



FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

DEPARTAMENTO DE
ENGENHARIA MECÂNICA

Desenvolvimento de folhas de cálculo para o apoio a tarefas de sequenciamento da produção

Dissertação apresentada para a obtenção do grau de Mestre em Engenharia e Gestão Industrial

Autor

Tiago Rafael Silva Cardoso

Orientador

Prof. Cristóvão Silva

Júri

Presidente Professor Doutor José Luís Ferreira Afonso
Professor da Universidade de Coimbra

Vogal Professora Doutora Marta Cristina Cardoso de Oliveira
Professora da Universidade de Coimbra

Agradecimentos

Começo por me dirigir à minha família, por todos os sacrifícios, carinho, e apoio incondicional. Muito obrigado ao meu Pai, Mãe, Irmã, e Avós. Só com a ajuda deles foi possível chegar até aqui.

Um palavra também para o Prof. Cristóvão Silva pela oportunidade que me ofereceu de trabalhar num tema que é do meu interesse, e pela disponibilidade demonstrada ao longo deste caminho.

Agradeço também à minha namorada por toda a paciência, carinho, e conselhos que fizeram com que me mantivesse focado e dedicado.

Finalizo com um agradecimento a todos os meus amigos, àqueles que fiz neste trajeto em Coimbra, e que sei ficarão para a vida, e aos da terrinha, os de sempre. A todos muito obrigado pelo apoio, palavras de incentivo, e pelos momentos de descontração, também eles essenciais.

Resumo

As ferramentas de sequenciamento da produção são um objeto essencial a um gestor industrial, no entanto o seu lecionamento nem sempre se mostra um processo fácil. A resolução deste tipo de problemas requer, em certas regras, um número elevado de cálculos, tornando assim a aprendizagem num processo moroso e nada eficiente. Para combater este fato, este trabalho tem como objetivo o desenvolvimento em Excel, módulo *Developer (Visual Basic)*, de ferramentas que permitam uma rápida resolução deste tipo de problemas, e que possibilitem o acompanhamento e fácil compreensão de todo o processo. Assim são abordadas ferramentas para resolução de problemas de sequenciamento para Máquina Única: *EDD*, *SPT*, *LPUL*, *COVERT*, *CR*, *Backward-Forward*, e *Backward-Forward with Early Penalties*, para configurações *Flow-Shop* os algoritmos *Duas Máquinas*, *Três Máquinas*, *n - Trabalhos m - Máquinas*, e ainda uma heurística para problemas *Máquinas Paralelas*. Com a programação destes algoritmos, ao utilizador compete apenas inserir os dados do(s) problema(s), e utilizar os botões de comando para interagir com as ferramentas, e obter a solução para o sequenciamento.

Palavras-chave: Sequenciamento, Produção, métodos heurísticos.

Abstract

The heuristic tools for production scheduling are essential for every production engineer. Even if they don't make any use of them, they will benefit from knowing of its existence and how they work. The resolution of this sort of problems requires, in some of them, a lot of calculation, what makes the process of teaching them, long and not efficient. In order to change this situation, this study, has as its goals the development of simple tools in Excel's Developer (Visual Basic), whose purpose is to help students performing a quick resolution of the problems, offering the possibility of doing it step by step, allowing this way a better comprehension of all process. Thus, the heuristics that will be studied in this work are, for single machine problems: EDD, SPT, LPUL, COVERT, CR, Backward-Forward, and Backward-Forward with Early Penalties, for Flow-Shop configurations: the Two-Machine problem, the Three-Machine problem, the n-Job m-machine problem, and also an heuristic for Parallel-Processing problems. With the programming of these tools, the user only has to insert the data of the problem and make use of the buttons to interact with the the Excel sheet, and get the solution for the scheduling problem.

Keywords Scheduling, production, heuristic.

Índice

Índice de Figuras	xi
Índice de Tabelas	xiii
Siglas	xv
1. Introdução	1
2. Sequenciamento da Produção	5
3. Ferramentas de Sequenciamento	7
3.1. Máquina Única	7
3.1.1. Earlier Due Date (EDD)	8
3.1.2. Shortest Processing Time (SPT)	10
3.1.3. Largest Penalty per Unit Length (LPUL)	12
3.1.4. Cost Over Time (COVERT)	13
3.1.5. Critical Ratio (CR)	16
3.1.6. Backward-Forward (BF)	17
3.1.7. Backward-Forward com Penalizações de Atraso e Antecipação	21
3.2. Problemas Flowshop	22
3.2.1. Duas Máquinas	24
3.2.2. Três Máquinas	27
3.2.3. n - Trabalhos m - Máquinas	29
3.3. Máquinas Paralelas	34
4. Conclusões	39
5. Bibliografia	43
ANEXO A – Código Máquina Única	45
ANEXO B – Código Duas Máquinas	51
ANEXO C – Código Três Máquinas	55
ANEXO D – Código n - Trabalhos m - Máquinas	61
ANEXO E – Código máquinas Paralelas	69

ÍNDICE DE FIGURAS

Figura 1 - Comandos presentes na folha EDD	9
Figura 2 - Comandos presentes na folha <i>SPT</i>	11
Figura 3 - Comandos presentes na folha <i>LPUL</i>	12
Figura 4 - Comandos presentes na folha <i>COVERT</i>	14
Figura 5 - Apresentação da solução.....	15
Figura 6 - Comandos presentes na folha <i>CR</i>	16
Figura 7 - Folha <i>CR</i> após sequenciamento concluído	17
Figura 8 - Comandos folha "B-F".....	20
Figura 9 - Comandos folha <i>B-F Early</i>	22
Figura 10 - Esquema de um <i>Flow-Shop</i>	23
Figura 11- Apresentação da solução num problema Duas-Máquinas	25
Figura 12 - Apresentação dos vários elementos presentes no esesquema da solução.....	25
Figura 13 - Comandos presentes na folha <i>Duas-Máquinas</i>	26
Figura 14 - Apresentação da solução para um problema <i>Três Máquinas</i>	28
Figura 15 - Comandos utilizados na folha <i>Três Máquinas</i>	28
Figura 16 - Esquema de apresentação de alguns dados relativos à solução encontrada	28
Figura 17 - Primeiro trabalho a ser sequenciado, e esquema de apresentação	30
Figura 18 - Comandos utilizados na folha <i>n - Trabalhos m - Máquinas</i>	31
Figura 19 - Sequências geradas duante a resolução	33
Figura 20 - Apresentação dos resultados.....	33
Figura 21 - Comandos utilizados na folha <i>Máquinas Paralelas</i>	37
Figura 22 - Makespan da solução alcançada	37

ÍNDICE DE TABELAS

Tabela 1 - Folha <i>Dados</i>	8
Tabela 2 – Folha <i>EDD</i> após dados importados	9
Tabela 3 - Após sequenciamento <i>EDD</i> concluído.....	10
Tabela 4 – Folha <i>SPT</i> após dados importados.....	11
Tabela 5 - Após sequenciamento <i>SPT</i> concluído	11
Tabela 6 - Folha <i>LPUL</i> após dados importados	12
Tabela 7 - Após sequenciamento segundo a regra <i>LPUL</i>	13
Tabela 8 - Folha <i>COVERT</i> após dados importados	14
Tabela 9 - Apresentação das variáveis ao longo da resolução	14
Tabela 10 - Dois trabalhos já sequenciados.....	15
Tabela 11 - Valores actuais das variáveis.....	15
Tabela 12 - Tabela final, com sequenciamento já concluído	15
Tabela 13- Valores finais das variáveis	15
Tabela 14 - Folha <i>CR</i> após dados importados	16
Tabela 15 - Tabela que acompanha a variação de "T"	16
Tabela 16 - Folha <i>B-F</i> após dados importados	18
Tabela 17 - Variáveis na fase <i>Backward</i>	18
Tabela 18 - Após conclusão da fase <i>Backward</i>	19
Tabela 19 - Variáveis na fase <i>Forward</i>	20
Tabela 20 - Sequenciamento concluído após realizada fase <i>Forward</i>	21
Tabela 21 - Folha <i>B-F Early</i> após dados importados	21
Tabela 22 - Folha <i>B-F Early</i> após resolução	22
Tabela 23- Dados iniciais do problema Duas-Máquinas.....	24

Tabela 24 - Apresentação de alguns dados relativos à solução	26
Tabela 25 - Tabela resultante da transformação da anterior.....	27
Tabela 26 - Tabela onde são inseridos os dados iniciais	27
Tabela 27 - Local onde são definidos os números de trabalhos e máquinas do problema	29
Tabela 28 - Tabela onde são inseridos os dados iniciais	30
Tabela 29 - Trabalhos em teste, com vista a ocupar a próxima posição da sequência.....	31
Tabela 30 - Primeiro empate	32
Tabela 31 - Tabela onde são inseridos os dados iniciais de um problema Máquinas Paralelas.....	35
Tabela 32 - Tabela já com os Tempos de Processamento ordenados de forma decrescente	35
Tabela 33 - Local onde se insere o número de máquinas do problema.....	35
Tabela 34 - Esquema da tabela onde será resolvido o problema.....	35
Tabela 35 - Tabela de resolução já com o primeiro trabalho sequenciado.....	36
Tabela 36 - Tabela já com o sequenciamento concluído.....	36

SIGLAS

EDD – Earlier Due Date

SPT – Shortest Processing Time

LPUL – Largest Penalty per Unit Length

COVERT – Cost Over Time

CR – Critical Ratio

BF – Backward-Forward

1. INTRODUÇÃO

O Sequenciamento é o ato de definir prioridades e de alocar tarefas a um certo número de recursos, com o intuito de cumprir determinados requisitos ou objetivos. Esta é uma definição possível, que esclarece bem aquilo que se pode entender por sequenciamento, e que mostra que o ato de sequenciar está presente em vários aspectos da nossa vida, e não apenas na indústria. No próprio dia-a-dia as pessoas acabam por utilizar o sequenciamento, e sem sequer se aperceberem, pois trata-se já de uma função intrínseca a cada um. Todavia este fator não desvaloriza a importância do sequenciamento mesmo a este nível, por exemplo caso as pessoas não definissem o seu horário de acordar, trabalhar, brincar, e dormir, por certo as suas vidas seriam um caos. Outro exemplo, e com o qual alunos e professores facilmente se identificam, são as universidades, pois estas têm que dividir vários cursos, e suas respectivas unidades curriculares, por um limitado número de salas. Ora, isto só se torna possível recorrendo ao sequenciamento, ou seja, a realização de horários. Estes permitem que várias aulas ocorram simultaneamente em diferentes salas, e ao mesmo tempo garante que cada aula tem o seu espaço ao longo das várias semanas do semestre.

Como já se percebeu o sequenciamento está presente em inúmeras áreas e é fundamental tanto a nível pessoal, como organizacional, no entanto na presente dissertação o foco será mantido no sequenciamento da produção, numa óptica industrial. Numa área como esta em que é essencial obter vantagem competitiva, e em que se procura constantemente reduzir custos, o tempo tem obrigatoriamente que se tornar um recurso valiosíssimo, e conseqüentemente, o sequenciamento da produção uma ferramenta indispensável.

“O sequenciamento é a afetação óptima no tempo de recursos escassos na forma de máquinas, a atividades designadas por tarefas, sujeita às restrições básicas de que em qualquer instante nenhuma máquina processa mais do que uma tarefa e nenhuma tarefa é processada por mais do que uma máquina.” (Lenstra e Rinnoy Kan, 1983). O sequenciamento é uma tarefa complexa que envolve a manipulação de muitos dados e o

recurso a regras e/ou algoritmos cuja complexidade depende da dimensão e tipologia do problema que se pretende resolver.

A utilização de regras de sequenciamento em ambiente industrial assume uma importância muito relevante, podendo permitir que uma unidade fabril cumpra os objetivos definidos, seja em termos de cumprimentos de prazos de entrega, minimização de stocks ou utilização dos equipamentos. Muitas vezes o sequenciamento ainda é realizado por responsáveis de produção de uma forma ad-hoc, recorrendo a regras simplistas e baseadas em experiência e processos de tentativa e erro. Isso deve-se em grande parte aos processos complexos envolvidos em muitos algoritmos de sequenciamento que não são suficientemente dominados na prática.

Por outro lado, o ensino dos problemas de sequenciamento em ambiente académico também é complicado. Muitas das regras consistem em processo iterativos morosos o que impede que os alunos possam realizar exercícios e resolver problemas com dimensão próxima daquela encontrada na prática o que pode levar a uma falta de motivação para o processo de aprendizagem.

Assim, o objetivo deste trabalho é desenvolver em Excel vários algoritmos de sequenciamento, que facilitem e tornem mais rápida a obtenção de resultados a partir destes. É um fato que estas ferramentas produzem bons resultados, mesmo que nem sempre cheguem à solução óptima, mas também é sabido que grande parte delas são muito trabalhosas e implicam um elevado número de cálculos, consumindo, conseqüentemente, bastante tempo por parte de quem as resolve. No final deste estudo, e com estas ferramentas programadas através do módulo *Developer (Visual Basic)* em várias folhas de cálculo, pretende-se que este processo seja rápido e intuitivo.

Ainda que o desenvolvimento das folhas de cálculo tenha sido sugerido com o propósito de servir de apoio nas aulas da unidade curricular Gestão da Produção, ou em qualquer outra em que o tema Sequenciamento da Produção seja lecionado, estas podem também ser utilizadas em pequenas e médias organizações em que este tipo de cálculos continuem a ser efetuados à mão, ou através dos métodos *knowledge-based*.

Como existem várias ferramentas de sequenciamento, a prioridade será dada àquelas que são lecionadas nas aulas de Gestão da Produção. Estas têm no entanto também relevância a nível industrial, pois só assim faria sentido estarem a ser lecionadas num curso em que esta unidade curricular assume um papel tão importante, e cuja probabilidade dos alunos lidarem com tais ferramentas nos seus futuros profissionais é grande.

Este documento encontra-se estruturado da seguinte forma. Neste capítulo faz-se uma breve introdução ao trabalho desenvolvido referindo as motivações para a sua realização. No Capítulo 2 é feita uma breve introdução aos problemas de sequenciamento e formas de resolução. No Capítulo 3 é explicada cada uma das ferramentas abordadas neste estudo, através da apresentação da forma como estas foram desenvolvidas e um exemplo de aplicação de cada uma delas nas respetivas folhas de cálculo. No Capítulo 4 é feita uma avaliação ao trabalho desenvolvido, abordando os objectivos propostos e seu cumprimento, e as dificuldades encontradas para os atingir. São ainda deixadas algumas sugestões de trabalho futuro, tendo como base este estudo.

2. SEQUENCIAMENTO DA PRODUÇÃO

“O sequenciamento da produção consiste na alocação de recursos ao longo do tempo, para a realização de um conjunto de tarefas”, (Baker, 1974). O sequenciamento da produção só pode ser realizado após a tomada de um conjunto de decisões de planeamento da produção fundamentais. Em termos simplistas, existem três decisões de planeamento da produção fundamentais a tomar: (1) que produto ou serviço se deve oferecer; (2) em que escala (quantidade a produzir) deve ser oferecido e (3) que recursos existem para produzir o produto/serviço a oferecer. A resposta a estas questões consistem na função planeamento da produção. Só depois de responder a essas perguntas se pode passar à função sequenciamento. Assim, a função sequenciamento só é relevante para sistemas para os quais a natureza das tarefas a executar foi definida e a configuração dos recursos disponíveis foi determinada.

Os objetivos que se pretendem alcançar com o sequenciamento da produção podem ser vários: minimizar o makespan (tempo total necessário para processar todas as ordens de fabrico), maximizar a percentagem de utilização de recursos, minimizar o número de ordens de fabrico concluídas após a data de entrega, minimizar o maior atraso, minimizar o atraso médio, minimizar o somatório dos atrasos, minimizar o valor médio do stock intermédio.

Também a configuração dos recursos disponíveis difere de organização para organização, e os processos pelos quais os materiais passam variam de acordo com o produto e com a exigência qualitativa a que as empresas se propõe, o que faz com que o modelo de sequenciamento tenha que ser adaptado às características do ambiente industrial em que este se pretende realizar. Assim os modelos de sequenciamento podem ser divididos em várias categorias: máquina única, flow-shop (as ordens de fabrico são processadas sequencialmente em vários equipamentos, e a sequência de processamento é comum a todas as ordens de fabrico), máquinas paralelas (conjunto de máquinas, idênticas ou não, e cada ordem de fabrico deve ser processada numa delas), job-shop (existe um conjunto de equipamentos, e uma ordem de produção pode necessitar de ser processada nalguns ou em todos os equipamentos, de acordo com uma sequência pré-determinada), open-shop (similar ao job-shop, mas as ordens de produção podem ser processadas em

qualquer sequência), loteamento e sequenciamento (as ordens de produção são sequenciadas em lotes).

Os exemplos, acima apresentados, para objetivos e configurações de recursos associados a problemas de sequenciamento não pretendem ser exaustivos. Brucker (1995) refere que a teoria do sequenciamento é caracterizada por um número praticamente ilimitado de problemas. O leitor interessado em conhecer as tipologias e classificações dos problemas de sequenciamento poderá recorrer a Brucker (1995) ou Blazewicz (1996).

Segundo Sule (1997) os métodos que nos permitem sequenciar os modelos acima referidos podem ser divididos em dois grupos. Assim, existem os métodos que requerem menos esforços, também designados por *knowledge-based*, que são apenas praticáveis no ambiente de fábrica em que foram desenvolvidos, e que se criam recorrendo a um pensamento “se...então”, geralmente utilizando apenas um lápis, gráficos e diagramas, e que podem ou não levar à solução ótima. Por outro lado existem os mais complexos, denominados algoritmos de sequenciamento, e que tendem a chegar a soluções ótimas para determinados critérios, como por exemplo, minimizar o desvio do fim do processamento das ordens de fabrico e suas datas de entrega, minimizar penalizações, e minimizar o maior atraso. No entanto, as soluções algorítmicas envolvem cálculos matemáticos e requerem um maior nível de conhecimentos, o que pode levar a que os gestores não sintam confiança nos resultados que estes métodos geram, e assim os descartem.

Neste trabalho, sendo o objetivo essencial o desenvolvimento de uma ferramenta de apoio ao ensino do problema de sequenciamento, serão utilizados métodos heurísticos. Estes, são menos complexos facilitando o processo de ensino/aprendizagem, permitindo, ainda assim, transmitir os fundamentos associados ao problema de sequenciamento, no que diz respeito ao seu objetivo, restrições e configurações mais vulgares em meio industrial.

3. FERRAMENTAS DE SEQUENCIAMENTO

Neste capítulo faz-se uma descrição dos algoritmos implementados na folha de cálculo desenvolvida ao longo do trabalho realizado. Em cada sub-capítulo refere-se os algoritmos desenvolvidos para as configurações de recursos escolhidas para este trabalho fazendo-se uma breve apresentação do modo de funcionamento do algoritmo/regra e de como esta foi implementada em Excel.

3.1. Máquina Única

Sequenciar um determinado número de trabalhos numa única máquina, de forma a otimizar certos objetivos é o que se pretende neste tipo de sequenciamento. Existem vários tipos de objetivos, sendo que uns se sobrepõem a outros dependendo daquilo que se pretende, e existem também variações nos problemas e soluções utilizadas para a resolução destes. Enquanto que por vezes o mais importante pode ser diminuir as penalizações dos trabalhos em que há atraso, noutras situações a prioridade pode passar por cumprir um número específico de datas de entrega. Nesta seção serão abordadas várias ferramentas de sequenciamento, cada uma respondendo a um determinado tipo de problema, e procurando assim alcançar soluções para um vasto leque de situações.

Como as variáveis utilizadas na resolução das diversas ferramentas de sequenciamento em máquina única não variam, a primeira folha criada no documento Excel, à qual se chamou *Dados*, destina-se à introdução dos dados do problema, que depois serão copiados para as folhas dos respetivos algoritmos, evitando assim a introdução dos mesmos inúmeras vezes.

Na Tabela 1 é apresentada a tabela presente na folha *Dados*, onde a primeira linha, que contém os títulos de cada coluna, já estará preenchida, sendo que as outras terão de ser adicionadas com os dados relativos a cada problema.

OF	Numero de Trabalho	Tempo de Processamento, Pi	Data de Entrega, Di	Late Penalty, Li	Early Penalty, Ei
1	1	37	49	1	2
2	2	27	36	5	0
3	3	1	1	1	3
4	4	28	37	5	2

Tabela 1 - Folha *Dados*

- **OF (Ordem de Fabrico)** – variável adicionada para apoiar na programação. Determina a ordem pela qual os trabalhos vão ser produzidos.

- **Número de Trabalho** – trabalho associado a uma operação de fabrico. O tempo de processamento, data de entrega, e penalizações estarão sempre relacionados com o seu número de trabalho, e acompanhá-lo-ão durante toda a resolução do sequenciamento.

- **Tempo de Processamento, Pi** – unidades de tempo durante as quais um trabalho ocupa a máquina, ou seja, desde o início da sua realização até estar completo.

- **Data de Entrega, Di** – unidades de tempo que, no instante 0, ou seja, quando se dá início ao processamento da primeira ordem de fabrico, restam até à data em que o trabalho devia estar completo.

- **Late Penalty, Li** - valor que deve ser somado por cada dia em que a data de entrega é ultrapassada.

- **Early Penalty, Ei** - valor que deve ser somado por cada dia em que o trabalho fica completo antes da data de entrega.

3.1.1. Earlier Due Date (EDD)

Nesta regra o sequenciamento é desenvolvido considerando apenas as datas de entrega de cada trabalho. Assim, o trabalho cuja data de entrega esteja mais próxima será realizado em primeiro lugar, e de seguida entra em máquina aquele que, de entre os trabalhos ainda não sequenciados, tenha a data de entrega mais próxima, ou seja, as datas de entrega são colocadas por ordem crescente.

Esta é uma regra muito intuitiva, seguida muitas vezes na programação das nossas atividades, em que se faz em primeiro lugar o trabalho mais urgente.

Os dados do problema de tese utilizado, seguindo a regra EDD, encontram-se apresentados na Tabela 2.

OF	Numero de Trabalho	Tempo de Processamento, Pi	Data de Entrega, Di	Late Penalty, Li	Atraso	Penalização
1	3	1	1	1	0	0
2	2	27	36	5	8	0
3	4	28	37	5	-19	95
4	1	37	49	1	-44	44
Total		93			-63	139

Tabela 2 – Folha EDD após dados importados

Esta é a tabela que se encontra na folha dedicada à regra EDD. Como se pode observar, nesta tabela foi descartada a variável *Early Penalty*, isto porque na regra EDD pretende-se minimizar o atraso na entrega de trabalhos, não interessando se estes são finalizados em cima do prazo definido para entrega ou muito tempo antes. Assim, apenas são acionadas penalizações para os trabalhos cuja conclusão ultrapasse a data de entrega. De notar também que foram acrescentadas duas colunas, uma que contém os atrasos, e outra as penalizações relativas a cada trabalho, e ainda uma linha onde são mostrados os totais destas. Na coluna *Atrasos*, a verde, constarão os trabalhos cuja sua produção foi concluída antes do prazo de entrega, mas apenas por uma questão informativa. Os números a vermelho, esses sim relevantes para este problema, indicam que houve um atraso nessa quantidade de unidades de tempo, e são multiplicados pelo valor do respetivo penalty para assim se obter a penalização proveniente desse trabalho.

