: 10.3233/FI-2012-754.

Daglayan, H. and Karakaya, M. (2016) 'The Impact of Crossover and Mutation Operators on a GA Solution for the Capacitated Vehicle Routing Problem', *Universal Journal of Engineering Science*, 4(3), pp. 39–44. doi: 10.13189/ujes.2016.040301.

Davis, L. (1985) 'Applying Adaptive Algorithms to Epistatic Domains', *Proceedings of the 9th International Joint Conference on Artificial Intelligence - Volume 1*, 1, pp. 162–164. Available at: <http://dl.acm.org/citation.cfm?id=1625135.1625164>.

Dellaert, N., Jeunet, J. and Jonard, N. (2000) 'Genetic algorithm to solve the general multi-level lot-sizing problem with time-varying costs', *International Journal of Production Economics*, 68(3), pp. 241–257. doi: 10.1016/S0925-5273(00)00084-0.

Dinis-Carvalho, J. (2008) 'Fundamentos de Produção', 2014, pp. 1–13. doi: 10.1017/CBO9781107415324.004.

Eeles, P. (2004) *What, no supplementary specification?* Available at: <https://www.ibm.com/developerworks/rational/library/3975.html> (Accessed: 11 February 2019).

Fishburn, P. C. (1967) 'Additive Utilities with Incomplete Product Sets: Application to Priorities and Assignments', *Operations Research*, 15(3), pp. 537–542. doi: 10.1287/opre.15.3.537.

Fisher, R. A. and Yates, F. (1938) 'Statistical Tables', *Statistical tables for biological, agricultural and medical research (3rd ed.)*, pp. 26–27. doi: 10.1017/S1755146300183556.

Fonseca, M. (2013) 'Sistema integrado de balanceamento de linhas de produção na indústria

do calçado'. Available at: <http://repositorio-aberto.up.pt/handle/10216/57575>.

Gallego, G. (2003) 'Material Requirements Planning (MRP) Material', (C), pp. 1–6. Available at: <https://www.pharma-excipients.ch/basics/>.

Gendreau, M. (2006) 'Chapter 2: AN INTRODUCTION TO TABU SEARCH', *Search Methodologies*, Springer, pp. 165–186. doi: 10.1007/0-306-48056-5.

Genero, M., Jiménez, L. and Piattini, M. (2000) 'Conceptual Modeling - ER 2000 ontological analysis of taxonomic relationships P226', *19th International Conference on Conceptual Modeling Salt Lake City, Utah, USA, October 9-12, 2000*, (October), pp. 513–525. doi: 10.1007/3-540-45393-8.

Hertz, A. *et al.* (2000) 'A TUTORIAL ON TABU SEARCH Alain'.

Holland, J. (1975) 'Adaptation in natural and artificial systems', *University of Michigan, Ann Arbor*.

José Lindenberg (2016) *Ilustração do funcionamento dos algoritmos genéticos., Um Sistema de Informação para Melhoramento Genético de Caprinos e Ovinos*. Available at: https://www.researchgate.net/figure/Figura-3-Illustracao-do-funcionamento-dos-algoritmos-geneticos_fig3_303707833 (Accessed: 17 January 2019).

Kljajić, M., Bernik, I. and Breskvar, U. (2014) 'Production planning using simulation and genetic algorithms in multi-criteria scheduling optimization', *SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.98CH36218)*, 1(2), pp. 445–450. doi: 10.1109/ICSMC.1998.725452.

Koen, P. *et al.* (2001) 'PROVIDING CLARITY AND A COMMON LANGUAGE TO THE " FUZZY FRONT END "', pp. 46–55.

Kolisch, R. and Heß, K. (1998) 'Efficient methods for scheduling make-to-order assemblies under resource, assembly area, and part availability constraints', *Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel*, No. 474.

Kruchten, P. and Rational Software (1995) 'The 4+ 1 view model of architecture', *Software, IEEE*, (November). doi: 10.1109/52.469759.

Li, L. (1988) *Make-To-Order vs. Make-To-Stock: The Role of Inventory in Delivery-Time Competition*. MASSACHUSETTS.

Madureira, A. (2013) 'Multi-agent system for distributed manufacturing scheduling with genetic algorithms and tabu search'.

Mcgreedy, J. (2006) 'The Paired t-test and Hypothesis Testing'.

Microsoft (2020) *GC.GetTotalMemory(Boolean) Method (System)*, *Microsoft Docs*. Available at: <https://docs.microsoft.com/en-us/dotnet/api/system.gc.gettotalmemory?view=netcore-3.1> (Accessed: 11 October 2020).