Nesta folha serão usados três botões, cada um com um determinada função:



Figura 1 - Comandos presentes na folha EDD

- **Importar Dados Iniciais** – copia os dados da tabela da folha *Dados*;
- **EDD** – faz o sequenciamento segundo a regra EDD, com apenas um clique;

- **EDD Progressivo** – faz o sequenciamento segundo a regra EDD, mas fá-lo por etapas. No primeiro clique encontra o trabalho com a data de entrega mais próxima, move-o para a primeira posição, e depois pára. Ao segundo procura, nos trabalhos por sequenciar, aquele com o prazo de entrega mais pequeno e move-o para a segunda posição, parando logo de seguida. Este processo continua até não existirem mais trabalhos por sequenciar. Nessa fase se o botão for premido nada acontece.

Após a inserção dos dados na primeira folha, estes botões, no caso das regras para *Máquina Única* servirão como única interação necessária para a resolução do sequenciamento.

OF	Numero de Trabalho	Tempo de Processamento, Pi	Data de Entrega, Di	Late Penalty, Li	Atraso	Penalização
1	3	1	1	1	0	0
2	2	27	36	5	8	0
3	4	28	37	5	-19	95
4	1	37	49	1	-44	44
Total		93			-63	139

Sequência da Produção	3 2 4 1
-----------------------	---------

Tabela 3 - Após sequenciamento EDD concluído

Com a utilização destes, e após o sequenciamento estar concluído, poderemos ver qual a sua ordem a partir da coluna *OF*, ou através da informação *Sequência da Produção*, que será apresentada ao longo de todo o processo, e cujo fundo se tornará verde assim que todos os trabalhos estejam sequenciados. Na Tabela 3 pode observar-se a tabela da folha *Dados* após o sequenciamento estar concluído.

3.1.2. Shortest Processing Time (SPT)

O SPT é em tudo semelhante à regra EDD, com a diferença de que são colocados os tempos de processamento por ordem decrescente, em vez das datas de entrega. Assim o trabalho que tiver menor tempo de processamento é colocado na primeira posição, na segunda é colocado aquele que de entre os trabalhos que ficam por sequenciar tem menor tempo de sequenciamento, e assim sucessivamente até todos os trabalhos estarem sequenciados.

OF	Numero de Trabalho	Tempo de Processamento, Pi	Data de Entrega, Di	Late Penalty, Li	Atraso	Penalização
1	3	1	1	1	0	0
2	2	27	36	5	8	0
3	4	28	37	5	-19	95
4	1	37	49	1	-44	44
Total		93			-63	139

Tabela 4 – Folha SPT após dados importados

A tabela inicial é a mesma que aquela utilizada na folha *EDD*, variando apenas o seu aspecto final, consequência das diferenças entre as regras (em casos pontuais ambas as regras podem apresentar o mesmo resultado).

O modelo de funcionamento é o mesmo, ou seja três botões, com funções idênticas às utilizadas na ferramenta *EDD* mas adaptadas à regra *SPT*.



Figura 2 - Comandos presentes na folha SPT

- **Importar Dados Iniciais** – copia os dados da tabela da folha *Dados*;
- **SPT** – faz o sequenciamento segundo a regra *SPT*, com apenas um clique;
- **SPT Progressivo** – faz o sequenciamento segundo a regra *SPT*, mas fá-lo por etapas. No primeiro clique encontra o trabalho com o menor tempo de processamento, move-o para a primeira posição, e depois pára. Ao segundo procura, nos trabalhos por sequenciar, aquele com o tempo de processamento mais pequeno e move-o para a segunda posição, parando logo de seguida. Este processo continua até não existirem mais trabalhos por sequenciar. Nessa fase se o botão for premido nada acontece.

OF	Numero de Trabalho	Tempo de Processamento, Pi	Data de Entrega, Di	Late Penalty, Li	Atraso	Penalização
1	3	1	1	1	0	0
2	2	27	36	5	8	0
3	4	28	37	5	-19	95
4	1	37	49	1	-44	44
Total		93			-63	139

Sequência da Produção

3 2 4 1

Tabela 5 - Após sequenciamento SPT concluído

A Tabela 5 apresenta a solução do problema, cujo sequenciamento poderá ser visto a partir da coluna *OF*, ou através da informação *Sequência da Produção*.

3.1.3. Largest Penalty per Unit Length (LPUL)

Com esta regra entra-se agora num tipo de algoritmos em que serão utilizadas novas variáveis, que diferem de regra para regra, e cujos valores são atingidos através de cálculos, onde se utilizam os dados iniciais do problema. Nesta em específico calcula-se a penalização por tempo de processamento, sendo U a designação desta nova variável, e assim: $U_i = L_i / P_i$. Esta variável é calculado para todos os trabalhos, e estes serão sequenciados de forma decrescente, relativamente ao seu valor de U .

OF	Numero do Trabalho	Tempo de Processamento, P_i	Data de Entrega, D_i	Late Penalty, L_i	Atraso	Penalização	$U_i = L_i / P_i$
1	1	37	49	1	12	0	0,027
2	2	27	36	5	-28	140	0,185
3	3	1	1	1	-64	64	1,000
4	4	28	37	5	-56	280	0,179
Total		93			-148	484	

Tabela 6 - Folha *LPUL* após dados importados

Como se pode observar na tabela anterior, existe apenas uma diferença para as tabelas das ferramentas anteriores, que é a introdução de uma coluna para a variável U , e seus respetivos valores relativos a cada trabalho. Estes valores são calculados automaticamente, assim que se importam os dados do problema para a folha desta ferramenta. As ordens de importação e de sequenciamento são mais uma vez dadas através de botões de comando.

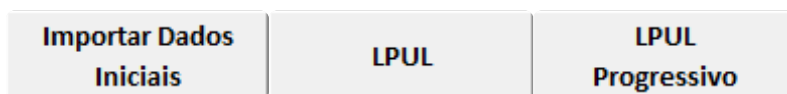


Figura 3 - Comandos presentes na folha *LPUL*

- **Importar Dados Iniciais** – copia os dados da tabela da folha *Dados*;
- **LPUL** – faz o sequenciamento segundo a regra LPUL, ou seja, ordena os trabalhos por ordem decrescente segundo o seu valor de U ;
- **LPUL Progressivo** – faz o sequenciamento segundo a regra *LPUL*, sequenciando um trabalho por cada clique. Assim, ao primeiro clique coloca o trabalho com maior valor de U na primeira posição. No segundo procura, de entre os trabalhos por

sequenciar, aquele com o maior U e coloca-o na segunda posição, e assim sucessivamente até todos os trabalhos estarem sequenciados.

OF	Numero do Trabalho	Tempo de Processamento, P_i	Data de Entrega, D_i	Late Penalty, L_i	Atraso	Penalização	$U_i = L_i / P_i$
1	3	1	1	1	0	0	-
2	2	27	36	5	8	0	-
3	4	28	37	5	-19	95	-
4	1	37	49	1	-44	44	-
Total		93			-63	139	
Sequência da Produção		3 2 4 1					

Tabela 7 - Após sequenciamento segundo a regra LPUL

Em cima pode ver-se o aspecto da folha *LPUL* após o sequenciamento estar completo.

3.1.4. Cost Over Time (COVERT)

Na realização do sequenciamento através da ferramenta COVERT utiliza-se já um algoritmo um pouco mais complexo. Antes de se passar à explicação desta regra serão apresentadas as siglas que serão utilizadas nos vários passos deste processo. Assim:

- TT – soma do tempo de processamento de todas as OF;
- RT – soma do tempo de processamento das OF por sequenciar;
- ST – duração atual da sequência, que é o mesmo que, tempo de início da próxima OF a ser sequenciada;

A regra COVERT é realizada em três passos:

1. Cálculo da Prioridade (PR) para todos os trabalhos por sequenciar. PR é calculado da seguinte forma:

- Se $D_i \leq (ST + P_i)$, então $PR = 1$
- Se $D_i > (ST + P_i)$ e $D_i < TT$, então $PR = (TT - D_i) / (RT - P_i)$
- Se $TT \leq D_i$, então $PR = 0$

2. Calcular o Coeficiente (CF) para os trabalhos por sequenciar:

- $CF = PR (L_i / P_i)$

3. O trabalho com maior CF é o próximo a ser sequenciado.

Após a realização destes três passos, ST e RT são recalculados, e volta-se ao passo número 1. O sequenciamento está concluído quando todos os trabalhos estiverem sequenciados.

OF	Numero do Trabalho	Tempo de Processamento, Pi	Data de Entrega, Di	Late Penalty, Li	Atraso	Penalização	PR	CF
1	1	37	49	1	12	0	0,786	0,021
2	2	27	36	5	-28	140	0,864	0,160
3	3	1	1	1	-64	64	1	1,000
4	4	28	37	5	-56	280	0,862	0,154
Total		93			-148	484		

Tabela 8 - Folha COVERT após dados importados

Na Tabela 8 pode observar-se que serão apresentados os dados relativos ao PR e CF de cada trabalho, ajudando assim a que se perceba o porquê de certo trabalho ter sido sequenciado numa dada posição. Estes dados serão atualizados à medida que o sequenciamento avança. Também estarão visíveis numa pequena tabela (Tabela 9), as variáveis utilizadas no cálculo da Prioridade e Coeficiente, e também estas serão actualizadas com o avançar do sequenciamento.

TT	93
RT	93
ST	0

Tabela 9 - Apresentação das variáveis ao longo da resolução

Em relação aos botões, nesta regra, tal como nas outras, a base das suas funções são as mesmas, sendo estes apenas adaptados ao algoritmo da folha da ferramenta em que se inserem.

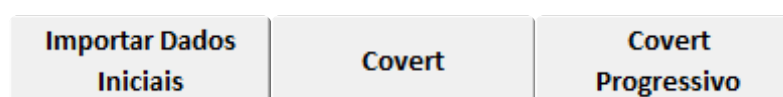


Figura 4 - Comandos presentes na folha COVERT

- **Importar Dados Iniciais** – copia os dados da tabela da folha *Dados*;
- **COVERT** – faz, através de um só clique, o sequenciamento segundo a regra COVERT;
- **COVERT Progressivo** – faz o sequenciamento segundo a regra COVERT, sequenciando um trabalho por cada clique. Assim, ao primeiro clique coloca o trabalho com maior CF na primeira posição e recalcula as variáveis ST, RT, e PR, de forma a obter novos valores de CF para os trabalhos que se encontram por sequenciar. No segundo clique volta a procurar, naqueles por sequenciar, o trabalho com maior CF, e coloca-o na

segunda posição, voltando depois a calcular os novos valores de ST, RT, PR, e CF. Este processo repete-se até estarem todos os trabalhos sequenciados.

A tabela seguinte representa um processo de sequenciamento em que se está a utilizar o botão *COVERT Progressivo*, e onde os dois primeiros trabalhos foram já sequenciados, e por isso nas colunas que PR e CF os seus valores foram substituídos por um traço.

OF	Numero do Trabalho	Tempo de Processamento, Pi	Data de Entrega, Di	Late Penalty, Li	Atraso	Penalização	PR	CF
1	3	1	1	1	0	0	-	-
2	2	27	36	5	8	0	-	-
3	1	37	49	1	-16	16	1,000	0,027
4	4	28	37	5	-56	280	1,000	0,179
Total		93			-72	296		

Tabela 10 - Dois trabalhos já sequenciados

Já na Tabela 11 pode observar-se que, nesta fase o tempo total de sequenciamento dos trabalhos por sequenciar equivale a 65 (37 + 28), enquanto que o dos trabalhos sequenciados é 28 (1 + 28).

TT	93
RT	65
ST	28

Tabela 11 - Valores actuais das variáveis

Concluído o sequenciamento, estas mesmas tabelas apresentariam o seguinte aspecto:

OF	Numero do Trabalho	Tempo de Processamento, Pi	Data de Entrega, Di	Late Penalty, Li	Atraso	Penalização	PR	CF
1	3	1	1	1	0	0	-	-
2	2	27	36	5	8	0	-	-
3	4	28	37	5	-19	95	-	-
4	1	37	49	1	-44	44	-	-
Total		93			-63	139		

Tabela 12 - Tabela final, com sequenciamento já concluído

TT	93
RT	0
ST	93

Tabela 13- Valores finais das variáveis

A sequência pela qual os trabalhos seriam produzidos pode ser observada, na mesma folha, como apresentado na Figura 5.

Sequência da Produção

3 2 4 1

Figura 5 - Apresentação da solução

3.1.5. Critical Ratio (CR)

Nesta ferramenta calculam-se os rácios críticos (CR) de todos os trabalhos por sequenciar, sendo que o trabalho a sequenciar na próxima posição é aquele com menor valor de CR. Para ser possível o cálculo de CR é necessário primeiro calcular T, que é a soma dos tempos de processamento de todos os trabalhos sequenciados. Após o sequenciamento de um trabalho o seu tempo de processamento é somado ao valor de T, e os valores de CR para os trabalhos por sequenciar recalculados, utilizando já o novo valor de T. A fórmula de CR é: $(D_i - T) / P_i$, em que D_i e P_i são, respectivamente, a data de entrega, e tempo de processamento do trabalho i .

OF	Numero do Trabalho	Tempo de Processamento, P_i	Data de Entrega, D_i	Late Penalty, L_i	Atraso	Penalização	$CR_i = (D_i - T)/P_i$	
	1	1	37	49	1	12	0	1,324
	2	2	27	36	5	-28	140	1,333
	3	3	1	1	1	-64	64	1,000
	4	4	28	37	5	-56	280	1,321
	Total		93			-148	484	

Tabela 14 - Folha CR após dados importados

A tabela é idêntica a todas as outras, mostrando neste caso, na última coluna, os dados relativos aos rácios críticos de cada trabalho. Após a importação dos dados iniciais para a folha desta ferramenta, os valores CR e de T (que inicialmente é zero – Tabela 15) são automaticamente calculados. À medida que o sequenciamento vai avançando estas variáveis serão recalculadas, e os seus novos valores actualizados nas respectivas tabelas. Os botões de comando, cujas funções serão apresentadas de seguida, podem ser observados na Figura 6.

T	0
---	---

Tabela 15 - Tabela que acompanha a variação de "T"

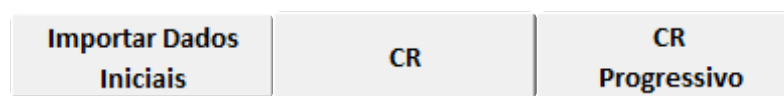


Figura 6 - Comandos presentes na folha CR

- **Importar Dados Iniciais** – copia os dados da tabela da folha *Dados*;
- **CR** – faz, através de um só clique, o sequenciamento segundo a regra CR;
- **CR Progressivo** – faz o sequenciamento segundo a regra CR, sequenciando um trabalho por cada clique. Assim, ao primeiro clique coloca o trabalho com menor CR na primeira posição, adiciona a T o valor do tempo de processamento do trabalho acabado

de sequenciar, e recalcula os valores de CR para os trabalhos por sequenciar, utilizando o novo valor de T. No segundo clique volta a procurar, naqueles por sequenciar, o trabalho com menor CR, e coloca-o na segunda posição, voltando depois a calcular os novos valores de T e CR. Este processo repete-se até todos os trabalhos estarem sequenciados.

Quando um trabalho é sequenciado o seu valor de CR é substituído por um traço, e sabemos que o sequenciamento está concluído quando todos os trabalhos tiverem “-“ como CR, ou quando o valor de T for igual à soma de todos os tempos de processamento. No fim, com o sequenciamento concluído, a folha relativa a esta ferramenta terá o aspeto que se pode observar na Figura 7, onde se inclui a tabela onde os dados foram manipulados, a tabela que apresentou ao longo de toda a resolução a variável T , e a célula da folha de cálculo onde é apresentada a sequência que foi alcançada.

OF	Numero do Trabalho	Tempo de Processamento, P_i	Data de Entrega, D_i	Late Penalty, L_i	Atraso	Penalização	$CR_i = (D_i - T)/P_i$	
1	3	1	1	1	0	0	-	T
2	4	28	37	5	8	0	-	93
3	2	27	36	5	-20	100	-	
4	1	37	49	1	-44	44	-	
Total		93			-64	144		
Sequência da Produção		3 4 2 1						

Figura 7 - Folha CR após sequenciamento concluído

3.1.6. Backward-Forward (BF)

As regras estudadas anteriormente são relativamente simples, o que faz com que não alcancem o resultado óptimo de uma forma consistente. A heurística Backward-Forward pretende assim colmatar esta falha pois apresenta resultados mais eficientes, e apesar de ser um pouco mais complexa é ainda assim fácil de aplicar. O procedimento é aplicado em duas fases: a *Backward* que é realizada em primeiro lugar, e a *Forward* que é utilizada para atingir melhorias numa segunda e última fase.

3.1.6.1. Backward

Nesta fase o sequenciamento é feito em sentido contrário, ou seja, é iniciado no último elemento da tabela e prosseguirá até se atingir a primeira posição desta. Em cada uma das posições um conjunto de passos terá que ser desenvolvido, sendo esses:

1. Ver qual a posição atual da sequência (valor do contador da posição), ou seja, qual o próximo trabalho a ser sequenciado. De notar que o valor inicial do contador é N (número total de trabalhos), e o sequenciamento será desenvolvido de forma descendente até ser atingida a posição 1;

2. Calcular o valor de T (soma do tempo de processamento de todos os trabalhos por sequenciar);

3. Calcular a penalização para cada trabalho por sequenciar, através da fórmula $(T - D_i) L_i$, em que D_i é a data de entrega, e L_i o peso do trabalho na posição i, ou seja na posição em que se encontra o contador. Se $D_i > T$, então a penalização é zero, isto porque, mais uma vez, estamos só a considerar penalizações por atrasos na entrega;

4. O trabalho a ser sequenciado (colocado na posição que o contador indica) é aquele com a menor penalização calculada no passo 3. Em caso de empate escolhe-se aquele com maior tempo de processamento;

5. Reduzir em 1 a posição do contador, e repetir todos os passos até todos os trabalhos estarem sequenciados.

OF	Numero do Trabalho	Tempo de Processamento, Pi	Data de Entrega, Di	Late Penalty, Li	Atraso	Penalização	-
	1	1	37	49	1	12	44
	2	2	27	36	5	-28	285
	3	3	1	1	1	-64	92
	4	4	28	37	5	-56	280
	Total		93			-148	484

Tabela 16 - Folha B-F após dados importados

Na tabela anterior pode observar-se uma tabela possível de um problema BF (Backward – Forward), cuja única diferença em relação áquelas utilizadas nas heurísticas anteriores, é a inclusão de uma coluna onde é apresentada a penalização calculada no passo 3 da fase *Backward*.

Durante a execução da resolução poderá ainda contar-se com o apoio de uma pequena tabela que contém informação acerca do tempo total de processamento dos trabalhos por sequenciar (T), e a posição do contador (N), para além de mais duas variáveis que apenas serão utilizados na fase *Forward*, e que

T	93
N	4
k	-
OT inicial	-

Tabela 17 - Variáveis na fase Backward

serão então explicados mais à frente. Após a conclusão desta primeira fase a informação *Sequência da Produção* aparecerá com o fundo a amarelo, a tabela terá o seguinte aspecto:

OF	Numero do Trabalho	Tempo de Processamento, Pi	Data de Entrega, Di	Late Penalty, Li	Atraso	Penalização	-
1	2	27	36	5	9	0	-
2	4	28	37	5	-18	90	-
3	3	1	1	1	-55	55	-
4	1	37	49	1	-44	44	-
Total		93			-117	189	

Tabela 18 - Após conclusão da fase *Backward*

3.1.6.2. Forward

A fase *Forward* é executada tendo como base a sequência encontrada anteriormente, que é até agora a melhor sequência. Esta fase recorre a uma heurística de melhoramento, geralmente designada por *2Opt*, que consiste em procurar melhorias numa dada solução trocando sucessivamente a posição entre 2 elementos.

Nesta fase, ao contrário da etapa *Backward*, a resolução avança desde a posição 1 até N, e aqui entra em jogo outra variável, k, que define a distância entre os dois trabalhos da sequência cuja posição se pretende trocar. Os passos que devem ser realizados nesta fase são:

1. Definir $k = N - 1$;
2. Definir j (OF inicial) = 1;
3. Determinar os efeitos da troca entre os trabalhos j e j + k na sequência atual (se $j + k > N$, saltar para o passo 6), isto é, calcular a penalização após troca e compará-la com aquela obtida na melhor sequência conseguida até então, ou seja, com a sequência sem troca;
4. Se houver redução da penalização, ou mesmo se a penalização se mantiver igual, salta-se para o passo 5. Caso contrário, significa que a troca não foi vantajosa, pois originou mais custos (penalização maior), e então a troca é rejeitada. Aumenta-se então em uma unidade o valor de j, e no caso de este continuar igual ou menor a N volta-se ao passo 3. Caso se torne maior, avança-se para o passo 6;
5. Se a penalização decrescer a troca é aceite e realizada. A nova sequência é agora a “melhor”, e volta-se ao passo 1. Mesmo que a penalização se mantenha igual, a troca é realizada e volta-se ao passo 1, a não ser que os trabalhos em causa tenham já sido trocados anteriormente durante a fase *Forward*. Nesse caso a troca não é efectuada. Aumenta-se o valor de j, e caso este seja menor que N volta-se ao passo 3, caso seja igual a N, avança-se para o passo 6;

6. Diminui-se em uma unidade o valor de k . Se $k > 0$, vai-se para o passo 2, se $k = 0$, avança-se para a sétima etapa;

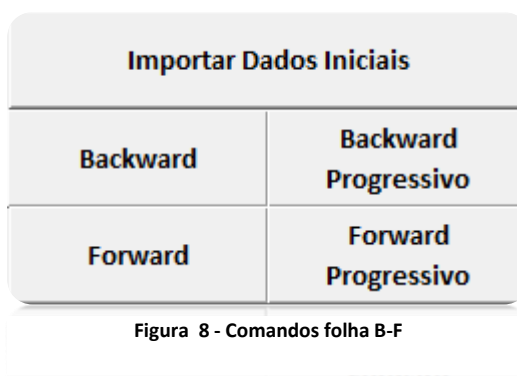
7. A sequência resultante é a melhor sequência gerada por este procedimento.

Os valores de j e k que são acima abordados serão apresentados na tabela que foi já apresentada anteriormente, e tal como acontecia na fase Backward, também durante a realização da fase Forward a sua evolução poderá ser acompanhada em tempo real.

T	-
N	4
k	3
OF inicial	1

Tabela 19 - Variáveis na fase Forward

Mais uma vez apenas será necessário recorrer aos botões para a resolução do exercício.



- **Importar Dados Iniciais** – Importa os dados do problema para a folha *B-F*, onde o problema é resolvido. O *Atraso*, *Penalização*, e $W(T-D_i)L_i$ não são importados, mas sim calculados automaticamente assim que os outros dados estiverem presentes nesta folha;

- **Backward** – realiza a fase Backward procedendo a todas as trocas necessárias até esta estar concluída;

- **Backward Progressivo** – realiza a fase Backward passo-a-passo, ou seja, sequencia um trabalho e espera por nova ordem (novo clique no botão);

- **Forward** - resolve a fase Forward, realizando todas as alterações necessárias até se obter a melhor sequência possível;

- **Forward Progressivo** – resolve a fase Forward passo-a-passo, ou seja, por cada clique no botão compara a troca de dois trabalhos, e se necessário troca-os.

Pode ver-se na tabela seguinte o resultado conseguido após a resolução da fase *Forward*, utilizando como ponto de partida a tabela com a melhor sequência depois de efectuada a fase *Backward*.

OF	Numero do Trabalho	Tempo de Processamento, Pi	Data de Entrega, Di	Late Penalty, Li	Atraso	Penalização	-
1	3	1	1	1	0	0	-
2	2	27	36	5	8	0	-
3	4	28	37	5	-19	95	-
4	1	37	49	1	-44	44	-
Total		93			-63	139	

Sequência da Produção

3 2 4 1

Tabela 20 - Sequenciamento concluído após realizada fase *Forward*

Como se pode observar na Tabela 20, obteve-se uma sequência diferente após a aplicação da fase *Forward*, o que nem sempre acontece, pois por vezes a melhor sequência é logo obtida após a fase *Backward*.

3.1.7. Backward-Forward com Penalizações de Atraso e Antecipação

Até agora as penalizações têm sido geradas apenas por atrasos em relação à data de entrega, no entanto o facto de os trabalhos estarem concluídos antes desta data conduz também a custos, como por exemplo os custos relacionados com stock. Esta regra, que é apenas uma variante da Backward-Forward, vem assim colmatar esta lacuna. As diferenças são poucas para a regra “mãe”. Uma delas é a introdução da variável *Early Penalty, Ei*, e a outra é o facto de existirem agora duas possibilidades para o cálculo auxiliar utilizado na fase *Backward*, e daí a última coluna apresentar agora duas fórmulas.

OF	Numero do Trabalho	Tempo de Processamento, Pi	Data de Entrega, Di	Late Penalty, Li	Early Penalty, Ei	Diferença	Penalização	(T-Di)Li ou (Di-T)Ei
1	1	37	49	1	2	12	24	44
2	2	27	36	5	0	-28	140	285
3	3	1	1	1	3	-64	64	92
4	4	28	37	5	2	-56	280	280
Total		93				-148	508	

Tabela 21 - Folha B-F *Early* após dados importados

A nova variável representa a penalização por dia, a que determinado trabalho estará sujeito, em caso de a sua produção ser concluída antes da data de entrega. Em relação ao cálculo que auxilia a etapa Backward, fica tudo igual quando o trabalho é terminado após o prazo definido para entrega, no entanto quando este é concluído antes da data de entrega é utilizada uma nova fórmula: $(Di - T)Ei$. A nova fórmula utiliza o valor da penalização para trabalhos terminados antes da data limite, em vez da penalização de atraso, e este é agora multiplicado à subtração do tempo de processamento atual à data de entrega do respetivo trabalho, garantindo assim que o resultado apresente um valor positivo.

Tudo o resto é igual, desde os botões de comando, até à apresentação da sequência gerada.

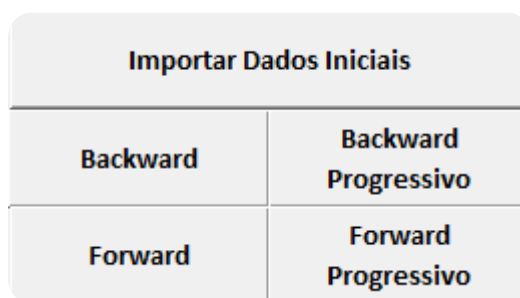


Figura 9 - Comandos folha B-F Early

Neste caso a sequência resultante seria:

OF	Numero do Trabalho	Tempo de Processamento, Pi	Data de Entrega, Di	Late Penalty, Li	Early Penalty, Ei	Diferença	Penalização	$(T-Di)Li$ ou $(Di-T)Ei$	
1	3	1	1	1	3	0	0	-	
2	2	27	36	5	0	8	0	-	
3	4	28	37	5	2	-19	95	-	
4	1	37	49	1	2	-44	44	-	
Total		93				-63	139		
Sequência da Produção		3 2 4 1							

Tabela 22 - Folha B-F Early após resolução

3.2. Problemas Flow-shop

As regras de sequenciamento apresentadas anteriormente dizem respeito a unidades de produção constituídas por apenas uma máquina. No presente capítulo serão

discutidos métodos de sequenciamento para organizações em que a produção é realizada por m máquinas ou postos de trabalho, e em que os trabalhos são processados nestas máquinas pela mesma ordem, ou seja, a sequência de processamento é comum a todas as ordens de fabrico, ver Figura 10.

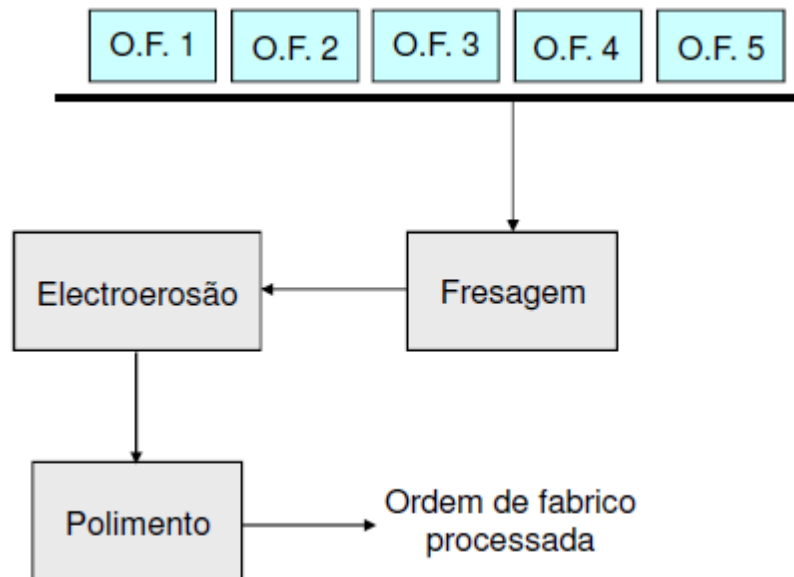


Figura 10 - Esquema de um *Flow-Shop*

O tempo de processamento de cada trabalho em cada máquina pode variar, e assume-se que todos os trabalhos se encontram disponíveis no instante 0. É também assumido que existe espaço físico suficiente entre duas máquinas sucessivas, permitindo assim que cada equipamento liberte o trabalho sem estar preocupado com a lotação, ou inactividade da próxima máquina. Um objetivo muito comum neste tipo de sequenciamento é a minimização do makespan – duração total da sequência (até todas as tarefas estarem processadas). Num problema de flow-shop, um sequenciamento que minimize o makespan, minimiza também outros aspectos importantes, como a soma dos trabalhos em espera, e a soma do tempo de inactividade das máquinas.

Ao contrário do que aconteceu no sub-capítulo *Máquina Única*, neste não será criada uma folha exclusivamente para a inserção de dados, isto porque cada regra pretende encontrar o melhor sequenciamento para unidades de produção com diferentes características, e assim os dados diferem de regra para regra. Desta forma os dados serão inseridos na mesma página em que o problema será resolvido.

3.2.1. Duas Máquinas

Esta é a mais simples disposição de um problema flow-shop, e consiste em apenas duas máquinas nas quais cada trabalho é processado consecutivamente: a primeira operação na primeira máquina, e a segunda operação na segunda máquina. Para o sequenciamento da produção de um chão de fábrica deste tipo será utilizado um método desenvolvido por Johnson (1954), e que ficou conhecido como *regra de Johnson*. Os passos utilizados no algoritmo de Johnson são os seguintes:

1. Nos trabalhos por sequenciar, determina-se o tempo mínimo entre A_i ($= P_{i1}$, tempo de processamento da máquina 1) e B_i ($= P_{i2}$, tempo de processamento da máquina 2); Se o mínimo é o A_i , o trabalho é colocado na primeira posição disponível na sequência. Caso seja o B_i , então o trabalho é colocado na última posição possível na sequência;

2. O trabalho i é marcado como sequenciado;

3. Se todos os trabalhos estão sequenciados avança-se para o passo 4, caso contrário volta-se ao passo 1;

4. A sequência ótima foi encontrada.

Na tabela 23, apresentam-se os dados referentes a 6 OF (ordens de fabrico) que devem ser processadas num chão de fábrica com dois equipamentos, e que serão utilizados para exemplificar o algoritmo de Johnson.

Job	Máquina 1 (P_{i1})	Máquina 2 (P_{i2})
1	15	3
2	8	20
3	18	5
4	25	8
5	17	20
6	22	30

Tabela 23- Dados iniciais do problema Duas-Máquinas

Os dados introduzidos são:

- **Job** – número do trabalho;
- **Máquina 1 (P_{i1})** – tempo de processamento do trabalho i , na máquina 1;

- **Máquina 2 (P_{i2})** – tempo de processamento do trabalho i , na máquina 2.

As cores são escolhidas pelo utilizador, bastando para isso alterar a cor de fundo, na coluna *Job*, do trabalho cuja cor se pretende alterar. As cores que aqui forem escolhidas serão as mesmas que serão utilizadas pelo programa na apresentação da sequência final, tal como se pode observar na Figura 11.

Utilizando os passos da regra de Johnson abordados acima, e os dados da Tabela 23, o sequenciamento obtido iria ser aquele que se pode observar na Figura 11. Como se pode observar nessa mesma figura, em cima encontra-se o sequenciamento da máquina 1, e em baixo o da máquina 2, que são iguais, pois como foi dito anteriormente, os trabalhos são processados nas duas máquinas pela mesma ordem.

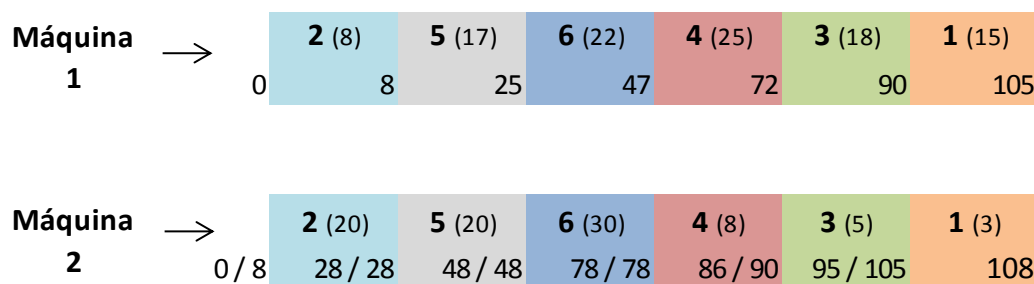


Figura 11- Apresentação da solução num problema Duas-Máquinas

Na Figura 12 apresenta-se esquematicamente a informação que a folha de cálculo desenvolvida apresenta para cada uma da OF sequenciadas.

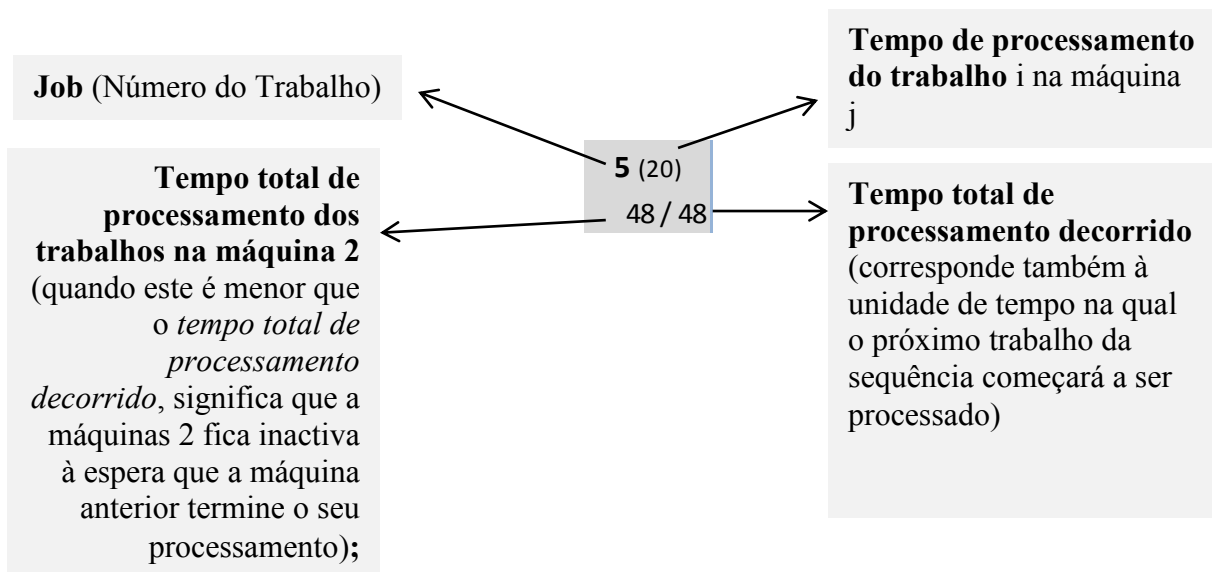


Figura 12 - Apresentação dos vários elementos presentes no esquema da solução

Para resolver um problema nesta folha basta então introduzir na tabela os dados do problema, ou seja, os valores dos tempos de processamento para cada máquina, e de seguida, utilizar os botões de comando seguintes:

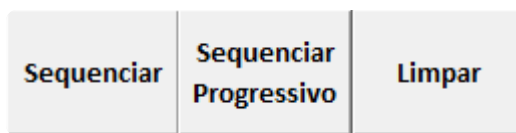


Figura 13 - Comandos presentes na folha *Dois-Máquinas*

- **Sequenciar** – faz, em apenas um clique, o sequenciamento dos trabalhos presentes na tabela, segundo a regra de Johnson;
- **Sequenciar Progressivo** – faz o sequenciamento passo-a-passo, segundo a regra de Johnson. Ou seja, por cada clique, coloca um trabalho no esquema relativo às máquinas, dando-o assim por sequenciado. Deixa de realizar qualquer tarefa quando todos os trabalhos da tabela se encontrarem sequenciados;
- **Limpar** – apaga do esquema das máquinas qualquer trabalho lá presente, e cria os espaços necessários para sequenciar os trabalhos presentes na tabela. Apaga ainda os dados estatísticos que permitem avaliar a sequência, e que serão abordados já de seguida. Pode também ser utilizado sempre que se altera (adiciona ou retira) um trabalho da tabela, pois recalcula o número de espaços necessários no esquema que representa a sequência gerada para cada máquina.

Depois de concluído o sequenciamento serão apresentados na folha alguns dados relativos a este, que permitem assim tirar algumas conclusões em relação à sua eficiência. Estes dados são o *Makespan*, *Tempo Inactivo da Máquina 2*, e *Tempo em Espera Pela Máquina 2*.

Makespan	Tempo Inactivo	Tempo de espera pela...
108	Máquina 2 14	Máquina 2 10

Tabela 24 - Apresentação de alguns dados relativos à solução

Ao observar-se os dados relativos ao problema que tem servido de exemplo, pode concluir-se que:

- o processamento de todos os trabalhos nas duas máquinas tem a duração de 108 unidades de tempo;

- durante todo o processo a máquina 2 fica 14 unidades de tempo inactiva, o que significa que depois de acabar o seu trabalho esta fica à espera que a máquina 1 acabe aquele que está a realizar;
- o somatório de unidades de tempo em que a máquina 1 acaba a sua ordem de fabrico antes da máquina 2 é 10.

3.2.2. Três Máquinas

Apesar de a regra de Johnson não poder ser utilizada para um problema de n máquinas, esta pode ser extendida aos problemas de três máquinas. No entanto a utilização desta regra num problema deste tipo só é válida se a máquina nº 2 não constituir um estrangulamento para o sistema. Para isso ela não pode atrasar nenhum trabalho, isto é, assim que um trabalho é libertado pela máquina 1, a máquina 2 tem que estar pronta a processá-lo de imediato. Se esse fôr o caso, então a regra de Johnson pode ser aplicada, e o após um simples cálculo dos valores de A_i e B_i o procedimento é igual ao aplicado nos problemas de duas máquinas. Assim: $A_i = P_{i1} + P_{i2}$ e $B_i = P_{i2} + P_{i3}$, e a tabela original do problema, Tabela 26, dá origem à Tabela 25.

Job	Máquina 1 (P_{i1})	Máquina 2 (P_{i2})	Máquina 3 (P_{i3})		$A_i = P_{i1} + P_{i2}$	$B_i = P_{i2} + P_{i3}$
1	15	2	1	→	17	3
2	8	6	14		14	20
3	18	2	3		20	5
4	25	5	5		30	10
5	17	10	10		27	20
6	22	10	20		32	30

Tabela 26 - Tabela onde são inseridos os dados iniciais

Tabela 25 - Tabela resultante da transformação da anterior

Usando a tabela resultante e o procedimento discutido no capítulo anterior facilmente se chega à melhor sequência.

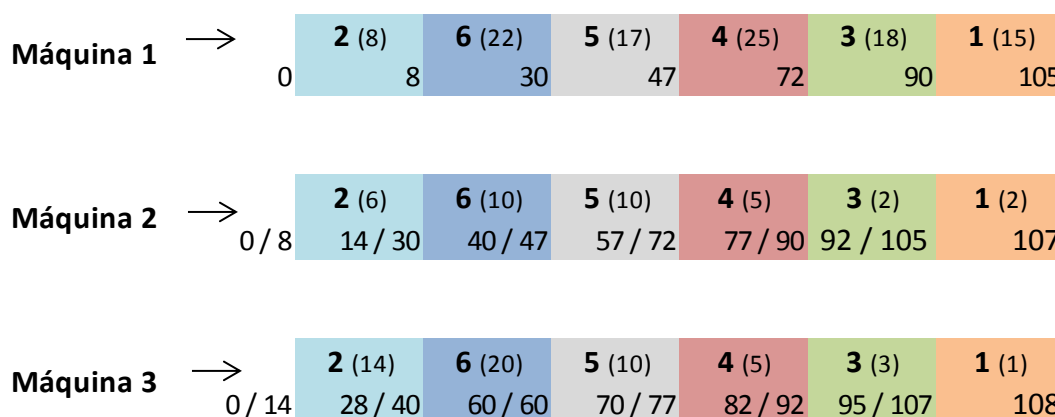


Figura 14 - Apresentação da solução para um problema *Três Máquinas*

A representação da sequência é também ela idêntica, e o significado dos vários valores em cada posição é o mesmo, sendo que aqui a única diferença prende-se com o fato de neste problema serem apresentadas três máquinas. Os botões de comando têm exatamente a mesma função.

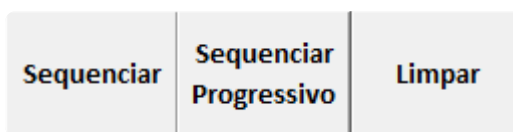


Figura 15 - Comandos utilizados na folha *Três Máquinas*

As últimas diferenças podem ser encontradas na apresentação dos dados relativos a tempos de espera e de inatividade das máquinas.

Tempo Inactivo		Tempo de espera pela...	
<u>Máquina 2</u>	64	<u>Máquina 2</u>	0
<u>Máquina 3</u>	41	<u>Máquina 3</u>	3
Makespan	108		

Figura 16 - Esquema de apresentação de alguns dados relativos à solução encontrada

Os valores para a nova máquina têm a mesma definição que aqueles apresentados na regra anterior, com a diferença de que estes são relativos à máquina 2, com o tempo que a faz esperar e com o tempo que está inativa à espera dela. Os dados apresentados para a máquina 2 continuam a estar relacionados com a máquina 1. No entanto, como a máquina 2, e mais precisamente o tempo de espera por esta, é um dado

importante para a validação, ou não validação do sequenciamento, há também aqui uma diferença. Assim, sempre que a máquina 2 não constitui um estrangulamento, ou seja, o tempo de espera pela máquina 2 é igual a 0, o fundo da célula relativa a este valor fica verde, atestando assim a validade de todo o sequenciamento. Caso isto não aconteça, isto é, caso a máquina 2 constitua um estrangulamento do sistema, então o fundo dessa mesma célula ficará vermelho, significando assim que a sequência da produção gerada para o problema em questão não pode ser realizada através da regra de Johnson. Relativamente ao *makespan* continua tudo igual.

3.2.3. n - Trabalhos m - Máquinas

Até agora foram estudadas ferramentas para um chão de fábrica com duas ou três máquinas, sendo que esta última é praticamente uma extensão da primeira. Neste subcapítulo será analisado um método heurístico que permite alargar este estudo a um chão de fábrica composto por m máquinas, ou seja, por 4 ou mais máquinas. Tal como anteriormente, como se trata de um método heurístico não é garantida a solução ótima, mas ainda assim estão garantidos bom resultados, muito próximos da solução ótima.

Sabendo que existem vários métodos para resolver este tipo de problemas, foi escolhido aquele utilizado nas aulas de Gestão da Produção, nas quais se pretende utilizar as folhas de cálculo elaboradas neste estudo. Assim, o método escolhido foi o *Minimizar o tempo em inatividade da máquina*, sendo que o nome ilustra bem o princípio básico deste. A explicação desta regra torna-se mais fácil através da exemplificação da resolução de um problema (ainda que não na sua totalidade), e para este efeito será utilizado um problema com 6 trabalhos e 4 máquinas cujos valores se encontram na Tabela 28. Na Tabela 27 pode observar-se o local onde são introduzidos e alterados os valores das variáveis *Número de Trabalhos* e *Número de Máquinas*.

Número de Trabalhos	6
Número de Máquinas	4

Tabela 27 - Local onde são definidos os números de trabalhos e máquinas do problema

Trabalhos	Máquina				ΣP_{ij}
	1	2	3	4	
1	25	45	52	40	162
2	7	41	22	66	136
3	41	55	33	21	150
4	74	12	24	48	158
5	7	15	72	52	146
6	12	14	22	32	80

Tabela 28 - Tabela onde são inseridos os dados iniciais

Na primeira coluna encontram-se os números dos trabalhos, na última, a soma dos tempos de processamento de um trabalho em todas as máquinas (calculado automaticamente pela folha de cálculo), e nas restantes o tempo de processamento de um trabalho na máquina a que corresponde a coluna. Este método explica-se então em 5 passos:

1. Calcula-se a soma dos tempos de processamento de cada trabalho. O trabalho com menor soma é sequenciado na primeira posição, e são calculados os tempos em que a sua produção acaba em cada máquina;

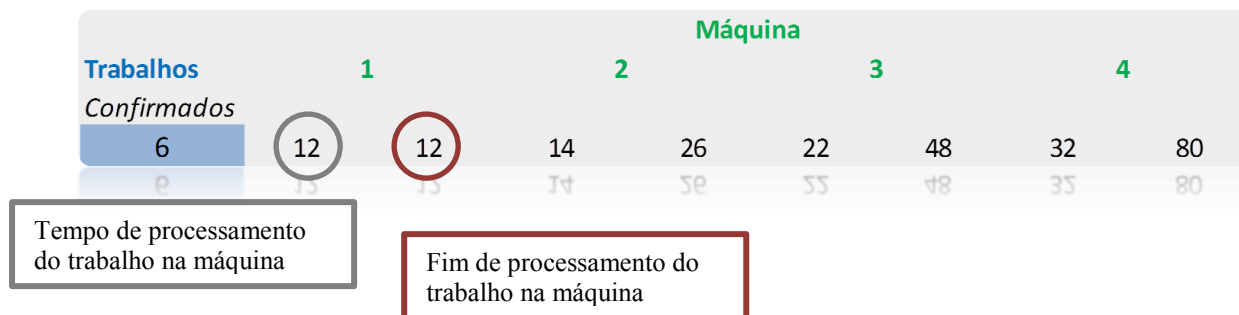


Figura 17 - Primeiro trabalho a ser sequenciado, e esquema de apresentação

2. Seleccionam-se os 3 trabalhos (se ainda disponíveis) com menores soma dos tempos de processamento, e estes são colocados “em teste” numa tabela de iteração. Os tempos em que, em cada máquina, as suas produções acabariam caso fossem sequenciados na próxima posição disponível, são calculados e apresentados na tabela de iteração. Esses valores podem ser observados na Tabela 29;

Trabalhos	Máquina								Total Machine Idle Time	
	1	2	3	4	5	6	7	8		
Confirmados										
6	12	12	14	26	22	48	32	80		
Em teste										
2	7	19	41	67	22	89	66	155	28	
5	7	19	15	41	72	120	52	172	40	
3	41	53	55	108	33	141	21	162	148	

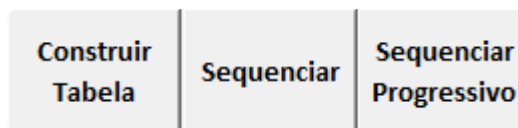
Tabela 29 - Trabalhos em teste, com vista a ocupar a próxima posição da sequência

3. Calcula-se o tempo total de inatividade das máquinas para cada trabalho. Existe inatividade de uma máquina quando o tempo de conclusão do trabalho que está a ser produzido na máquina imediatamente anterior (linha i , coluna $k - 1$) é maior que o tempo em que a máquina em questão (coluna k) fica livre, ou seja, em que o último trabalho que foi sequenciado termina a sua produção nessa máquina. O tempo de inatividade é a diferença entre estes dois valores. Na última coluna da tabela de resolução acima encontra-se o total dos tempos de inatividade das máquinas na produção de cada trabalho;

4. O trabalho a ser sequenciado de seguida é aquele com menor valor no total de tempo de inatividade das máquinas em teste. Em caso de empate devem ser testados sequenciamentos com cada um dos trabalhos empatados a ser sequenciado na próxima posição. Assim, aquele que for sequenciado em primeiro lugar numa primeira hipótese, irá ser segunda opção na hipótese seguinte;

5. Caso todos os trabalhos estejam sequenciados, para-se. Caso contrário, calcula-se o tempo de conclusão do trabalho que se acrescentou à sequência no passo 4, e volta-se ao passo 2.