Mitchell, M. (1995) 'Genetic algorithms: An overview', *Complexity*, 1(1), pp. 31–39. doi: 10.1002/cplx.6130010108.

- Moon, C. *et al.* (2006) 'Adaptive genetic algorithm for advanced planning in manufacturing supply chain', *Journal of Intelligent Manufacturing*, 17(4), pp. 509–522. doi: 10.1007/s10845-005-0010-0.
- Moses, S. A. and Sangplung, W. (2017) 'Resource planning for just-in-time make-to-order environments: A scalable methodology using tabu search', *Cogent Engineering*, 4(1), pp. 1–19. doi: 10.1080/23311916.2017.1341289.
- Osterwalder, A. *et al.* (2010) *You're holding a handbook for visionaries, game changers, and challengers striving to defy outmoded business models and design tomorrow's enterprises. It's a book for the . . . written by.* Edited by T. Clark. John Wiley & Sons, Inc., Hoboken, New Jersey.
- Palacios, J. J. *et al.* (2015) 'Genetic tabu search for the fuzzy flexible job shop problem', *Computers and Operations Research*, 54, pp. 74–89. doi: 10.1016/j.cor.2014.08.023.
- Pinedo, M. L. (2008) *Scheduling: Theory, algorithms, and systems, Scheduling: Theory, Algorithms, and Systems.* doi: 10.1007/978-0-387-78935-4.
- Rodrigues, E. E. da S. (2000) 'Algoritmos genéticos - Escalonamento automático de reuniões escolares'.
- Rossit, D. A., Tohmé, F. and Frutos, M. (2019) 'Industry 4.0: Smart Scheduling', *International Journal of Production Research*, 57(12), pp. 3802–3813. doi: 10.1080/00207543.2018.1504248.
- Roth, D., Rick, A. and Luttin, S. (2019) *Introduction to ASP.NET Core.* Available at: <https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-2.2> (Accessed: 23 February 2020).
- Safe, M. *et al.* (2004) 'On Stopping Criteria for Genetic Algorithms', *Lecture Notes in Computer Science.*, pp. 405–413.
- Schneider, U. (2011) 'A tabu search tutorial based on a real-world scheduling problem', *CEJOR*, pp. 467–493. doi: 10.1007/s10100-010-0137-8.
- Shaban, O. and Al-zubi, Z. (2014) 'The Effect of Using Break-Even-Point in Planning , Controlling , and Decision Making in the Industrial Jordanian Companies', *The Effect of Using Break-Even-Point in Planning, Controlling, and Decision Making in the Industrial Jordanian Companies*, (May). doi: 10.6007/IJARBS/v4-i5/888.
- Silva, A. C. S. *et al.* (2017) 'Indústria 4.0: proposta de mapa conceitual', *XXXVII Encontro Nacional De Engenharia De Producao, Joinville*, 1(1), pp. 1–10.
- Sistrade Software Consulting, S. A. (2018) *Planeamento da Produção - Sistrade MIS/ERP Software.* Available at: <http://www.sistrade.com/pt/solucoes/gestao-producao/scheduling/> (Accessed: 26 January 2019).
- Sistrade Software Consulting, S. A. (2020) *Gestão da Produção | Manufacturing 4.0 - Sistrade MIS/ERP Software.* Available at: <https://www.sistrade.com/pt/solucoes/gestao-producao/manufacturing-40/> (Accessed: 16 February 2020).

Study, C., Mohamed, A. M. and Coutry, N. (2015) 'Analysis of Lead Time Delays in Supply Chain : A', *International Journal of Social, Behavioral, Educational, Economic, Business and Industrial Engineering*, 9(6), pp. 2065–2070.

Toub, S. (2017) *Performance Improvements in .NET Core*. Available at: <https://devblogs.microsoft.com/dotnet/performance-improvements-in-net-core/> (Accessed: 5 October 2020).

Trovinger, S. C. and Bohn, R. E. (2005) 'Setup time reduction for electronics assembly: Combining simple (SMED) and IT-based methods', *Production and Operations Management*, 14(2), pp. 205–217. doi: 10.1111/j.1937-5956.2005.tb00019.x.

Villalobos-Arias, M., Coello, C. A. C. and Hernández-Lerma, O. (2006) 'Asymptotic convergence of metaheuristics for multiobjective optimization problems', *Soft Computing*, 10(11), pp. 1001–1005. doi: 10.1007/s00500-005-0027-5.

Widmer, M., Hertz, A. and Costa, D. (2010) 'Metaheuristics and Scheduling', *Production Scheduling*, pp. 33–68. doi: 10.1002/9780470611050.ch3.