No capítulo dos botões existe agora uma novidade, para além dos já habituais *Sequenciar*, e *Sequenciar Progressivo*.

Figura 18 - Comandos utilizados na folha n - Trabalhos m - Máquinas

- **Construir Tabela** – Este comando tem como intuito, em primeiro lugar, apagar a tabela na qual foi criada a solução de um problema resolvido anteriormente, e em segundo, criar a tabela que será utilizada para resolver o novo problema, ou tentar nova resolução do mesmo problema. Esta é criada segundo os valores escolhidos para números

de trabalho e de máquinas. Só após a introdução destes se deve utilizar este botão, pois ele definirá a dimensão da tabela de introdução dos dados. Depois sim podem ser inseridos/alterados os dados do problema (estes são apagados manualmente pelo utilizador, e não através de comando), e utilizados os botões que visam sequenciar os trabalhos;

- **Sequenciar** – Este botão pode ser utilizado, para de uma só vez, e sempre respeitando as regras deste método heurístico, resolver este problema de sequenciamento;

- **Sequenciar Progressivo** – Cada vez que se utiliza este comando, um trabalho é sequenciado, e três (se ainda disponíveis) são colocados na tabela de interação “em teste”. Sempre que se verificar um empate entre dois tempos de inatividade durante a resolução do problema, a tabela de sequenciamento atual será copiada para outra zona da folha de cálculo, para que após a conclusão do sequenciamento que está a ser realizado, se possa testar na nova tabela um novo sequenciamento, no qual se sequenciará em primeiro lugar o trabalho empatado, ao qual não foi dada preferência na resolução anterior. (Esta situação acontecerá também aquando da utilização do botão *Sequenciar*, e ambos os casos, cada vez que um empate se verificar)

No caso do problema que tem sido usado como exemplo são geradas três sequências, isto porque são encontrados dois empates ao longo da resolução:

Em teste									
3	41	67	55	137	33	194	21	234	0
4	74	100	12	112	24	185	48	261	18
1	25	51	45	127	52	213	40	253	0

Tabela 30 - Primeiro empate

O primeiro destes empates é aquele que se observa na Tabela 30, onde, como se pode ver, os trabalhos número 3 e número 1 têm o mesmo tempo total de inatividade. Isto só é um problema porque este tempo é o mais reduzido de entre todos os trabalhos em teste. Caso estes tivessem empatados com um tempo total de inatividade igual a 22, por exemplo, o trabalho número 4 seria o escolhido para ser sequenciado de seguida, e seriam escolhidos três novos trabalhos para serem testados. Em casos como este opta-se então aleatoriamente por sequenciar na próxima posição um dos trabalhos em situação de empate (neste optou-se pelo trabalho nº3) e leva-se a resolução até ao fim, testando-se de seguida a sequência com o outro trabalho (trabalho nº1) a ser sequenciado de seguida.

O segundo empate surge na segunda hipótese de sequência, entre os trabalhos número 3 e número 4, após se escolher, o trabalho nº1 para ser sequenciado de seguida. Este facto obriga então à realização de uma terceira hipótese. As três sequências geradas são apresentadas na Figura 19.

Trabalhos	Máquina							
	1	2	3	4	5	6	7	8
<i>Confirmados</i>								
6	12	12	14	26	22	48	32	80
2	7	19	41	67	22	89	66	155
5	7	26	15	82	72	161	52	213
3	41	67	55	137	33	194	21	234
4	74	141	12	153	24	218	48	282
1	25	166	45	211	52	270	40	322
<i>Confirmados</i>								
6	12	12	14	26	22	48	32	80
2	7	19	41	67	22	89	66	155
5	7	26	15	82	72	161	52	213
1	25	51	45	127	52	213	40	253
3	41	92	55	182	33	246	21	274
4	74	166	12	194	24	270	48	322
<i>Confirmados</i>								
6	12	12	14	26	22	48	32	80
2	7	19	41	67	22	89	66	155
5	7	26	15	82	72	161	52	213
1	25	51	45	127	52	213	40	253
4	74	125	12	139	24	237	48	301
3	41	166	55	221	33	270	21	322

Figura 19 - Sequências geradas duante a resolução

O makespan de cada uma das sequências pode ser observado na última coluna de cada trabalho, que é o mesmo que dizer, quando a última máquina conclui o processamento do último trabalho da sequência. Na Figura 19 estes valores encontram-se sublinhados a verde, isto porque, curiosamente, as três sequências têm o mesmo makespan, o que faz com que a todas façam parte da solução deste problema. O sublinhamento do(s) makespan(s) mais reduzido(s) é uma das formas de apresentação dos resultados obtidos com esta ferramenta: a sequência cujo makespan se encontrar sublinhado faz parte da solução do problema. Estas são ainda

Sequência da Produção	Makespan
-6 - 2 - 5 - 3 - 4 - 1	322
-6 - 2 - 5 - 1 - 3 - 4	
-6 - 2 - 5 - 1 - 4 - 3	

Figura 20 - Apresentação dos resultados

apresentadas numa tabela, como se pode ver na Figura 20, onde se pode observar também o makespan originado pela solução encontrada.

3.3. Máquinas Paralelas

Até este momento foram apenas apresentadas ferramentas para resolver problemas em que apenas existia uma máquina ou instalação onde é realizado determinado processo, ou seja, para cada tarefa a realizar havia apenas uma possibilidade de equipamento. No entanto existem casos em que existe mais do que uma máquina capaz de realizar os mesmos serviços. Tal como existem situações em que não há só uma instalação capaz de realizar um determinado trabalho, mas sim várias, idênticas e com capacidade para realizar o mesmo tipo de processos. A este tipo de configuração dá-se o nome de processamento paralelo, e neste sub-capítulo estudar-se-á uma ferramenta capaz de realizar o sequenciamento para este tipo de chão-de-fábrica, e com isso tirar partido da sua grande vantagem, que é o facto de conseguir reduções consideráveis no makespan.

Existem para este tipo de configurações, tal como acontece nos capítulos anteriores, várias opções de heurísticas, e seguindo o critério de decisão utilizado anteriormente, foi escolhido o método ao qual se deu maior ênfase nas aulas de Gestão da Produção, neste caso o: *Trabalhos sem datas de entrega*. Neste tipo de trabalhos a única variável relevante é o tempo de processamento, e assim sendo o único objectivo é minimizar o makespan. De seguida são apresentados os passos necessários para, utilizando esta ferramenta, resolver um problema deste tipo.

1. Ordenar os trabalhos por ordem decrescente do seu tempo de processamento;

2. Calcular o limite inferior para o makespan. Este é calculado através da somatório de todos os tempos de processamento, dividido pelo número de máquinas;

3. Alocar os trabalhos às máquinas, começando por aquele que tem o maior tempo de processamento (em caso de empate, escolher aleatoriamente), ou seja, seguindo a ordem obtida no passo 1, até que uma das seguintes situações suceda:

a. A soma dos tempos de processamento dos trabalhos alocados a uma máquina iguale o limite calculado no passo 2. Se tal acontecer começa-se a alocar os trabalhos na próxima máquina disponível.

b. A soma dos tempos de processamento alocados à máquina em consideração ultrapassa o limite obtido no passo 2. Neste caso, o trabalho “culpado” por

esta situação é retirado da máquina em questão e alocado numa máquina onde a sua adição não implique a violação do limite inferior. Os trabalhos continuam nesta nova máquina. Se a adição de um trabalho fizer com que o limite inferior seja ultrapassado em todas as máquinas, este deve ser alocado à máquina em que após a adição deste o tempo total de processamento fique mais próximo do limite inferior.

Assim que todos os trabalhos estiverem alocados a uma máquina o sequenciamento está concluído e o makespan é o maior dos tempos de processamento das várias máquinas.

A Tabela 31 representa uma tabela típica de um problema deste tipo, e será utilizada para se ilustrar como foi desenvolvida a ferramenta na folha de cálculo, que através da utilização dos passos referidos anteriormente, permite atingir uma solução que nos garante um makespan mínimo.

Tabalho	1	2	3	4	5	6	7	8	9
Tempo de Processamento	10	12	5	8	7	3	5	15	12

Tabela 31 - Tabela onde são inseridos os dados iniciais de um problema Máquinas Paralelas

Esta tabela é composta por nove trabalhos, e respetivos tempos de processamento. Os valores dos tempos de processamento são utilizados meramente como exemplo, e o utilizador na sua folha de cálculo tem total liberdade para os alterar, e adicionar novos trabalhos. Como indicado anteriormente, o 1º passo é ordenar os trabalhos por ordem decrescente dos seus tempos de processamento. A tabela ordenada é apresentada abaixo.

Tabalho	8	2	9	1	4	5	3	7	6
Tempo de Processamento	15	12	12	10	8	7	5	5	3

Tabela 32 - Tabela já com os Tempos de Processamento ordenados de forma decrescente

O número de máquinas pode variar num problema deste tipo, e neste caso serão utilizadas 3. Esse valor será apresentado como indica a Tabela 33, e poderá também ele ser alterado pelo utilizador. A tabela onde será apresentada a resolução tem a seguinte configuração:

Número de Máquinas	3
---------------------------	---

Tabela 34 - Local onde se insere o número de máquinas do problema

Máquina 1			Máquina 2			Máquina 3		
Trabalho	T. Proc.	RCT = 26	Trabalho	T. Proc.	RCT = 26	Trabalho	T. Proc.	RCT = 26

Tabela 33 - Esquema da tabela onde será resolvido o problema

Cada máquina é composta por três colunas: a primeira serve para identificar o número do trabalho, a segunda o seu respetivo tempo de processamento, e a terceira para além de mostrar no seu título o valor do limite inferior, calculado como indica o passo 2, tem ainda como intuito mostrar o *Remaining Cumulative Time (RCT)*, ou seja, o tempo de processamento que pode ainda ser alocado à respetiva máquina antes do limite ser atingido.

Máquina 1			Máquina 2			Máquina 3		
Trabalho	T. Proc.	RCT = 26	Trabalho	T. Proc.	RCT = 26	Trabalho	T. Proc.	RCT = 26
8	15	11						

Tabela 35 - Tabela de resolução já com o primeiro trabalho sequenciado

Como pode ser observado na tabela acima apresentada, ao ser alocado o primeiro trabalho o RCT da máquina 1 toma o valor de 11, isto porque tinha antes 26 unidades de tempo disponíveis para processamento, e o trabalho 8 ocupou-lhe 15 dessas unidades.

No fim, com o sequenciamento concluído a tabela terá o seguinte aspeto:

Máquina 1			Máquina 2			Máquina 3		
Trabalho	T. Proc.	RCT = 26	Trabalho	T. Proc.	RCT = 26	Trabalho	T. Proc.	RCT = 26
8	15	11	2	12	14	4	8	18
1	10	1	9	12	2	5	7	11
			6	3	-1	3	5	6
						7	5	1

Tabela 36 - Tabela já com o sequenciamento concluído

Pode observar-se que logo no 2º trabalho sequenciado se encontrou uma das situações descritas no 3º passo, em que o tempo de processamento do trabalho em questão, o trabalho nº2 (12), ultrapassava o RCT da primeira máquina (11), e por isso se passou para a máquina seguinte, onde este foi alocado pois a máquina em questão estava disponível (tinha RCT suficiente). Há ainda outro aspeto interessante nesta resolução, que tem que ver com a alocação do último trabalho (nº6). Nenhuma das máquinas tinha RCT disponível para o acolher, e por isso foi alocado na máquina que ainda assim tinha RCT maior, garantindo desta forma que a máquina à qual foi afectado ficou mais próxima do limite inferior, do que qualquer outra onde este tivesse sido colocado.



Figura 21 - Comandos utilizados na folha *Máquinas Paralelas*

Seguindo o modelo utilizado em ferramentas anteriores, serão utilizados comandos para realizar o sequenciamento e outras ações relacionadas com a resolução deste. Assim, nesta folha teremos os seguintes:

- **Limpar** – Este comando elimina a tabela onde é resolvido o problema, e os dados presentes na tabela ordenada. Deixa assim oportunidade de se usarem os mesmos dados para voltar a resolver o problema, ou fazê-lo alterando pequenas coisas, como o número de máquinas, o número de trabalhos, ou os tempos de processamento de alguns deles. Faz assim mais sentido estes dados serem alterados/eliminados pelo utilizador, oferecendo assim a oportunidade de este o fazer quando bem entender. Não há portanto comandos para esse efeito.

- **Construir Tabela** – Este comando deve ser utilizado após o utilizador ter definido qual o número de trabalhos e seus tempos de processamento, e também o número de máquinas paralelas. Após ter inserido estes valores, este comando ordenará os trabalhos na segunda tabela por ordem decrescente do tempo de processamento, e desenhará a tabela onde será construída a solução do problema.

- **Sequenciar** – após utilizar os comandos anteriores, ou no caso de ser a primeira resolução, apenas o *Construir Tabela*, clicar no botão *Sequenciar* fará com que o programa corra, e pare apenas quando estiver concluído o sequenciamento, ou seja, quando se obtiver a Tabela 36;

- **Sequenciar Progressivo** – mais uma vez, e com o intuito de tornar perceptível a resolução, ou pelo menos possível de ser acompanhada, este comando permite que se sequencie um trabalho de cada vez. Assim após a sua primeira utilização surgiria a Tabela35, na segunda juntar-se a este, na máquina 2, o trabalho nº2, e por aí adiante até se obter a tabela final.

Sempre que se atinge a conclusão de um sequenciamento, quer este seja realizado trabalho a trabalho, ou de uma só vez, será apresentado, como mostrado na Figura 22, o makespan gerado pela solução alcançada.

Makespan
27

Figura 22 - Makespan da solução alcançada

Este não é nada menos que o maior tempo de processamento de entre todas as máquinas, sendo que no caso do problema utilizado nesta explicação, o makespan iria ser 27, valor este gerado pela máquina 2, cujo tempo de processamento ultrapassa em uma unidade o limite inferior (RCT) que era de 26.

4. CONCLUSÕES

Desde os algoritmos mais simples até aos mais complexos, os objectivos foram cumpridos. Ao utilizar as ferramentas agora construídas nas folhas de cálculo fica claro que resolver problemas de sequenciamento através destas é um processo muito menos moroso do que o fazer manualmente. Além disto, também a fácil compreensão e melhor acompanhamento dos vários passos da resolução dos problemas foi atingido, através da inclusão de um comando *Sequenciar Progressivo*, que permite ao utilizador realizar o sequenciamento ao ritmo que bem entender, e melhor se adaptar a si. No entanto, para dar ainda uma ajuda extra na compreensão, e porque por vezes há passos mais difíceis de compreender, parece-me importante incluir um comando que permita retroceder no processo de resolução, e assim percorrer as várias etapas já realizadas. Este comando não foi incluído pois desde início foi estabelecido que a prioridade seria programar as ferramentas, de forma a que elas cumprissem os objectivos a que se propunham, e no fim, caso houvesse tempo, seriam pensados alguns extras com vista a melhorar pequenos aspetos. Ainda assim, como se pode observar ao utilizar as folhas de cálculo construídas, existem alguns pormenores que foram incluídos e que se podem considerar extras, como por exemplo o comando *Limpar*, presente em algumas das folhas, que é uma tarefa que se poderia deixar perfeitamente à responsabilidade do utilizador. No entanto, como eram de relativamente fácil programação, e traziam claras melhorias ao nível da intuitividade do programa, foram incluídas logo na fase inicial.

Para além de se propôr ajudar na compreensão das várias regras de sequenciamento, há ainda a possibilidade de se utilizar a ferramenta construída para consulta da solução de um determinado problema de sequenciamento, quer a nível académico como a nível profissional, numa pequena organização cuja configuração de chão-de-fábrica tenha sido estudada neste trabalho, e onde a tarefa de sequenciamento seja ainda realizada manualmente. Para este efeito foi disponibilizado um comando *Sequenciar* que realiza, em todas as regras estudadas, o sequenciamento de uma só vez, ou seja, desde a tabela inicial até à solução final.

À medida que, dentro de cada sub-capítulo, se avança nas várias regras, é fácil notar que o tempo que estas levam a chegar à solução final vai aumentando, não passando

no entanto da casa da uma dezena de segundos. Este fato deve-se ao aumento da complexidade dos algoritmos, e por consequência à necessidade de um número maior de cálculos e iterações. Este fenómeno também se verifica quando estes algoritmos se resolvem manualmente. Teria sido também interessante ter realizado um estudo comparativo entre os tempos de resolução de vários problemas, manual e computacionalmente. Sendo certo que a resolução destes com a ajuda de um programa de computador é mais rápida, podia ser interessante perceber a diferença de tempos entre estes, ou até mesmo verificar se é proporcional o aumento de tempo que se regista com a maior complexidade dos problemas na resolução manual e computacional. Esta fica também como sugestão de trabalho futuro. Pode ainda considerar-se realizar um estudo em conjunto com uma ou várias organizações que não possuam ainda este tipo de ferramentas, e que vise o desenvolvimento de folhas deste tipo, à medida de cada uma das organizações.

O facto de o tempo ser limitado e de haver um grande número de ferramentas, devido à possibilidade de inclusão de novas variáveis (tempo de setup, produção por lotes, mão de obra limitada) nos algoritmos mais simples, com vista a aproximá-los da realidade das organizações, fez com que fosse impossível abordá-las todas. daquelas que são leccionadas durante a unidade curricular Gestão da Produção, ficaram no entanto apenas por estudar as ferramentas para configurações job-shop. Assim seria interessante futuramente adicioná-las à folha de cálculo que foi elaborada neste trabalho, e até mesmo completá-la com outras regras que surgem naturalmente com o aumento da complexidade das várias configurações de chão-de-fábrica, tornando-a assim mais suscetível de ser utilizada a nível industrial.

Outro constrangimento foi o facto de as aulas de Gestão da Produção, no Departamento de Engenharia Mecânica da Universidade de Coimbra, local onde foi realizado o estudo, não terem coincidido com o semestre de elaboração deste. Teria sido interessante ter aliado o desenvolvimento das ferramentas na folha de cálculo ao feedback dos alunos após as testarem nas aulas. Sendo certo no entanto que estas serão utilizadas no semestre que se avizinha, e caso não hajam alunos interessados em dar continuidade a este trabalho, desde já me disponibilizo, tanto para resolver algum erro proveniente da

programação, como para adicionar às ferramentas algum aspeto cuja inclusão se mostre pertinente.

5. BIBLIOGRAFIA

- Dileep R. Sule (1997): “Industrial Scheduling” – PWS Publishing Company
- Blazewics, J.; Ecker, K.H.; Pesch, E.; Schmidt, G.S.; Weglarz, J. (1996): “Scheduling computer and manufacturing processes”, Springer-Verlag.
- Baker, K. R. (1974): “Introduction to Sequencing and Scheduling” – John Wiley & Sons
- Brucker P. (1995): “Scheduling Algorithms” – Springer-Verlag
- Blazewics, J.; Lenstra, J.; Rinnooy Kan, A. (1993): “Scheduling subject to resource constraints: classification and complexity” – Discrete Applied Mathematics, Volume 5, Issue 1

ANEXO A – CÓDIGO MÁQUINA ÚNICA

```

Public jobs_3(1 To 100, 1 To 6) As Integer
Public linha_inicio As Integer
Public num_jobs As Integer
Public ultimo As Integer
Public pos(1 To 100) As Integer
Public ocupado(1 To 100) As Integer
Public row1 As Integer "maquina 1 indice linha
Public col As Integer "maquina 1 e 2 indice coluna
Public row2 As Integer "maquina 2 indice linha
Public row3 As Integer "maquina 3 indice linha

Public Function calcular_ultimo(l) As Integer
    ultimo = 1
    contador = 0
    Do Until ultimo = 0
        If Application.Workbooks("Sequenciamento da
        Producao.xlsm").Worksheets("Duas Máquinas").Cells(l,
        2).Value = Empty Then
            ultimo = 0
        Else
            ultimo = 1
            contador = contador + 1
            l = l + 1
        End If
    Loop
    calcular_ultimo = contador
End Function

Function Create_Spot(linha, coluna) As Integer
    ref = 8

    With ActiveSheet.Range(Cells(linha, coluna), Cells(linha +
    1, coluna))
        .Borders(xlEdgeLeft).LineStyle = xlContinuous
        .Borders(xlEdgeRight).LineStyle = xlContinuous
        .Borders(xlEdgeBottom).LineStyle = xlContinuous
        .Borders(xlEdgeTop).LineStyle = xlContinuous
        .HorizontalAlignment = xlCenter
        .VerticalAlignment = xlCenter
        .WrapText = True
        .Orientation = 0
        .AddIndent = False
        .IndentLevel = 0
        .ShrinkToFit = False
        .ReadingOrder = xlContext
        .MergeCells = True
    End With

    Create_Spot = coluna - ref

End Function

Sub Seq_3_maq()

Application.Workbooks("Sequenciamento da
Producao.xlsm").Worksheets("Três Máquinas").Activate

Dim i As Integer
Dim j As Integer
Dim num_spot As Integer
Dim menor As Integer
Dim maior As Integer
Dim temp_maq1 As Integer
Dim temp_maq2(1 To 3) As Integer
Dim temp_maq3(1 To 3) As Integer
Dim maximo3(1 To 100) As Integer
Dim maximo2(1 To 100) As Integer

Dim idle_maq2 As Integer
Dim idle_maq3 As Integer
Dim wait_maq2 As Integer
Dim wait_maq3 As Integer

col = 9
row1 = 14
row2 = 18
row3 = 22

idle_maq2 = 0
idle_maq3 = 0
wait_maq2 = 0
wait_maq3 = 0

linha_inicio = 4
num_jobs = calcular_ultimo(linha_inicio)
ultimo = 3 + num_elementos

i = 1
j = 1

Do Until i = num_jobs + 1
    Do Until j = 6 + 1
        If j <= 4 Then
            jobs_3(i, j) = Cells(i + 3, j + 1).Text
            j = j + 1
        Else
            jobs_3(i, j) = jobs_3(i, j - 3) + jobs_3(i, j - 2)
            Cells(i + 3, j + 1) = jobs_3(i, j)
            j = j + 1
        End If
    Loop
    j = 1
    pos(i) = 0
    ocupado(i) = 0
    i = i + 1
Loop

"MsgBox (jobs_2(4, 3))

Do Until num_spot = num_jobs
    num_spot = Create_Spot(row1, col)
    num_spot = Create_Spot(row2, col)
    num_spot = Create_Spot(row3, col)
    "MsgBox (num_spot)
    col = col + 1
Loop

col = 9

Cells(row1 + 1, col - 1).Value = 0
Cells(row1 + 1, col - 1).HorizontalAlignment = xlRight
Cells(row1 + 1, col - 1).Font.Size = 11

Do Until sequenciados = num_jobs

    i = 1
    j = 5

    Do Until i = num_jobs + 1
        Do Until j = 6 + 1
            If i = 1 And j = 5 Then
                maior = jobs_3(i, j)
                menor = jobs_3(i, j)
                menor_ind = i
                menor_maq = j - 4
            Elseif jobs_3(i, j) < menor Then

```

```

        menor = jobs_3(i, j)
        menor_ind = i
        menor_maq = j - 4
        ElseIf jobs_3(i, j) > maior Then
            maior = jobs_3(i, j)
        End If
        j = j + 1
    Loop
    j = 5
    i = i + 1
Loop

sequenciados = Copytomachine_3(menor, menor_ind,
menor_maq)
If sequenciados = 1 Then
    maior_inicial = maior
End If
jobs_3(menor_ind, 5) = menor + maior_inicial + 1
jobs_3(menor_ind, 6) = menor + maior_inicial + 1
"MsgBox (sequenciados)

Loop

i = 1
Do Until i = 4
    temp_maq1 = 0
    temp_maq2(i) = 0
    temp_maq3(i) = 0
    i = i + 1
Loop

i = 1

col = 9
row1 = 14
row2 = 18
row3 = 22

Do Until i = num_jobs + 3
    If i <= num_jobs Then
        temp_maq1 = temp_maq1 + Cells(3 + pos(i), 3)
        Cells(row1 + 1, col - 1 + i).Value = temp_maq1
        Cells(row1 + 1, col - 1 + i).HorizontalAlignment =
xlRight
        Cells(row1 + 1, col - 1 + i).Font.Size = 11

        If i = 1 Then
            temp_maq2(1) = 0
            temp_maq2(2) = Cells(3 + pos(1), 3)
            temp_maq2(3) = temp_maq2(2)
            Cells(row2 + 1, col - 2 + i).Value = temp_maq2(3)
            maximo2(i) = temp_maq2(3)
        Else
            temp_maq2(1) = temp_maq1
            temp_maq2(2) = maximo2(i - 1) + Cells(3 + pos(i - 1),
4)

            If temp_maq2(1) >= temp_maq2(2) Then
                temp_maq2(3) = temp_maq2(1)
                maximo2(i) = temp_maq2(3)
                Cells(row2 + 1, col - 2 + i).Value = temp_maq2(3)
            Else
                temp_maq2(3) = temp_maq2(2)
                maximo2(i) = temp_maq2(3)
                Cells(row2 + 1, col - 2 + i).Value = temp_maq2(3)
            End If
            Cells(row2 + 1, col - 2 + i).HorizontalAlignment =
xlRight
            Cells(row2 + 1, col - 2 + i).Font.Size = 11
        End If

        If i = 1 Then

```

```

            temp_maq3(1) = 0
            temp_maq3(2) = temp_maq2(3) + Cells(3 + pos(1), 4)
            temp_maq3(3) = temp_maq3(2)
            Cells(row3 + 1, col - 2 + i).Value = temp_maq3(3)
            maximo3(1) = temp_maq3(3)
        Else
            temp_maq3(1) = temp_maq2(3) + Cells(3 + pos(i), 4)
            temp_maq3(2) = maximo3(i - 1) + Cells(3 + pos(i - 1),
5)

            If temp_maq3(1) >= temp_maq3(2) Then
                temp_maq3(3) = temp_maq3(1)
                maximo3(i) = temp_maq3(3)
                Cells(row3 + 1, col - 2 + i).Value = temp_maq3(3)
            Else
                temp_maq3(3) = temp_maq3(2)
                maximo3(i) = temp_maq3(3)
                Cells(row3 + 1, col - 2 + i).Value = temp_maq3(3)
            End If
        End If
        Cells(row3 + 1, col - 2 + i).HorizontalAlignment =
xlRight
        Cells(row3 + 1, col - 2 + i).Font.Size = 11

        If i > 1 Then
            If temp_maq2(1) >= temp_maq2(2) Then
                idle_maq2 = idle_maq2 + temp_maq2(1) -
temp_maq2(2)
            Else
                wait_maq2 = wait_maq2 + temp_maq2(2) -
temp_maq2(1)
            End If
        End If

        If i > 1 Then
            If temp_maq3(1) >= temp_maq3(2) Then
                idle_maq3 = idle_maq3 + temp_maq3(1) -
temp_maq3(2)
            Else
                wait_maq3 = wait_maq3 + temp_maq3(2) -
temp_maq3(1)
                "MsgBox (wait_maq3)
            End If
        End If

        i = i + 1

    ElseIf i = num_jobs + 1 Then
        If temp_maq1 >= temp_maq2(3) + Cells(3 + pos(i - 1), 4)
Then
            temp_maq2(3) = temp_maq1
        Else
            temp_maq2(3) = temp_maq2(3) + Cells(3 + pos(i - 1),
4)

        End If
        Cells(row2 + 1, col - 2 + i).Value = temp_maq2(3)
        Cells(row2 + 1, col - 2 + i).HorizontalAlignment =
xlRight
        i = i + 1

    Else
        If temp_maq2(3) >= temp_maq3(3) + Cells(3 + pos(i -
2), 5) Then
            temp_maq3(3) = temp_maq2(3)
        Else
            temp_maq3(3) = maximo3(i - 2) + Cells(3 + pos(i - 2),
5)

        End If
        Cells(row3 + 1, col - 3 + i).Value = temp_maq3(3)
        Cells(row3 + 1, col - 3 + i).HorizontalAlignment =
xlRight
        Range("N10").Value = temp_maq3(3)

```

```

    i = i + 1
End If

Loop

Range("Q6", "Q7").HorizontalAlignment = xlCenter
Range("L6", "L7").HorizontalAlignment = xlCenter
Range("L6").Value = idle_maq2
Range("Q6").Value = wait_maq2
Range("Q6").Font.ColorIndex = 2
If wait_maq2 = 0 Then
    Range("Q6").Interior.ColorIndex = 10
Else
    Range("Q6").Interior.ColorIndex = 9
End If
Range("L7").Value = idle_maq3
Range("Q7").Value = wait_maq3

End Sub

Function Copytomachine_3(menor, indice, maquina) As Integer

Dim cont As Integer
Dim seq_job As Integer

seq_jobs = 0

col = 9
row1 = 14
row2 = 18
row3 = 22

Cells(indice + 3, 2).Copy
If maquina = 1 Then
    cont = 1
    Do Until pos(cont) = 0
        cont = cont + 1
        Loop
        pos(cont) = indice
        ocupado(cont) = 1
        Range(Cells(row1, col - 1 + cont), Cells(row1 + 1, col - 1 +
cont)).PasteSpecial Paste:=xlPasteFormats
        Range(Cells(row2, col - 1 + cont), Cells(row2 + 1, col - 1 +
cont)).PasteSpecial Paste:=xlPasteFormats
        Range(Cells(row3, col - 1 + cont), Cells(row3 + 1, col - 1 +
cont)).PasteSpecial Paste:=xlPasteFormats
        Cells(row1, col - 1 + cont).Value = indice
        Cells(row1, col - 1 + cont).Value = Cells(row1, col - 1 +
cont).Value & "(" & Cells(3 + pos(cont), 3) & ")"
        Cells(row1, col - 1 + cont).Characters(1, 2).Font.Bold =
True
        Cells(row1, col - 1 + cont).Characters(3).Font.Size = 10

        Cells(row2, col - 1 + cont).Value = indice
        Cells(row2, col - 1 + cont).Value = Cells(row2, col - 1 +
cont).Value & "(" & Cells(3 + pos(cont), 4) & ")"
        Cells(row2, col - 1 + cont).Characters(1, 2).Font.Bold =
True
        Cells(row2, col - 1 + cont).Characters(3).Font.Size = 10

        Cells(row3, col - 1 + cont).Value = indice
        Cells(row3, col - 1 + cont).Value = Cells(row3, col - 1 +
cont).Value & "(" & Cells(3 + pos(cont), 5) & ")"
        Cells(row3, col - 1 + cont).Characters(1, 2).Font.Bold =
True
        Cells(row3, col - 1 + cont).Characters(3).Font.Size = 10
    Elseif maquina = 2 Then
        cont = num_jobs
        Do Until pos(cont) = 0
            cont = cont - 1

```

```

Loop
pos(cont) = indice
ocupado(cont) = 1
Range(Cells(row1, col - 1 + cont), Cells(row1 + 1, col - 1 +
cont)).PasteSpecial Paste:=xlPasteFormats
Range(Cells(row2, col - 1 + cont), Cells(row2 + 1, col - 1 +
cont)).PasteSpecial Paste:=xlPasteFormats
Range(Cells(row3, col - 1 + cont), Cells(row3 + 1, col - 1 +
cont)).PasteSpecial Paste:=xlPasteFormats
Cells(row1, col - 1 + cont).Value = indice
Cells(row1, col - 1 + cont).Value = Cells(row1, col - 1 +
cont).Value & "(" & Cells(3 + pos(cont), 3) & ")"
Cells(row1, col - 1 + cont).Characters(1, 2).Font.Bold =
True
Cells(row1, col - 1 + cont).Characters(3).Font.Size = 10

Cells(row2, col - 1 + cont).Value = indice
Cells(row2, col - 1 + cont).Value = Cells(row2, col - 1 +
cont).Value & "(" & Cells(3 + pos(cont), 4) & ")"
Cells(row2, col - 1 + cont).Characters(1, 2).Font.Bold =
True
Cells(row2, col - 1 + cont).Characters(3).Font.Size = 10

Cells(row3, col - 1 + cont).Value = indice
Cells(row3, col - 1 + cont).Value = Cells(row3, col - 1 +
cont).Value & "(" & Cells(3 + pos(cont), 5) & ")"
Cells(row3, col - 1 + cont).Characters(1, 2).Font.Bold =
True
Cells(row3, col - 1 + cont).Characters(3).Font.Size = 10
End If

cont = 1
Do Until cont = num_jobs + 1
    If pos(cont) <> 0 Then
        seq_jobs = seq_jobs + 1
    End If
    cont = cont + 1
Loop

Copytomachine_3 = seq_jobs

End Function

Sub Limpar_3()

Application.Workbooks("Sequenciamento da
Producao.xlsm").Worksheets("Três Máquinas").Activate

i = 1
col = 9
row1 = 14
row2 = 18
row3 = 22
linha_inicio = 4

num_jobs = calcular_ultimo(linha_inicio)
Range(Cells(row1, col - 1), Cells(row3 + 1, col +
20)).ClearFormats
Range(Cells(row1, col - 1), Cells(row3 + 1, col + 20)).Clear

soma = 0
Range("L6").Clear
Range("Q6").Clear
Range("L7").Clear
Range("Q7").Clear
Range("N10").ClearContents

Do Until i = num_jobs + 1
    ocupado(i) = 0
    pos(i) = 0
    i = i + 1
Loop

```

```

Do Until num_spot = num_jobs
  num_spot = Create_Spot(row1, col)
  num_spot = Create_Spot(row2, col)
  num_spot = Create_Spot(row3, col)
  MsgBox (num_spot)
  col = col + 1
Loop

col = 9

Cells(row1 + 1, col - 1).Value = 0
Cells(row1 + 1, col - 1).HorizontalAlignment = xlRight
Cells(row1 + 1, col - 1).Font.Size = 11

End Sub

Sub Seq_3_maq_prog()

Application.Workbooks("Sequenciamento da
Producao.xlsm").Worksheets("Três Máquinas").Activate

```

```

Dim i As Integer
Dim j As Integer
Dim num_spot As Integer
Dim menor As Integer
Dim maior As Integer
Dim k As Integer
Dim soma As Integer
Dim temp_maq1 As Integer
Dim temp_maq2(1 To 3) As Integer
Dim temp_maq3(1 To 3) As Integer
Dim maximo3(1 To 100) As Integer
Dim maximo2(1 To 100) As Integer
Dim idle_maq2 As Integer
Dim idle_maq3 As Integer
Dim wait_maq2 As Integer
Dim wait_maq3 As Integer

col = 9
row1 = 14
row2 = 18
row3 = 22

maior = 0
menor = 0
menor_ind = 0
menor_maq = 0

linha_inicio = 4
num_jobs = calcular_ultimo(linha_inicio)
ultimo = 3 + num_elementos

```

```

i = 1
j = 1
k = 1
soma = 0

Do Until k = num_jobs + 1
  soma = soma + ocupado(k)
  k = k + 1
Loop

If soma = 0 Then
  Do Until i = num_jobs + 1
    Do Until j = 6 + 1
      If j <= 4 Then
        jobs_3(i, j) = Cells(i + 3, j + 1).Text
        j = j + 1
      Else

```

```

        jobs_3(i, j) = jobs_3(i, j - 3) + jobs_3(i, j - 2)
        Cells(i + 3, j + 1) = jobs_3(i, j)
        j = j + 1
      End If
    Loop
  j = 1
  pos(i) = 0
  ocupado(i) = 0
  i = i + 1
Loop

```

```

Do Until num_spot = num_jobs
  num_spot = Create_Spot(row1, col)
  num_spot = Create_Spot(row2, col)
  num_spot = Create_Spot(row3, col)
  col = col + 1
Loop
End If

```

```

Do Until soma = num_jobs
  i = 1
  j = 5

```

```

Do Until i = num_jobs + 1
  Do Until j = 6 + 1
    If i = 1 And j = 5 Then
      maior = jobs_3(i, j)
      menor = jobs_3(i, j)
      menor_ind = i
      menor_maq = j - 4

      idle_maq2 = 0
      idle_maq3 = 0
      wait_maq2 = 0
      wait_maq3 = 0
    ElseIf jobs_3(i, j) < menor Then
      menor = jobs_3(i, j)
      menor_ind = i
      menor_maq = j - 4
    ElseIf jobs_3(i, j) > maior Then
      maior = jobs_3(i, j)
    End If
    j = j + 1
  Loop
  j = 5
  i = i + 1
Loop

sequenciados = Copytomachine_3(menor, menor_ind,
menor_maq)

```

```

jobs_3(menor_ind, 5) = menor + maior + 1
jobs_3(menor_ind, 6) = menor + maior + 1

```

```

If soma < num_jobs - 1 Then
  Exit Sub

```

```

ElseIf soma = num_jobs - 1 Then
  i = 1
  Do Until i = 4
    temp_maq1 = 0
    temp_maq2(i) = 0
    temp_maq3(i) = 0
    i = i + 1
  Loop

```

```

i = 1

col = 9
row1 = 14

```



```

row2 = 18
row3 = 22

Do Until i = num_jobs + 3

    If i <= num_jobs Then
        temp_maq1 = temp_maq1 + Cells(3 + pos(i), 3)
        Cells(row1 + 1, col - 1 + i).Value = temp_maq1
        Cells(row1 + 1, col - 1 + i).HorizontalAlignment =
xlRight
        Cells(row1 + 1, col - 1 + i).Font.Size = 11

        If i = 1 Then
            temp_maq2(1) = 0
            temp_maq2(2) = Cells(3 + pos(1), 3)
            temp_maq2(3) = temp_maq2(2)
            Cells(row2 + 1, col - 2 + i).Value =
temp_maq2(3)
            maximo2(i) = temp_maq2(3)
        Else
            temp_maq2(1) = temp_maq1
            temp_maq2(2) = maximo2(i - 1) + Cells(3 + pos(i
- 1), 4)

            If temp_maq2(1) >= temp_maq2(2) Then
                temp_maq2(3) = temp_maq2(1)
                maximo2(i) = temp_maq2(3)
                Cells(row2 + 1, col - 2 + i).Value =
temp_maq2(3)
            Else
                temp_maq2(3) = temp_maq2(2)
                maximo2(i) = temp_maq2(3)
                Cells(row2 + 1, col - 2 + i).Value =
temp_maq2(3)
            End If
            Cells(row2 + 1, col - 2 + i).HorizontalAlignment
= xlRight
            Cells(row2 + 1, col - 2 + i).Font.Size = 11
        End If

        If i = 1 Then
            temp_maq3(1) = 0
            temp_maq3(2) = temp_maq2(3) + Cells(3 +
pos(1), 4)
            temp_maq3(3) = temp_maq3(2)
            maximo3(1) = temp_maq3(3)
            Cells(row3 + 1, col - 2 + i).Value =
temp_maq3(3)
        Else
            temp_maq3(1) = temp_maq2(3) + Cells(3 +
pos(i), 4)
            temp_maq3(2) = maximo3(i - 1) + Cells(3 + pos(i
- 1), 5)

            If temp_maq3(1) >= temp_maq3(2) Then
                temp_maq3(3) = temp_maq3(1)
                maximo3(i) = temp_maq3(3)
                Cells(row3 + 1, col - 2 + i).Value =
temp_maq3(3)
            Else
                temp_maq3(3) = temp_maq3(2)
                maximo3(i) = temp_maq3(3)
                Cells(row3 + 1, col - 2 + i).Value =
temp_maq3(3)
            End If
        End If
        Cells(row3 + 1, col - 2 + i).HorizontalAlignment =
xlRight
        Cells(row3 + 1, col - 2 + i).Font.Size = 11

        If i > 1 Then
            If temp_maq2(1) >= temp_maq2(2) Then

```

```

            idle_maq2 = idle_maq2 + temp_maq2(1) -
temp_maq2(2)
            Else
                wait_maq2 = wait_maq2 + temp_maq2(2) -
temp_maq2(1)
            End If
        End If

        If i > 1 Then
            If temp_maq3(1) >= temp_maq3(2) Then
                idle_maq3 = idle_maq3 + temp_maq3(1) -
temp_maq3(2)
            Else
                wait_maq3 = wait_maq3 + temp_maq3(2) -
temp_maq3(1)
            End If
        End If

        i = i + 1

    ElseIf i = num_jobs + 1 Then
        If temp_maq1 >= temp_maq2(3) + Cells(3 + pos(i -
1), 4) Then
            temp_maq2(3) = temp_maq1
        Else
            temp_maq2(3) = temp_maq2(3) + Cells(3 + pos(i
- 1), 4)
        End If
        Cells(row2 + 1, col - 2 + i).Value = temp_maq2(3)
        Cells(row2 + 1, col - 2 + i).HorizontalAlignment =
xlRight
        i = i + 1

    Else
        If temp_maq2(3) >= temp_maq3(3) + Cells(3 +
pos(i - 2), 5) Then
            temp_maq3(3) = temp_maq2(3)
        Else
            temp_maq3(3) = maximo3(i - 2) + Cells(3 + pos(i
- 2), 5)
        End If
        Cells(row3 + 1, col - 3 + i).Value = temp_maq3(3)
        Range("N10").Value = temp_maq3(3)
        Cells(row3 + 1, col - 3 + i).HorizontalAlignment =
xlRight

        Range("Q6", "Q7").HorizontalAlignment = xlCenter
        Range("L6", "L7").HorizontalAlignment = xlCenter
        Range("L6").Value = idle_maq2
        Range("Q6").Value = wait_maq2
        Range("Q6").Font.ColorIndex = 2
        If wait_maq2 = 0 Then
            Range("Q6").Interior.ColorIndex = 10
        Else
            Range("Q6").Interior.ColorIndex = 9
        End If
        Range("L7").Value = idle_maq3
        Range("Q7").Value = wait_maq3

        i = i + 1
    End If

Loop

Exit Sub

End If

```

Loop

End Sub

ANEXO B – CÓDIGO DUAS MÁQUINAS

```
Public jobs_2(1 To 100, 1 To 3) As Integer
Public linha_inicio As Integer
Public num_jobs As Integer
Public ultimo As Integer
Public pos(1 To 100) As Integer
Public ocupado(1 To 100) As Integer
Public row1 As Integer "maquina 1 indice linha
Public col As Integer "maquina 1 e 2 indice coluna
Public row2 As Integer "maquina 2 indice linha

Public Function calcular_ultimo(l) As Integer
    ultimo = 1
    contador = 0
    Do Until ultimo = 0
        If Application.Workbooks("Flow-
shop.xlsm").Worksheets("Duas Máquinas").Cells(1, 2).Value =
Empty Then
            ultimo = 0
        Else
            ultimo = 1
            contador = contador + 1
            l = l + 1
        End If
    Loop
    calcular_ultimo = contador
End Function

Sub Seq_2_maq()

Application.Workbooks("Flow-
shop.xlsm").Worksheets("Duas Máquinas").Activate

Dim i As Integer
Dim j As Integer
Dim num_spot As Integer
Dim menor As Integer
Dim maior As Integer
Dim temp_maq1 As Integer
Dim temp_maq2 As Integer
Dim aux_tem_maq2 As Integer
Dim idle_maq2 As Integer
Dim wait_maq2 As Integer
Dim t2 As Integer

col = 9
row1 = 3
row2 = 7
t2 = 0

idle_maq2 = 0
wait_maq2 = 0

linha_inicio = 4
num_jobs = calcular_ultimo(linha_inicio)
ultimo = 3 + num_elementos

i = 1
j = 1

Do Until i = num_jobs + 1
    Do Until j = 3 + 1
        jobs_2(i, j) = Cells(i + 3, j + 1).Text
        j = j + 1
```

```
Loop
j = 1
pos(i) = 0
i = i + 1
Loop

"MsgBox (jobs_2(4, 3))

Do Until num_spot = num_jobs
    num_spot = Create_Spot(row1, col)
    num_spot = Create_Spot(row2, col)
    "MsgBox (num_spot)
    col = col + 1
Loop

Cells(4, 8).Value = 0
Cells(4, 8).HorizontalAlignment = xlRight
Cells(4, 8).Font.Size = 11

Do Until sequenciados = num_jobs

    i = 1
    j = 2

    Do Until i = num_jobs + 1
        Do Until j = 3 + 1
            If i = 1 And j = 2 Then
                maior = jobs_2(i, j)
                menor = jobs_2(i, j)
                menor_ind = i
                menor_maq = j - 1
            ElseIf jobs_2(i, j) < menor Then
                menor = jobs_2(i, j)
                menor_ind = i
                menor_maq = j - 1
            ElseIf jobs_2(i, j) > maior Then
                maior = jobs_2(i, j)
            End If
            j = j + 1
        Loop
        j = 2
        i = i + 1
    Loop

    sequenciados = Copytomachine(menor, menor_ind,
menor_maq)
    If sequenciados = 1 Then
        maior_inicial = maior
    End If
    jobs_2(menor_ind, 2) = menor + maior_inicial + 1
    jobs_2(menor_ind, 3) = menor + maior_inicial + 1
    "MsgBox (sequenciados)

Loop

temp_maq1 = 0
temp_maq2 = 0
i = 1

Do Until i = num_jobs + 2
    If i <= num_jobs Then
        temp_maq1 = temp_maq1 + Cells(3 + pos(i), 3)
        Cells(4, 8 + i).Value = temp_maq1
        Cells(4, 8 + i).HorizontalAlignment = xlRight
```

```

Cells(4, 8 + i).Font.Size = 11

If i = 1 Then
    temp_maq2 = temp_maq1
    t2 = temp_maq
    ElseIf temp_maq1 >= temp_maq2 + Cells(3 + pos(i -
1), 4) Then
        aux_temp_maq2 = temp_maq2 + Cells(3 + pos(i -
1), 4)
        t2 = aux_temp_maq2
        temp_maq2 = temp_maq1
    Else
        temp_maq2 = temp_maq2 + Cells(3 + pos(i - 1), 4)
        t2 = temp_maq2
    End If
    Cells(8, 7 + i).Value = t2 & " / " & temp_maq2
    Cells(8, 7 + i).HorizontalAlignment = xlRight
    Cells(8, 7 + i).Font.Size = 11

    If i > 1 Then
        If temp_maq1 >= temp_maq2 Then
            idle_maq2 = idle_maq2 + temp_maq1 -
aux_temp_maq2
        Else
            wait_maq2 = wait_maq2 + temp_maq2 -
temp_maq1
        End If
    End If

    i = i + 1

    ElseIf i = num_jobs + 1 Then
        If temp_maq1 >= temp_maq2 + Cells(3 + pos(i - 1), 4)
Then
            temp_maq2 = temp_maq1
        Else
            temp_maq2 = temp_maq2 + Cells(3 + pos(i - 1), 4)
        End If
        Cells(8, 7 + i).Value = temp_maq2
        Cells(8, 7 + i).HorizontalAlignment = xlRight
        Range("H14", "I14").Value = temp_maq2
        i = i + 1
    End If
Loop

i = 1
j = 1
Do Until i = num_jobs + 1
    Do Until j = 3 + 1
        jobs_2(i, j) = Cells(i + 3, j + 1).Text
        j = j + 1
    Loop
    j = 1
    pos(i) = 0
    i = i + 1
Loop

Range("M14").HorizontalAlignment = xlCenter
Range("Q14").HorizontalAlignment = xlCenter
Range("M14").Value = idle_maq2
Range("Q14").Value = wait_maq2

End Sub

Function Create_Spot(linha, coluna) As Integer
    ref = 8

    With ActiveSheet.Range(Cells(linha, coluna), Cells(linha +
1, coluna))
        .Borders(xlEdgeLeft).LineStyle = xlContinuous
        .Borders(xlEdgeRight).LineStyle = xlContinuous

```

```

        .Borders(xlEdgeBottom).LineStyle = xlContinuous
        .Borders(xlEdgeTop).LineStyle = xlContinuous
        .HorizontalAlignment = xlCenter
        .VerticalAlignment = xlCenter
        .WrapText = True
        .Orientation = 0
        .AddIndent = False
        .IndentLevel = 0
        .ShrinkToFit = False
        .ReadingOrder = xlContext
        ".MergeCells = True
    End With

    Create_Spot = coluna - ref

End Function

Function Copytomachine(menor, indice, maquina) As Integer

    Dim cont As Integer
    Dim seq_job As Integer

    seq_jobs = 0

    Cells(indice + 3, 2).Copy
    If maquina = 1 Then
        cont = 1
        Do Until pos(cont) = 0
            cont = cont + 1
        Loop
        pos(cont) = indice
        ocupado(cont) = 1
        Range(Cells(3, 8 + cont), Cells(4, 8 + cont)).PasteSpecial
Paste:=xlPasteFormats
        Range(Cells(7, 8 + cont), Cells(8, 8 + cont)).PasteSpecial
Paste:=xlPasteFormats
        Cells(3, 8 + cont).Value = indice
        Cells(3, 8 + cont).Value = Cells(3, 8 + cont).Value & " ("
& Cells(3 + pos(cont), 3) & ")"
        Cells(3, 8 + cont).Characters(1, 2).Font.Bold = True
        Cells(3, 8 + cont).Characters(3).Font.Size = 10

        Cells(7, 8 + cont).Value = indice
        Cells(7, 8 + cont).Value = Cells(7, 8 + cont).Value & " ("
& Cells(3 + pos(cont), 4) & ")"
        Cells(7, 8 + cont).Characters(1, 2).Font.Bold = True
        Cells(7, 8 + cont).Characters(3).Font.Size = 10

    ElseIf maquina = 2 Then
        cont = num_jobs
        Do Until pos(cont) = 0
            cont = cont - 1
        Loop
        pos(cont) = indice
        ocupado(cont) = 1
        Range(Cells(3, 8 + cont), Cells(4, 8 + cont)).PasteSpecial
Paste:=xlPasteFormats
        Range(Cells(7, 8 + cont), Cells(8, 8 + cont)).PasteSpecial
Paste:=xlPasteFormats
        Cells(3, 8 + cont).Value = indice
        Cells(3, 8 + cont).Value = Cells(3, 8 + cont).Value & " ("
& Cells(3 + pos(cont), 3) & ")"
        Cells(3, 8 + cont).Characters(1, 2).Font.Bold = True
        Cells(3, 8 + cont).Characters(3).Font.Size = 10

        Cells(7, 8 + cont).Value = indice
        Cells(7, 8 + cont).Value = Cells(7, 8 + cont).Value & " ("
& Cells(3 + pos(cont), 4) & ")"
        Cells(7, 8 + cont).Characters(1, 2).Font.Bold = True
        Cells(7, 8 + cont).Characters(3).Font.Size = 10
    End If

```

```

cont = 1
Do Until cont = num_jobs + 1
  If pos(cont) <> 0 Then
    seq_jobs = seq_jobs + 1
  End If
  cont = cont + 1
Loop

Copytomachine = seq_jobs

End Function

Sub Limpar()

linha_inicio = 4
num_jobs = calcular_ultimo(linha_inicio)
Range(Cells(3, 8), Cells(8, 9 + 20)).ClearFormats
Range(Cells(3, 8), Cells(8, 9 + 20)).Clear

i = 1
col = 9
row1 = 3
row2 = 7

soma = 0
sequenciados = 0

Range("M14").ClearContents
Range("Q14").ClearContents
Range("H14", "I14").ClearContents

Do Until i = num_jobs
  ocupado(i) = 0
  pos(i) = 0
  i = i + 1
Loop

Do Until num_spot = num_jobs
  num_spot = Create_Spot(row1, col)
  num_spot = Create_Spot(row2, col)
  "MsgBox (num_spot)
  col = col + 1
Loop

Cells(4, 8).Value = 0
Cells(4, 8).HorizontalAlignment = xlRight
Cells(4, 8).Font.Size = 11

End Sub

Sub Seq_2_maq_prog()

Application.Workbooks("Flow-
shop.xlsm").Worksheets("Duas Máquinas").Activate

Dim i As Integer
Dim j As Integer
Dim num_spot As Integer
Dim menor As Integer
Dim maior As Integer
Dim k As Integer
Dim soma As Integer
Dim aux_tem_maq2 As Integer
Dim idle_maq2 As Integer
Dim wait_maq2 As Integer
Dim t2 As Integer

col = 9
row1 = 3
row2 = 7
t2 = 0

```

```

linha_inicio = 4
num_jobs = calcular_ultimo(linha_inicio)
ultimo = 3 + num_elementos

i = 1
j = 1
k = 1
soma = 0

Do Until k = num_jobs + 1
  soma = soma + ocupado(k)
  k = k + 1
Loop

If soma = 0 Then
  Do Until i = num_jobs + 1
    Do Until j = 3 + 1
      jobs_2(i, j) = Cells(i + 3, j + 1)
      j = j + 1
    Loop
    j = 1
    ocupado(i) = 0
    pos(i) = 0
    i = i + 1
  Loop
End If

Do Until num_spot = num_jobs
  num_spot = Create_Spot(row1, col)
  num_spot = Create_Spot(row2, col)
  "MsgBox (num_spot)
  col = col + 1
Loop
End If

Do Until soma = num_jobs
  i = 1
  j = 2

  Do Until i = num_jobs + 1
    Do Until j = 3 + 1
      If i = 1 And j = 2 Then
        maior = jobs_2(i, j)
        menor = jobs_2(i, j)
        menor_ind = i
        menor_maq = j - 1
      ElseIf jobs_2(i, j) < menor Then
        menor = jobs_2(i, j)
        menor_ind = i
        menor_maq = j - 1
      ElseIf jobs_2(i, j) > maior Then
        maior = jobs_2(i, j)
      End If
      j = j + 1
    Loop
    j = 2
    i = i + 1
  Loop

  sequenciados = Copytomachine(menor, menor_ind,
menor_maq)
  jobs_2(menor_ind, 2) = menor + maior + 1
  jobs_2(menor_ind, 3) = menor + maior + 1

  If soma < num_jobs - 1 Then
    Exit Sub

  ElseIf soma = num_jobs - 1 Then

    temp_maq1 = 0
    temp_maq2 = 0
    i = 1

```

```

Do Until i = num_jobs + 2
  If i <= num_jobs Then
    temp_maq1 = temp_maq1 + Cells(3 + pos(i), 3)
    Cells(4, 8 + i).Value = temp_maq1
    Cells(4, 8 + i).HorizontalAlignment = xlRight
    Cells(4, 8 + i).Font.Size = 11

    If i = 1 Then
      temp_maq2 = temp_maq1
    ElseIf temp_maq1 >= temp_maq2 + Cells(3 + pos(i
- 1), 4) Then
      aux_temp_maq2 = temp_maq2 + Cells(3 + pos(i -
1), 4)
      t2 = aux_temp_maq2
      temp_maq2 = temp_maq1
    Else
      temp_maq2 = temp_maq2 + Cells(3 + pos(i - 1),
4)
      t2 = temp_maq2
    End If
    Cells(8, 7 + i).Value = t2 & " / " & temp_maq2
    Cells(8, 7 + i).HorizontalAlignment = xlRight
    Cells(8, 7 + i).Font.Size = 11

    If i > 1 Then
      If temp_maq1 >= temp_maq2 Then
        idle_maq2 = idle_maq2 + temp_maq1 -
aux_temp_maq2
      Else
        wait_maq2 = wait_maq2 + temp_maq2 -
temp_maq1
      End If
    End If

    i = i + 1
  ElseIf i = num_jobs + 1 Then

```

```

    If temp_maq1 >= temp_maq2 + Cells(3 + pos(i - 1),
4) Then
      temp_maq2 = temp_maq1
    Else
      temp_maq2 = temp_maq2 + Cells(3 + pos(i - 1),
4)
    End If
    Cells(8, 7 + i).Value = temp_maq2
    Range("H14", "I14").Value = temp_maq2
    Cells(8, 7 + i).HorizontalAlignment = xlRight
    i = i + 1
  End If
Loop

i = 1
j = 1
Do Until i = num_jobs + 1
  Do Until j = 3 + 1
    jobs_2(i, j) = Cells(i + 3, j + 1).Text
    j = j + 1
  Loop
  j = 1
  pos(i) = 0
  i = i + 1
Loop

Range("M14").HorizontalAlignment = xlCenter
Range("Q14").HorizontalAlignment = xlCenter
Range("M14").Value = idle_maq2
Range("Q14").Value = wait_maq2

End If

Exit Sub
Loop
End Sub

```

ANEXO C – CÓDIGO TRÊS MÁQUINAS

```
Public jobs_3(1 To 100, 1 To 6) As Integer
Public linha_inicio As Integer
Public num_jobs As Integer
Public ultimo As Integer
Public pos(1 To 100) As Integer
Public ocupado(1 To 100) As Integer
Public row1 As Integer "maquina 1 indice linha
Public col As Integer "maquina 1 e 2 indice coluna
Public row2 As Integer "maquina 2 indice linha
Public row3 As Integer "maquina 3 indice linha
```

```
Public Function calcular_ultimo(l) As Integer
    ultimo = 1
    contador = 0
    Do Until ultimo = 0
        If Application.Workbooks("Flow-
shop.xlsm").Worksheets("Duas Máquinas").Cells(1, 2).Value =
Empty Then
            ultimo = 0
        Else
            ultimo = 1
            contador = contador + 1
            l = l + 1
        End If
    Loop
    calcular_ultimo = contador
End Function
```

```
Function Create_Spot(linha, coluna) As Integer
    ref = 8

    With ActiveSheet.Range(Cells(linha, coluna), Cells(linha +
1, coluna))
        .Borders(xlEdgeLeft).LineStyle = xlContinuous
        .Borders(xlEdgeRight).LineStyle = xlContinuous
        .Borders(xlEdgeBottom).LineStyle = xlContinuous
        .Borders(xlEdgeTop).LineStyle = xlContinuous
        .HorizontalAlignment = xlCenter
        .VerticalAlignment = xlCenter
        .WrapText = True
        .Orientation = 0
        .AddIndent = False
        .IndentLevel = 0
        .ShrinkToFit = False
        .ReadingOrder = xlContext
        .MergeCells = True
    End With

    Create_Spot = coluna - ref

End Function
```

```
Sub Seq_3_maq()

Application.Workbooks("Flow-shop.xlsm").Worksheets("Três
Máquinas").Activate

Dim i As Integer
Dim j As Integer
Dim num_spot As Integer
Dim menor As Integer
Dim maior As Integer
Dim temp_maq1 As Integer
Dim temp_maq2(1 To 3) As Integer
Dim temp_maq3(1 To 3) As Integer
```

```
Dim maximo3(1 To 100) As Integer
Dim maximo2(1 To 100) As Integer
Dim idle_maq2 As Integer
Dim idle_maq3 As Integer
Dim wait_maq2 As Integer
Dim wait_maq3 As Integer
```

```
col = 9
row1 = 14
row2 = 18
row3 = 22
```

```
idle_maq2 = 0
idle_maq3 = 0
wait_maq2 = 0
wait_maq3 = 0
```

```
sequenciados = 0
linha_inicio = 4
num_jobs = calcular_ultimo(linha_inicio)
ultimo = 3 + num_elementos
```

```
i = 1
j = 1
```

```
Do Until i = num_jobs + 1
    Do Until j = 6 + 1
        If j <= 4 Then
            jobs_3(i, j) = Cells(i + 3, j + 1).Text
            j = j + 1
        Else
            jobs_3(i, j) = jobs_3(i, j - 3) + jobs_3(i, j - 2)
            Cells(i + 3, j + 1) = jobs_3(i, j)
            j = j + 1
        End If
    Loop
    j = 1
    pos(i) = 0
    ocupado(i) = 0
    i = i + 1
Loop
```

```
"MsgBox (jobs_2(4, 3))
```

```
Do Until num_spot = num_jobs
    num_spot = Create_Spot(row1, col)
    num_spot = Create_Spot(row2, col)
    num_spot = Create_Spot(row3, col)
    "MsgBox (num_spot)
    col = col + 1
Loop
```

```
col = 9
```

```
Cells(row1 + 1, col - 1).Value = 0
Cells(row1 + 1, col - 1).HorizontalAlignment = xlRight
Cells(row1 + 1, col - 1).Font.Size = 11
```

```
Do Until sequenciados = num_jobs
```

```
i = 1
j = 5
```

```

Do Until i = num_jobs + 1
  Do Until j = 6 + 1
    If i = 1 And j = 5 Then
      maior = jobs_3(i, j)
      menor = jobs_3(i, j)
      menor_ind = i
      menor_maq = j - 4
    ElseIf jobs_3(i, j) < menor Then
      menor = jobs_3(i, j)
      menor_ind = i
      menor_maq = j - 4
    ElseIf jobs_3(i, j) > maior Then
      maior = jobs_3(i, j)
    End If
    j = j + 1
  Loop
  j = 5
  i = i + 1
Loop

sequenciados = Copytomachine_3(menor, menor_ind,
menor_maq)
If sequenciados = 1 Then
  maior_inicial = maior
End If
jobs_3(menor_ind, 5) = menor + maior_inicial + 1
jobs_3(menor_ind, 6) = menor + maior_inicial + 1
"MsgBox (sequenciados)

Loop

i = 1
Do Until i = 4
  temp_maq1 = 0
  temp_maq2(i) = 0
  temp_maq3(i) = 0
  i = i + 1
Loop

i = 1

col = 9
row1 = 14
row2 = 18
row3 = 22

Do Until i = num_jobs + 3
  If i <= num_jobs Then
    temp_maq1 = temp_maq1 + Cells(3 + pos(i), 3)
    Cells(row1 + 1, col - 1 + i).Value = temp_maq1
    Cells(row1 + 1, col - 1 + i).HorizontalAlignment =
xlRight
    Cells(row1 + 1, col - 1 + i).Font.Size = 11

    If i = 1 Then
      temp_maq2(1) = 0
      temp_maq2(2) = Cells(3 + pos(1), 3)
      temp_maq2(3) = temp_maq2(2)
      Cells(row2 + 1, col - 2 + i).Value = "0 / " &
temp_maq2(3)
    temp_maq2(3)
      maximo2(i) = temp_maq2(3)
    Else
      temp_maq2(1) = temp_maq1
      temp_maq2(2) = maximo2(i - 1) + Cells(3 + pos(i - 1),
4)

      If temp_maq2(1) >= temp_maq2(2) Then
        temp_maq2(3) = temp_maq2(1)
        maximo2(i) = temp_maq2(3)
        Cells(row2 + 1, col - 2 + i).Value = temp_maq2(2)
& " / " & temp_maq2(3)
      Else

```

```

      temp_maq2(3) = temp_maq2(2)
      maximo2(i) = temp_maq2(3)
      Cells(row2 + 1, col - 2 + i).Value = temp_maq2(2)
& " / " & temp_maq2(3)
    End If
    Cells(row2 + 1, col - 3 + i).HorizontalAlignment =
xlRight
    Cells(row2 + 1, col - 3 + i).Font.Size = 11
    End If

    If i = 1 Then
      temp_maq3(1) = 0
      temp_maq3(2) = temp_maq2(3) + Cells(3 + pos(1), 4)
      temp_maq3(3) = temp_maq3(2)
      Cells(row3 + 1, col - 2 + i).Value = "0 / " &
temp_maq3(3)
    temp_maq3(3)
      maximo3(1) = temp_maq3(3)
    Else
      temp_maq3(1) = temp_maq2(3) + Cells(3 + pos(i), 4)
      temp_maq3(2) = maximo3(i - 1) + Cells(3 + pos(i - 1),
5)

      If temp_maq3(1) >= temp_maq3(2) Then
        temp_maq3(3) = temp_maq3(1)
        maximo3(i) = temp_maq3(3)
        Cells(row3 + 1, col - 2 + i).Value = temp_maq3(2)
& " / " & temp_maq3(3)
      Else
        temp_maq3(3) = temp_maq3(2)
        maximo3(i) = temp_maq3(3)
        Cells(row3 + 1, col - 2 + i).Value = temp_maq3(2)
& " / " & temp_maq3(3)
      End If
    End If
    Cells(row3 + 1, col - 2 + i).HorizontalAlignment =
xlRight
    Cells(row3 + 1, col - 2 + i).Font.Size = 11

    If i > 1 Then
      If temp_maq2(1) >= temp_maq2(2) Then
        idle_maq2 = idle_maq2 + temp_maq2(1) -
temp_maq2(2)
      Else
        wait_maq2 = wait_maq2 + temp_maq2(2) -
temp_maq2(1)
      End If
    End If

    If i > 1 Then
      If temp_maq3(1) >= temp_maq3(2) Then
        idle_maq3 = idle_maq3 + temp_maq3(1) -
temp_maq3(2)
      Else
        wait_maq3 = wait_maq3 + temp_maq3(2) -
temp_maq3(1)
        "MsgBox (wait_maq3)
      End If
    End If

    i = i + 1

  ElseIf i = num_jobs + 1 Then
    If temp_maq1 >= temp_maq2(3) + Cells(3 + pos(i - 1), 4)
Then
      temp_maq2(3) = temp_maq1
    Else
      temp_maq2(3) = temp_maq2(3) + Cells(3 + pos(i - 1),
4)
    End If
    Cells(row2 + 1, col - 2 + i).Value = temp_maq2(3)
    Cells(row2 + 1, col - 2 + i).HorizontalAlignment =
xlRight

```



```

        i = i + 1
    Else
        If temp_maq2(3) >= temp_maq3(3) + Cells(3 + pos(i - 2), 5) Then
            temp_maq3(3) = temp_maq2(3)
        Else
            temp_maq3(3) = maximo3(i - 2) + Cells(3 + pos(i - 2), 5)
        End If
        Cells(row3 + 1, col - 3 + i).Value = temp_maq3(3)
        Cells(row3 + 1, col - 3 + i).HorizontalAlignment = xlRight
        Range("N10").Value = temp_maq3(3)
        i = i + 1
    End If
End Sub

Loop

Range("Q6", "Q7").HorizontalAlignment = xlCenter
Range("L6", "L7").HorizontalAlignment = xlCenter
Range("L6").Value = idle_maq2
Range("Q6").Value = wait_maq2
Range("Q6").Font.ColorIndex = 2
If wait_maq2 = 0 Then
    Range("Q6").Interior.ColorIndex = 10
Else
    Range("Q6").Interior.ColorIndex = 9
End If
Range("L7").Value = idle_maq3
Range("Q7").Value = wait_maq3

End Sub

Function Copytomachine_3(menor, indice, maquina) As Integer

Dim cont As Integer
Dim seq_job As Integer

seq_jobs = 0

col = 9
row1 = 14
row2 = 18
row3 = 22

Cells(indice + 3, 2).Copy
If maquina = 1 Then
    cont = 1
    Do Until pos(cont) = 0
        cont = cont + 1
    Loop
    pos(cont) = indice
    ocupado(cont) = 1
    Range(Cells(row1, col - 1 + cont), Cells(row1 + 1, col - 1 + cont)).PasteSpecial Paste:=xlPasteFormats
    Range(Cells(row2, col - 1 + cont), Cells(row2 + 1, col - 1 + cont)).PasteSpecial Paste:=xlPasteFormats
    Range(Cells(row3, col - 1 + cont), Cells(row3 + 1, col - 1 + cont)).PasteSpecial Paste:=xlPasteFormats
    Cells(row1, col - 1 + cont).Value = indice
    Cells(row1, col - 1 + cont).Value = Cells(row1, col - 1 + cont).Value & "(" & Cells(3 + pos(cont), 3) & ")"
    Cells(row1, col - 1 + cont).Characters(1, 2).Font.Bold = True
    Cells(row1, col - 1 + cont).Characters(3).Font.Size = 10

    Cells(row2, col - 1 + cont).Value = indice
    Cells(row2, col - 1 + cont).Value = Cells(row2, col - 1 + cont).Value & "(" & Cells(3 + pos(cont), 4) & ")"

```

```

    Cells(row2, col - 1 + cont).Characters(1, 2).Font.Bold = True
    Cells(row2, col - 1 + cont).Characters(3).Font.Size = 10

    Cells(row3, col - 1 + cont).Value = indice
    Cells(row3, col - 1 + cont).Value = Cells(row3, col - 1 + cont).Value & "(" & Cells(3 + pos(cont), 5) & ")"
    Cells(row3, col - 1 + cont).Characters(1, 2).Font.Bold = True
    Cells(row3, col - 1 + cont).Characters(3).Font.Size = 10

ElseIf maquina = 2 Then
    cont = num_jobs
    Do Until pos(cont) = 0
        cont = cont - 1
    Loop
    pos(cont) = indice
    ocupado(cont) = 1
    Range(Cells(row1, col - 1 + cont), Cells(row1 + 1, col - 1 + cont)).PasteSpecial Paste:=xlPasteFormats
    Range(Cells(row2, col - 1 + cont), Cells(row2 + 1, col - 1 + cont)).PasteSpecial Paste:=xlPasteFormats
    Range(Cells(row3, col - 1 + cont), Cells(row3 + 1, col - 1 + cont)).PasteSpecial Paste:=xlPasteFormats
    Cells(row1, col - 1 + cont).Value = indice
    Cells(row1, col - 1 + cont).Value = Cells(row1, col - 1 + cont).Value & "(" & Cells(3 + pos(cont), 3) & ")"
    Cells(row1, col - 1 + cont).Characters(1, 2).Font.Bold = True
    Cells(row1, col - 1 + cont).Characters(3).Font.Size = 10

    Cells(row2, col - 1 + cont).Value = indice
    Cells(row2, col - 1 + cont).Value = Cells(row2, col - 1 + cont).Value & "(" & Cells(3 + pos(cont), 4) & ")"
    Cells(row2, col - 1 + cont).Characters(1, 2).Font.Bold = True
    Cells(row2, col - 1 + cont).Characters(3).Font.Size = 10

    Cells(row3, col - 1 + cont).Value = indice
    Cells(row3, col - 1 + cont).Value = Cells(row3, col - 1 + cont).Value & "(" & Cells(3 + pos(cont), 5) & ")"
    Cells(row3, col - 1 + cont).Characters(1, 2).Font.Bold = True
    Cells(row3, col - 1 + cont).Characters(3).Font.Size = 10
End If

cont = 1
Do Until cont = num_jobs + 1
    If pos(cont) <> 0 Then
        seq_jobs = seq_jobs + 1
    End If
    cont = cont + 1
Loop

Copytomachine_3 = seq_jobs

End Function

Sub Limpar_3()

Application.Workbooks("Flow-shop.xlsm").Worksheets("Três Máquinas").Activate

i = 1
col = 9
row1 = 14
row2 = 18
row3 = 22
linha_inicio = 4

num_jobs = calcular_ultimo(linha_inicio)

```

```
Range(Cells(row1, col - 1), Cells(row3 + 1, col + 20)).ClearFormats
Range(Cells(row1, col - 1), Cells(row3 + 1, col + 20)).Clear
```

```
soma = 0
Range("L6").Clear
Range("Q6").Clear
Range("L7").Clear
Range("Q7").Clear
Range("N10").ClearContents
```

```
Do Until i = num_jobs + 1
    ocupado(i) = 0
    pos(i) = 0
    i = i + 1
Loop
```

```
Do Until num_spot = num_jobs
    num_spot = Create_Spot(row1, col)
    num_spot = Create_Spot(row2, col)
    num_spot = Create_Spot(row3, col)
    MsgBox (num_spot)
    col = col + 1
Loop
```

```
col = 9
```

```
Cells(row1 + 1, col - 1).Value = 0
Cells(row1 + 1, col - 1).HorizontalAlignment = xlRight
Cells(row1 + 1, col - 1).Font.Size = 11
```

```
End Sub
```

```
Sub Seq_3_maq_prog()
```

```
Application.Workbooks("Flow-shop.xlsm").Worksheets("Três Máquinas").Activate
```

```
Dim i As Integer
Dim j As Integer
Dim num_spot As Integer
Dim menor As Integer
Dim maior As Integer
Dim k As Integer
Dim soma As Integer
Dim temp_maq1 As Integer
Dim temp_maq2(1 To 3) As Integer
Dim temp_maq3(1 To 3) As Integer
Dim maximo3(1 To 100) As Integer
Dim maximo2(1 To 100) As Integer
Dim idle_maq2 As Integer
Dim idle_maq3 As Integer
Dim wait_maq2 As Integer
Dim wait_maq3 As Integer
```

```
col = 9
row1 = 14
row2 = 18
row3 = 22
```

```
maior = 0
menor = 0
menor_ind = 0
menor_maq = 0
```

```
linha_inicio = 4
num_jobs = calcular_ultimo(linha_inicio)
ultimo = 3 + num_elementos
```

```
i = 1
```

```
j = 1
k = 1
soma = 0
```

```
Do Until k = num_jobs + 1
    soma = soma + ocupado(k)
    k = k + 1
Loop
```

```
If soma = 0 Then
    Do Until i = num_jobs + 1
        Do Until j = 6 + 1
            If j <= 4 Then
                jobs_3(i, j) = Cells(i + 3, j + 1).Text
                j = j + 1
            Else
                jobs_3(i, j) = jobs_3(i, j - 3) + jobs_3(i, j - 2)
                Cells(i + 3, j + 1) = jobs_3(i, j)
                j = j + 1
            End If
        Loop
        j = 1
        pos(i) = 0
        ocupado(i) = 0
        i = i + 1
    Loop
```

```
Do Until num_spot = num_jobs
    num_spot = Create_Spot(row1, col)
    num_spot = Create_Spot(row2, col)
    num_spot = Create_Spot(row3, col)
    col = col + 1
Loop
```

```
Do Until num_spot = num_jobs
    num_spot = Create_Spot(row1, col)
    num_spot = Create_Spot(row2, col)
    num_spot = Create_Spot(row3, col)
    col = col + 1
Loop
```

```
End If
```

```
Do Until soma = num_jobs
    i = 1
    j = 5
```

```
Do Until i = num_jobs + 1
    Do Until j = 6 + 1
        If i = 1 And j = 5 Then
            maior = jobs_3(i, j)
            menor = jobs_3(i, j)
            menor_ind = i
            menor_maq = j - 4

            idle_maq2 = 0
            idle_maq3 = 0
            wait_maq2 = 0
            wait_maq3 = 0
            ElseIf jobs_3(i, j) < menor Then
                menor = jobs_3(i, j)
                menor_ind = i
                menor_maq = j - 4
            ElseIf jobs_3(i, j) > maior Then
                maior = jobs_3(i, j)
            End If
            j = j + 1
        Loop
        j = 5
        i = i + 1
    Loop
```

```
sequenciados = Copytomachine_3(menor, menor_ind, menor_maq)
```

```
jobs_3(menor_ind, 5) = menor + maior + 1
jobs_3(menor_ind, 6) = menor + maior + 1
```

```
If soma < num_jobs - 1 Then
```

```

Exit Sub

ElseIf soma = num_jobs - 1 Then
    i = 1
    Do Until i = 4
        temp_maq1 = 0
        temp_maq2(i) = 0
        temp_maq3(i) = 0
        i = i + 1
    Loop

    i = 1

    col = 9
    row1 = 14
    row2 = 18
    row3 = 22

    Do Until i = num_jobs + 3

        If i <= num_jobs Then
            temp_maq1 = temp_maq1 + Cells(3 + pos(i), 3)
            Cells(row1 + 1, col - 1 + i).Value = temp_maq1
            Cells(row1 + 1, col - 1 + i).HorizontalAlignment =
xlRight
            Cells(row1 + 1, col - 1 + i).Font.Size = 11

            If i = 1 Then
                temp_maq2(1) = 0
                temp_maq2(2) = Cells(3 + pos(1), 3)
                temp_maq2(3) = temp_maq2(2)
                Cells(row2 + 1, col - 2 + i).Value = "0 / " &
temp_maq2(3)
                maximo2(i) = temp_maq2(3)
            Else
                temp_maq2(1) = temp_maq1
                temp_maq2(2) = maximo2(i - 1) + Cells(3 + pos(i
- 1), 4)

                If temp_maq2(1) >= temp_maq2(2) Then
                    temp_maq2(3) = temp_maq2(1)
                    maximo2(i) = temp_maq2(3)
                    Cells(row2 + 1, col - 2 + i).Value =
temp_maq2(2) & " / " & temp_maq2(3)
                Else
                    temp_maq2(3) = temp_maq2(2)
                    maximo2(i) = temp_maq2(3)
                    Cells(row2 + 1, col - 2 + i).Value =
temp_maq2(2) & " / " & temp_maq2(3)
                End If
                Cells(row2 + 1, col - 3 + i).HorizontalAlignment
= xlRight
                Cells(row2 + 1, col - 3 + i).Font.Size = 11
            End If

            If i = 1 Then
                temp_maq3(1) = 0
                temp_maq3(2) = temp_maq2(3) + Cells(3 +
pos(1), 4)
                temp_maq3(3) = temp_maq3(2)
                maximo3(1) = temp_maq3(3)
                Cells(row3 + 1, col - 2 + i).Value = "0 / " &
temp_maq3(3)
            Else
                temp_maq3(1) = temp_maq2(3) + Cells(3 +
pos(i), 4)
                temp_maq3(2) = maximo3(i - 1) + Cells(3 + pos(i
- 1), 5)

                If temp_maq3(1) >= temp_maq3(2) Then
                    temp_maq3(3) = temp_maq3(1)
                    maximo3(i) = temp_maq3(3)

```

```

                Cells(row3 + 1, col - 2 + i).Value =
temp_maq3(2) & " / " & temp_maq3(3)
            Else
                temp_maq3(3) = temp_maq3(2)
                maximo3(i) = temp_maq3(3)
                Cells(row3 + 1, col - 2 + i).Value =
temp_maq3(2) & " / " & temp_maq3(3)
            End If
        End If
        Cells(row3 + 1, col - 2 + i).HorizontalAlignment =
xlRight
        Cells(row3 + 1, col - 2 + i).Font.Size = 11

        If i > 1 Then
            If temp_maq2(1) >= temp_maq2(2) Then
                idle_maq2 = idle_maq2 + temp_maq2(1) -
temp_maq2(2)
            Else
                wait_maq2 = wait_maq2 + temp_maq2(2) -
temp_maq2(1)
            End If
        End If

        If i > 1 Then
            If temp_maq3(1) >= temp_maq3(2) Then
                idle_maq3 = idle_maq3 + temp_maq3(1) -
temp_maq3(2)
            Else
                wait_maq3 = wait_maq3 + temp_maq3(2) -
temp_maq3(1)
            End If
        End If

        i = i + 1

        ElseIf i = num_jobs + 1 Then
            If temp_maq1 >= temp_maq2(3) + Cells(3 + pos(i -
1), 4) Then
                temp_maq2(3) = temp_maq1
            Else
                temp_maq2(3) = temp_maq2(3) + Cells(3 + pos(i
- 1), 4)
            End If
            Cells(row2 + 1, col - 2 + i).Value = temp_maq2(3)
            Cells(row2 + 1, col - 2 + i).HorizontalAlignment =
xlRight
            i = i + 1

            Else
                If temp_maq2(3) >= temp_maq3(3) + Cells(3 +
pos(i - 2), 5) Then
                    temp_maq3(3) = temp_maq2(3)
                Else
                    temp_maq3(3) = maximo3(i - 2) + Cells(3 + pos(i
- 2), 5)
                End If
                Cells(row3 + 1, col - 3 + i).Value = temp_maq3(3)
                Range("N10").Value = temp_maq3(3)
                Cells(row3 + 1, col - 3 + i).HorizontalAlignment =
xlRight

                Range("Q6", "Q7").HorizontalAlignment = xlCenter
                Range("L6", "L7").HorizontalAlignment = xlCenter
                Range("L6").Value = idle_maq2
                Range("Q6").Value = wait_maq2
                Range("Q6").Font.ColorIndex = 2
                If wait_maq2 = 0 Then
                    Range("Q6").Interior.ColorIndex = 10
                Else
                    Range("Q6").Interior.ColorIndex = 9
                End If

```

```
Range("L7").Value = idle_maq3
Range("Q7").Value = wait_maq3

    i = i + 1
End If

Loop

Exit Sub
End If
Loop
End Sub
```

ANEXO D – CÓDIGO N - TRABALHOS M - MÁQUINAS

```

Public num_jobs As Integer
Public num_mach As Integer
Public Pij(100) As Integer
Public Pij_antes(100) As Integer
Public TMIT(100) As Integer
Public resol(1 To 100, 1 To 100) As Integer
Public seq As Integer
Public confirmados(100) As Integer
Public conf_antes(100) As Integer
Public i_menorTMIT As Integer
Public pos As Integer
Public empate As Integer
Public i_perdeu(100) As Integer
Public i_venceu(100) As Integer
Public copias As Integer
Public makespan As Integer
Public n_makespan As Integer
Public i_makespan(100) As Integer
Public desempate As Integer
Public seqsol(1 To 100, 1 To 100) As Integer
Public seqsol_antes(1 To 100, 1 To 100) As Integer
Public iguais As Integer
Public total_empates As Integer
Public seq_antes As Integer
Public primeira As Integer
Public lin As Integer
Public colp As Integer

Sub Build_table()
Dim i As Integer
Dim ex_jobs As Integer
Dim ex_mach As Integer
Dim letras As String

num_jobs = Range("E2").Value
num_mach = Range("E3").Value

k = 1
i = 1
l_border = 0
c_border = 0
seq = 0
makespan = 0
n_makespan = 0
desempate = 0
total_empates = 0

Do Until Range("B" & i + 7) = Empty
    i = i + 1
Loop

ex_jobs = i - 1

i = 1

Do Until Cells(7, i + 2) = Empty
    i = i + 1
Loop

c = 1
Do Until Range("L" & c + 2).Value = False
    Range("L" & c + 2).Clear

```

```

    c = c + 1
Loop

Range("P3").Clear

ex_mach = i - 2

i = ex_jobs

If num_jobs < ex_jobs Then
    Range(Cells(num_jobs + 8, 2), Cells(ex_jobs + 7, 2 +
ex_mach * 2)).Clear
    Range(Cells(num_jobs + 8, 2), Cells(ex_jobs + 7, ex_mach
+ 3)).Font.ColorIndex = 1

Elseif num_jobs > ex_jobs Then
    Do Until i = num_jobs
        Range("B" & i + 8, Cells(i + 8, 2 + ex_mach * 2)).Clear
        Range("B" & i + 8).Value = i + 1
        Range("B" & i + 8).HorizontalAlignment = xlCenter
        Range("B" & i + 8).Interior.ColorIndex = num_jobs + i *
2
        i = i + 1
    Loop
End If

i = ex_mach

If num_mach < ex_mach Then
    Range(Cells(7, num_mach + 3), Cells(ex_jobs + 7,
ex_mach + 3)).ClearContents
    Cells(7, num_mach + 3) = ChrW("&H" & 2211) & "Pij"
    Cells(7, num_mach + 3).Activate
    ActiveCell.Characters(Start:=3, Length:=2).Font.Subscript
= True
    ActiveCell.Font.Bold = True
    ActiveCell.Font.ColorIndex = 0
Elseif num_mach > ex_mach Then
    Do Until i = num_mach
        Range(Cells(7, 3 + i), Cells(7 + ex_jobs, 3 + i)).Clear
        Cells(7, 3 + i).Value = i + 1
        Cells(7, 3 + i).Font.Bold = True
        Cells(7, 3 + i).Font.Color = RGB(0, 172, 80)
        Cells(7, 3 + i).HorizontalAlignment = xlCenter
        Cells(7, num_mach + 3) = ChrW("&H" & 2211) & "Pij"
        Cells(7, num_mach + 3).Activate
        ActiveCell.Characters(Start:=3, Length:=2).Font.Subscript
= True
        ActiveCell.Font.Bold = True
        ActiveCell.Interior.ColorIndex = 0
        i = i + 1
    Loop
End If

letra = ConvertToLetter(num_mach + 2)
sum (letra)

"Range(Cells(ex_jobs + 9, 2), Cells(ex_jobs + 10 + ex_jobs *
2, ex_mach + 6 + ex_mach * 2)).Clear

linha_i = num_jobs + 9
col_i = 2

```

```

Tabela = tabela_res(linha_i, col_i)

End Sub

Public Function tabela_res(lin, col) As Integer
ActiveSheet.Range(Cells(lin - 1, col), Cells(lin + 500, col + 500)).Clear
ActiveSheet.Range("B6", Cells(7, num_mach + 3)).Copy
ActiveSheet.Cells(lin, col).PasteSpecial
ActiveSheet.Range(Cells(lin + 1, col + 1), Cells(lin + 1, col + 2 + num_mach)).Clear

i = 1
num = 1
Total = num_mach * 2
Do Until i >= Total
    With ActiveSheet.Range(Cells(lin + 1, col + i), Cells(lin + 1, col + i + 1))
        .MergeCells = True
        .HorizontalAlignment = xlCenter
        .VerticalAlignment = xlCenter
        .WrapText = True
        .Font.Color = RGB(0, 172, 80)
        .Value = num
        .Font.Bold = True
    End With
    i = i + 2
    num = num + 1
Loop

Range(Cells(lin, col + 1), Cells(lin, col + i - 1)).MergeCells = True

Cells(lin + 1, col + i) = "Total Machine Idle Time"
With Range(Cells(lin + 1, col + i), Cells(lin + 1, col + i + 2))
    .MergeCells = True
    .HorizontalAlignment = xlCenter
    .Font.Bold = True
End With

tabela_res = 1
End Function

Function sum(col) As Integer

l = 1

Do Until l >= num_jobs + 1

    Cells(7 + l, num_mach + 3) = "=sum(C" & 7 + l & ":" & col & 7 + l & ")"
    Cells(7 + l, num_mach + 3).HorizontalAlignment = xlRight
    Cells(7 + l, num_mach + 3).Font.ColorIndex = 0
    l = l + 1
Loop

End Function

Function ConvertToLetter(iCol As Integer) As String
Dim iAlpha As Integer
Dim iRemainder As Integer
iAlpha = Int(iCol / 27)
iRemainder = iCol - (iAlpha * 26)
If iAlpha > 0 Then
    ConvertToLetter = Chr(iAlpha + 64)
End If
If iRemainder > 0 Then
    ConvertToLetter = ConvertToLetter & Chr(iRemainder + 64)

```

```

End If
End Function

Function clear_i_makespan()

i = 1
Do Until i > n_jobs
    i_makespan(i) = 0
Loop
i = i + 1

End Function

Sub NJobs_MMachines_Prog()
Dim menorP(4) As Integer
Dim maior As Integer
Dim i_menorP(4) As Integer
Dim colp As Integer
Dim lin As Integer
Dim i As Integer
Dim k As Integer
Dim tp As Integer
Dim menorTMIT As Integer

If seq = 0 And total_empates = 0 Then
    empate = 0
    copias = 0
    desempate = 0
    primeira = 0
End If

If seq >= num_jobs And total_empates > desempate Then
    seq = seq_antes
    primeira = 0
    kont = 1
    Do Until kont >= num_jobs + 1
        confirmados(kont) = conf_antes(kont)
        Pij(kont) = Pij_antes(kont)
        seqsol(copias + 2, kont) = seqsol_antes(copias + 2, kont)
        kont = kont + 1
    Loop

    "confirmados(seq + 1) = i_perdeu(empate)
    copias = copias + 1
ElseIf seq >= num_jobs Then
    Exit Sub
End If

num_jobs = Range("E2").Value
num_mach = Range("E3").Value

lin = (copias + 1) * num_jobs + 11 + copias * 4
colp = num_mach + 3

If seq > 0 And copias > 0 And primeira = 0 Then
    Range(Cells(lin + seq + i_perdeu(empate) + 1, 2), Cells(lin + seq + i_perdeu(empate) + 1, num_mach * 2 + 2)).Copy
    Range("B" & lin + seq + 1).PasteSpecial
    Range(Cells(lin + seq + 2, 2), Cells(lin + seq + 4, num_mach * 2 + 4)).Clear

```

```

confirmados(seq + 1) = Range("B" & lin + seq +
1).Value
seqsol(copias + 1, seq + 1) = confirmados(seq + 1)
primeira = 1
desempate = desempate + 1
Elseif seq > 0 Then
  "MsgBox (i_menorTMIT)
  Range(Cells(lin + seq + i_menorTMIT + 1, 2), Cells(lin +
seq + i_menorTMIT + 1, num_mach * 2 + 2)).Copy
  Range("B" & lin + seq + 1).PasteSpecial
  Range(Cells(lin + seq + 2, 2), Cells(lin + seq + 4,
num_mach * 2 + 4)).Clear
  confirmados(seq + 1) = Range("B" & lin + seq + 1)
  seqsol(copias + 1, seq + 1) = confirmados(seq + 1)
End If
If seq <= num_jobs Then

  i = 1

  Do Until i >= num_jobs + 1
    Pij(i) = Cells(i + 7, colp)
    If i = 1 Then
      maior = Pij(i)
    Elseif Pij(i) > maior Then
      maior = Pij(i)
    End If

    i = i + 1
  Loop

  If seq > 0 Then
    i = seq + 1
    Do Until i = 1
      "MsgBox (confirmados(i))
      Pij(confirmados(i)) = Cells(confirmados(i) + 7, colp)
+ maior
      "MsgBox (Pij(confirmados(i)))

      i = i - 1
    Loop
  End If

  i = 1

  i = 1
  k = 1

  "Pij(confirmados(seq + 1)) = Cells(confirmados(seq + 1)
+ 9, colP)

  Do Until k >= 5
    Do Until i >= num_jobs + 1
      If i = 1 Then
        "maior = Pij(i)
        menorP(k) = Pij(i)
        i_menorP(k) = i
      Elseif Pij(i) < menorP(k) Then
        menorP(k) = Pij(i)
        i_menorP(k) = i
      End If
      "MsgBox (Pij(i))
      i = i + 1
    Loop
    Pij(i_menorP(k)) = maior + menorP(k)
    k = k + 1
    i = 1
  Loop

  If seq = 0 Then
    confirmados(seq + 1) = i_menorP(1)
    Cells(7 + i_menorP(1), 2).Copy
    Cells(lin + 1, 2).PasteSpecial

```

```

Cells(lin + 1, 2).HorizontalAlignment = xlCenter
seqsol(copias + 1, 1) = i_menorP(1)
End If
"MsgBox (confirmados(seq + 1))
"MsgBox (i_menorP(1))
Cells(lin, 2) = "Confirmados"
With Cells(lin, 2)
  .HorizontalAlignment = xlCenter
  .Font.Italic = True
End With

y = 1
k = 1
x = 1
pos = 0
seq = seq + 1
i = seq

Do Until i >= 5 + seq Or num_jobs - seq + 5 < i

  If seq > 0 And i = seq + 1 Then
    Cells(lin + i, 2) = "Em teste"
    Cells(lin + i, 2).HorizontalAlignment = xlCenter
    i = i + 1
  End If

  If y > 1 And y <= 4 Then
    Cells(7 + i_menorP(y), 2).Copy
    Cells(lin + i, 2).PasteSpecial
    Cells(lin + i, 2).HorizontalAlignment = xlCenter
  End If

  "MsgBox (i_menorP(y))

  Do Until k >= num_mach * 2 + 1
    If k Mod 2 = 1 Then
      If seq = 1 And k = 1 Then
        Cells(lin + i, k + 2) = Cells(7 + i_menorP(y), x
+ 2)
        Cells(lin + i, k + 2).HorizontalAlignment =
xlCenter
        tp = tp + Cells(lin + seq, k + 2)
        x = x + 1
      Elseif seq = 1 Then
        Cells(lin + i, k + 2) = Cells(7 + i_menorP(y), x
+ 2)
        Cells(lin + i, k + 2).HorizontalAlignment =
xlCenter
        If i <= 1 Then
          tp = tp + Cells(lin + i, k + 2)
        End If
        x = x + 1
      Elseif seq > 1 And i > seq Then
        Cells(lin + i, k + 2) = Cells(7 + i_menorP(y), x
+ 2)
        Cells(lin + i, k + 2).HorizontalAlignment =
xlCenter
        If i <= 1 Then
          tp = tp + Cells(lin + i, k + 2)
        End If
        x = x + 1
      End If
      "MsgBox (tp)
      Elseif k Mod 2 = 0 Then
        "tp = tp + Cells(lin + i, k + 1)
        If i > seq Then

          If (Cells(lin + seq, k + 2) + Cells(lin + i, k + 1))
>= tp + Cells(lin + i, k + 1) Then
            tp = (Cells(lin + seq, k + 2) + Cells(lin + i, k
+ 1))

```

```

Cells(lin + i, k + 2) = tp
Cells(lin + i, k + 2).HorizontalAlignment =
xlCenter
Else
TMIT(i_menorP(y)) = TMIT(i_menorP(y)) +
tp - Cells(lin + seq, k + 2)
If TMIT(i_menorP(y)) < i_menorTMIT Then
i_menorTMIT = TMIT(i_menorP(y))
End If
tp = tp + Cells(lin + i, k + 1)
Cells(lin + i, k + 2) = tp
Cells(lin + i, k + 2).HorizontalAlignment =
xlCenter
End If
ElseIf i <= seq And seq = 1 Then
Cells(lin + i, k + 2) = tp
Cells(lin + i, k + 2).HorizontalAlignment =
xlCenter
End If
Cells(lin + i, k + 2).HorizontalAlignment =
xlCenter
End If
"MsgBox (tp)
k = k + 1
Loop
If i > 1 Then
If i = seq + 2 Then
menorTMIT = TMIT(i_menorP(y))
pos = 1
i_menorTMIT = pos
ElseIf TMIT(i_menorP(y)) < menorTMIT Then
menorTMIT = TMIT(i_menorP(y))
i_menorTMIT = pos
ElseIf pos > 0 And TMIT(i_menorP(y)) =
menorTMIT Then
entrar = 1
If i_perdeu(empate) = pos And
i_venceu(empate) = i_menorTMIT Then
menorTMIT = TMIT(i_perdeu(empate))
i_menorTMIT = i_perdeu(empate)
confirmados(seq + 1) = i_perdeu(empate)
"MsgBox (i_menorTMIT)
desempate = desempate + 1
done = done + 1
empate = empate - 1
Else
"MsgBox (i_menorP(y))
empate = empate + 1
total_empates = total_empates + 1
i_venceu(empate) = i_menorTMIT
i_perdeu(empate) = pos
"MsgBox (i_menorTMIT)
"MsgBox (pos)
End If
End If
pos = pos + 1
If i > seq Then
Cells(lin + i, k + 2) = TMIT(i_menorP(y))
Cells(lin + i, k + 2).HorizontalAlignment =
xlCenter
End If
End If
x = 1

```

```

k = 1
i = i + 1
If y < 4 Then
y = y + 1
End If
tp = 0
Loop
If entrar = 1 Then
Range(Cells(lin, 2), Cells(lin + seq + 4, 4 + num_jobs
* 2)).Copy
Cells(lin + 4 + num_jobs, 2).PasteSpecial
seq_antes = seq
kont = 1
Do Until kont >= num_jobs + 1
conf_antes(kont) = confirmados(kont)
Pij_antes(kont) = Pij(confirmados(i))
seqsol_antes(copias + 2, kont) = seqsol(copias + 1,
kont)
kont = kont + 1
Loop
End If
k = 1
i = 1
tp = 0
Do Until i > 4
TMIT(i_menorP(i)) = 0
i = i + 1
Loop
If seq = num_jobs Then
If copias = 0 Then
makespan = Cells(lin + num_jobs, 2 + (num_mach *
2))
i_makespan(copias + 1) = copias + 1
n_makespan = n_makespan + 1
ElseIf Cells(lin + num_jobs, 2 + (num_mach * 2)) <
makespan Then
makespan = Cells(lin + num_jobs, 2 + (num_mach *
2))
makespan = 1
clear_i_makespan
i_makespan(copias + 1) = copias + 1
ElseIf Cells(lin + num_jobs, 2 + (num_mach * 2)) =
makespan Then
makespan = Cells(lin + num_jobs, 2 + (num_mach *
2))
n_makespan = n_makespan + 1
i_makespan(copias + 1) = copias + 1
End If
"MsgBox (total_empates)
"MsgBox (copias)
Range("L3").Activate
If total_empates = desempate Then
i = 1
Do Until i >= desempate + 2
If i_makespan(i) > 0 Then
Cells((i * num_jobs + 11 + (i - 1) * 4) +
num_jobs, 2 + (num_mach * 2)).Activate
ActiveCell.Interior.ColorIndex = 10
ActiveCell.Font.ColorIndex = 2
c = 1
Do Until c >= num_jobs + 1
If c = 1 Then
"MsgBox (seqsol(i_makespan(i), c))
" Range("L" & i_makespan(i) + 2).Value =
seqsol(i_makespan(i), c)

```



```

        "Else
            Range("L" & i_makespan(i) + 2).Value =
Range("L" & i_makespan(i) + 2).Value & " - " &
seqsol(i_makespan(i), c)
            "End If
            c = c + 1
        Loop

        i = i + 1
        Range("P3").Activate
        Range("P3").Value = makespan
        ActiveCell.Interior.ColorIndex = 10
        ActiveCell.Font.ColorIndex = 2
    End If
    Loop
End If

End If

End Sub

Sub NJobs_MMachines()
Dim menorP(4) As Integer
Dim maior As Integer
Dim i_menorP(4) As Integer
Dim colp As Integer
Dim lin As Integer
Dim i As Integer
Dim k As Integer
Dim tp As Integer
Dim menorTMIT As Integer

Do Until seq > num_jobs
    entrar = 0
    If seq = 0 And total_empates = 0 Then
        empate = 0
        copias = 0
        desempate = 0
        primeira = 0
    End If

    If seq >= num_jobs And total_empates > desempate Then
        seq = seq_antes
        primeira = 0
        kont = 1
        Do Until kont >= num_jobs + 1
            confirmados(kont) = conf_antes(kont)
            Pij(kont) = Pij_antes(kont)
            seqsol(copias + 2, kont) = seqsol_antes(copias + 2,
kont)
            kont = kont + 1
        Loop

        "confirmados(seq + 1) = i_perdeu(empate)
        copias = copias + 1
    ElseIf seq >= num_jobs Then
        Exit Sub
    End If

    num_jobs = Range("E2").Value
    num_mach = Range("E3").Value

    lin = (copias + 1) * num_jobs + 11 + copias * 4
    colp = num_mach + 3

```

```

    If seq > 0 And copias > 0 And primeira = 0 Then
        Range(Cells(lin + seq + i_perdeu(empate) + 1, 2),
Cells(lin + seq + i_perdeu(empate) + 1, num_mach * 2 +
2)).Copy
        Range("B" & lin + seq + 1).PasteSpecial
        Range(Cells(lin + seq + 2, 2), Cells(lin + seq + 4,
num_mach * 2 + 4)).Clear
        confirmados(seq + 1) = Range("B" & lin + seq +
1).Value
        seqsol(copias + 1, seq + 1) = confirmados(seq + 1)
        primeira = 1
        desempate = desempate + 1
    ElseIf seq > 0 Then
        "MsgBox (i_menorTMIT)
        Range(Cells(lin + seq + i_menorTMIT + 1, 2), Cells(lin +
seq + i_menorTMIT + 1, num_mach * 2 + 2)).Copy
        Range("B" & lin + seq + 1).PasteSpecial
        Range(Cells(lin + seq + 2, 2), Cells(lin + seq + 4,
num_mach * 2 + 4)).Clear
        confirmados(seq + 1) = Range("B" & lin + seq + 1)
        seqsol(copias + 1, seq + 1) = confirmados(seq + 1)
    End If
    If seq <= num_jobs Then

        i = 1

        Do Until i >= num_jobs + 1
            Pij(i) = Cells(i + 7, colp)
            If i = 1 Then
                maior = Pij(i)
            ElseIf Pij(i) > maior Then
                maior = Pij(i)
            End If

            i = i + 1
        Loop

        If seq > 0 Then
            i = seq + 1
            Do Until i = 1
                "MsgBox (confirmados(i))
                Pij(confirmados(i)) = Cells(confirmados(i) + 7, colp)
            + maior
                "MsgBox (Pij(confirmados(i)))

                i = i - 1
            Loop
        End If

        i = 1
        k = 1

        "Pij(confirmados(seq + 1)) = Cells(confirmados(seq + 1)
+ 9, colP)

        Do Until k >= 5
            Do Until i >= num_jobs + 1
                If i = 1 Then
                    "maior = Pij(i)
                    menorP(k) = Pij(i)
                    i_menorP(k) = i
                ElseIf Pij(i) < menorP(k) Then
                    menorP(k) = Pij(i)
                    i_menorP(k) = i
                End If
                "MsgBox (Pij(i))
                i = i + 1
            Loop
            Pij(i_menorP(k)) = maior + menorP(k)
            k = k + 1

```

```

i = 1
Loop

If seq = 0 Then
confirmados(seq + 1) = i_menorP(1)
Cells(7 + i_menorP(1), 2).Copy
Cells(lin + 1, 2).PasteSpecial
Cells(lin + 1, 2).HorizontalAlignment = xlCenter
seqsol(copias + 1, 1) = i_menorP(1)
End If
"MsgBox (confirmados(seq + 1))
"MsgBox (i_menorP(1))
Cells(lin, 2) = "Confirmados"
With Cells(lin, 2)
.HorizontalAlignment = xlCenter
.Font.Italic = True
End With

y = 1
k = 1
x = 1
pos = 0
seq = seq + 1
i = seq

Do Until i >= 5 + seq Or num_jobs - seq + 5 < i

If seq > 0 And i = seq + 1 Then
Cells(lin + i, 2) = "Em teste"
Cells(lin + i, 2).HorizontalAlignment = xlCenter
i = i + 1
End If

If y > 1 And y <= 4 Then
Cells(7 + i_menorP(y), 2).Copy
Cells(lin + i, 2).PasteSpecial
Cells(lin + i, 2).HorizontalAlignment = xlCenter
End If

" MsgBox (i_menorP(y))

Do Until k >= num_mach * 2 + 1
If k Mod 2 = 1 Then
If seq = 1 And k = 1 Then
Cells(lin + i, k + 2) = Cells(7 + i_menorP(y), x
+ 2)
Cells(lin + i, k + 2).HorizontalAlignment =
xlCenter
tp = tp + Cells(lin + seq, k + 2)
x = x + 1
ElseIf seq = 1 Then
Cells(lin + i, k + 2) = Cells(7 + i_menorP(y), x
+ 2)
Cells(lin + i, k + 2).HorizontalAlignment =
xlCenter
If i <= 1 Then
tp = tp + Cells(lin + i, k + 2)
End If
x = x + 1
ElseIf seq > 1 And i > seq Then
Cells(lin + i, k + 2) = Cells(7 + i_menorP(y), x
+ 2)
Cells(lin + i, k + 2).HorizontalAlignment =
xlCenter
If i <= 1 Then
tp = tp + Cells(lin + i, k + 2)
End If
x = x + 1
End If
" MsgBox (tp)
ElseIf k Mod 2 = 0 Then

```

```

"tp = tp + Cells(lin + i, k + 1)
If i > seq Then

If (Cells(lin + seq, k + 2) + Cells(lin + i, k + 1))
>= tp + Cells(lin + i, k + 1) Then
tp = (Cells(lin + seq, k + 2) + Cells(lin + i, k
+ 1))
Cells(lin + i, k + 2) = tp
Cells(lin + i, k + 2).HorizontalAlignment =
xlCenter

Else

TMIT(i_menorP(y)) = TMIT(i_menorP(y)) +
tp - Cells(lin + seq, k + 2)

If TMIT(i_menorP(y)) < i_menorTMIT Then
i_menorTMIT = TMIT(i_menorP(y))
End If
tp = tp + Cells(lin + i, k + 1)
Cells(lin + i, k + 2) = tp
Cells(lin + i, k + 2).HorizontalAlignment =
xlCenter

End If
ElseIf i <= seq And seq = 1 Then
Cells(lin + i, k + 2) = tp
Cells(lin + i, k + 2).HorizontalAlignment =
xlCenter

End If
Cells(lin + i, k + 2).HorizontalAlignment =
xlCenter

End If

"MsgBox (tp)
k = k + 1
Loop
If i > 1 Then
If i = seq + 2 Then
menorTMIT = TMIT(i_menorP(y))
pos = 1
i_menorTMIT = pos
ElseIf TMIT(i_menorP(y)) < menorTMIT Then
menorTMIT = TMIT(i_menorP(y))
i_menorTMIT = pos
ElseIf pos > 0 And TMIT(i_menorP(y)) =
menorTMIT Then
entrar = 1
If i_perdeu(empate) = pos And
i_venceu(empate) = i_menorTMIT Then
menorTMIT = TMIT(i_perdeu(empate))

i_menorTMIT = i_perdeu(empate)
confirmados(seq + 1) = i_perdeu(empate)
"MsgBox (i_menorTMIT)
desempate = desempate + 1

done = done + 1
empate = empate - 1
Else
"MsgBox (i_menorP(y))
empate = empate + 1
total_empates = total_empates + 1
i_venceu(empate) = i_menorTMIT
i_perdeu(empate) = pos
"MsgBox (i_menorTMIT)
"MsgBox (pos)

End If
End If

pos = pos + 1
If i > seq Then

```

```

Cells(lin + i, k + 2) = TMIT(i_menorP(y))
Cells(lin + i, k + 2).HorizontalAlignment =
xlCenter
End If
End If
x = 1
k = 1
i = i + 1
If y < 4 Then
y = y + 1
End If

tp = 0
Loop
If entrar = 1 Then

Range(Cells(lin, 2), Cells(lin + seq + 4, 4 + num_jobs
* 2)).Copy
Cells(lin + 4 + num_jobs, 2).PasteSpecial

seq_antes = seq
kont = 1
Do Until kont >= num_jobs + 1
conf_antes(kont) = confirmados(kont)
Pij_antes(kont) = Pij(confirmados(i))
seqsol_antes(copias + 2, kont) = seqsol(copias + 1,
kont)
kont = kont + 1
Loop
End If

k = 1

i = 1
tp = 0
Do Until i > 4
TMIT(i_menorP(i)) = 0
i = i + 1
Loop

If seq = num_jobs Then
If copias = 0 Then
makespan = Cells(lin + num_jobs, 2 + (num_mach *
2))
i_makespan(copias + 1) = copias + 1
n_makespan = n_makespan + 1
Elseif Cells(lin + num_jobs, 2 + (num_mach * 2)) <
makespan Then
makespan = Cells(lin + num_jobs, 2 + (num_mach *
2))
makespan = 1
clear_i_makespan

```

```

i_makespan(copias + 1) = copias + 1
Elseif Cells(lin + num_jobs, 2 + (num_mach * 2)) =
makespan Then
makespan = Cells(lin + num_jobs, 2 + (num_mach *
2))
n_makespan = n_makespan + 1
i_makespan(copias + 1) = copias + 1
End If

"MsgBox (total_empates)
"MsgBox (copias)
Range("L3").Activate
If total_empates = desempate Then
i = 1
Do Until i >= desempate + 2
If i_makespan(i) > 0 Then
Cells((i * num_jobs + 11 + (i - 1) * 4) +
num_jobs, 2 + (num_mach * 2)).Activate
ActiveCell.Interior.ColorIndex = 10
ActiveCell.Font.ColorIndex = 2
c = 1
Do Until c >= num_jobs + 1
If c = 1 Then
"MsgBox (seqsol(i_makespan(i), c))
" Range("L" & i_makespan(i) + 2).Value =
seqsol(i_makespan(i), c)
"Else
Range("L" & i_makespan(i) + 2).Value =
Range("L" & i_makespan(i) + 2).Value & " - " &
seqsol(i_makespan(i), c)
"End If
c = c + 1
Loop

i = i + 1
Range("P3").Activate
Range("P3").Value = makespan
ActiveCell.Interior.ColorIndex = 10
ActiveCell.Font.ColorIndex = 2
End If
Loop
End If

End If

Loop

End Sub

```


ANEXO E – CÓDIGO MÁQUINAS PARALELAS

```

Public njobs As Integer
Public jobs(1 To 2, 1 To 100) As Integer
Public ordjobs(1 To 2, 1 To 100) As Integer
Public RCT As Integer
Public TT As Integer
Public nmach As Integer
Public ord As Integer
Public seq As Integer
Public dispRCT(100) As Integer
Public seqmach(100) As Integer

Sub LimparDados()

i = 1

Do Until Cells(2, i + 2).Value = False And Cells(3, i +
2).Value = False
    Cells(5, i + 2).Value = ""
    Cells(6, i + 2).Value = ""
    Cells(5, i + 2).Interior.ColorIndex = 0
    jobs(1, i) = 0
    jobs(2, i) = 0
    seqmach(i) = 0
    i = i + 1
Loop
njobs = i - 1
RCT = 0
TT = 0
done = 0
ord = 0
seq = 0

Range(Cells(16, 3), Cells(16 + njobs, 100)).Clear

End Sub

Sub Table_Parall()
Dim i As Integer
Dim maior As Integer
Dim ord As Integer
Dim auxjobs(1 To 2, 1 To 100) As Integer

maior = 0
TT = 0
i = 1

Do Until Cells(2, i + 2).Value = False And Cells(3, i +
2).Value = False
    jobs(1, i) = Cells(2, i + 2).Value
    jobs(2, i) = Cells(3, i + 2).Value
    auxjobs(1, i) = Cells(2, i + 2).Value
    auxjobs(2, i) = Cells(3, i + 2).Value
    TT = TT + jobs(2, i)
    i = i + 1
Loop

njobs = i - 1
nmach = Range("C9")
RCT = TT / nmach

Range(Cells(2, 2), Cells(3, 2 + njobs)).Copy
Cells(5, 2).PasteSpecial
"Application.SendKeys "{ESC}"

Do Until ord >= njobs
    i = 1

Do Until i = njobs + 1
    If i = 1 Then
        maior = auxjobs(2, i)
        imaior = i
    ElseIf auxjobs(2, i) > maior Then
        maior = auxjobs(2, i)
        imaior = i
    End If
    i = i + 1
Loop

Cells(5, ord + 3).Value = imaior
cor = Cells(2, 2 + imaior).Interior.ColorIndex
Cells(5, ord + 3).Interior.ColorIndex = cor
Cells(6, ord + 3).Value = maior
"MsgBox (imaior)
auxjobs(2, imaior) = 0
maior = 0
imaior = 0
ord = ord + 1
Loop

Build_Mach (nmach)

End Sub

Function Build_Mach(nmach) As Integer
Dim col As Integer
Dim lin As Integer
Dim done As Integer

lin = 16
col = 3
done = 0

Do Until done = nmach
    Range(Cells(lin, col + done * 3), Cells(lin, col + done * 3 +
2)).MergeCells = True
    Cells(lin, col + done * 3) = "Máquina " & done + 1
    Cells(lin, col + done * 3).HorizontalAlignment = xlCenter
    Cells(lin, col + done * 3).Font.Bold = True
    Cells(lin, col + done * 3).Interior.ColorIndex = 33 + done +
1
    Cells(lin + 1, col + done * 3).Value = "Trabalho"
    Cells(lin + 1, col + done * 3).Font.Underline =
xlUnderlineStyleSingle
    Cells(lin + 1, col + done * 3).HorizontalAlignment =
xlCenter
    Cells(lin + 1, col + done * 3 + 1).Value = "T. Proc."
    Cells(lin + 1, col + done * 3 + 1).Font.Underline =
xlUnderlineStyleSingle
    Cells(lin + 1, col + done * 3 + 1).HorizontalAlignment =
xlCenter
    Cells(lin + 1, col + done * 3 + 2).Value = ("RCT = " &
RCT)
    Cells(lin + 1, col + done * 3 + 2).Font.Underline =
xlUnderlineStyleSingle
    Cells(lin + 1, col + done * 3 + 2).HorizontalAlignment =
xlCenter
    If done < nmach - 1 Then
        Range(Cells(lin, col + done * 3), Cells(lin + 1, col + done
* 3 + 2)).Borders(xlEdgeRight).LineStyle = xlContinuous
    End If
    Range(Cells(lin, col + done * 3), Cells(lin + 1, col + done *
3 + 2)).Borders(xlInsideHorizontal).LineStyle = xlContinuous
    done = done + 1
Loop

```

```

End Function

Sub Paral()
Dim maior As Integer
Dim imaior As Integer
Dim col As Integer
Dim lin As Integer

lin = 16
col = 3
TT = 0
i = 1

If seq = 0 Then
  Do Until Cells(5, i + 2).Value = False And Cells(6, i +
2).Value = False
    ordjobs(1, i) = Cells(5, i + 2).Value
    ordjobs(2, i) = Cells(6, i + 2).Value
    TT = TT + ordjobs(2, i)
    i = i + 1
  Loop
  njobs = i - 1
  i = 1
  Do Until i = nmach + 1
    dispRCT(i) = RCT
    i = i + 1
  Loop
End If

nmach = Range("C9")

RCT = TT / nmach

Do Until seq >= njobs - 1

  i = 1

  Do Until i + seq >= njobs
    If i = 1 Then
      imaior = ordjobs(1, i + seq)
      maior = ordjobs(2, i + seq)
    ElseIf ordjobs(2, i + seq) > maior Then
      imaior = ordjobs(1, i + seq)
      maior = ordjobs(2, i + seq)
    End If
    i = i + 1
  Loop

  i = 1

  Do Until i > nmach
    If maior < dispRCT(i) And seq + 1 < njobs Then
      Cells(lin + 2 + seqmach(i), col + (i - 1) * 3).Value =
imaior
      Cells(lin + 2 + seqmach(i), col + (i - 1) *
3).HorizontalAlignment = xlCenter
      Cells(lin + 2 + seqmach(i), col + 1 + (i - 1) *
3).Value = maior
      Cells(lin + 2 + seqmach(i), col + 1 + (i - 1) *
3).HorizontalAlignment = xlCenter
      dispRCT(i) = dispRCT(i) - maior
      Cells(lin + 2 + seqmach(i), col + 2 + (i - 1) *
3).Value = dispRCT(i)
      Cells(lin + 2 + seqmach(i), col + 2 + (i - 1) *
3).HorizontalAlignment = xlCenter
      If i < nmach Then
        Cells(lin + 2 + seqmach(i), col + 2 + (i - 1) *
3).Borders(xlEdgeRight).LineStyle = xlContinuous
      Else
        Cells(lin + 2 + seqmach(i), col + (i - 1) *
3).Borders(xlEdgeLeft).LineStyle = xlContinuous

```

```

End If
seqmach(i) = seqmach(i) + 1
seq = seq + 1
i = nmach + 1

Else
  i = i + 1
End If
Loop
Loop
End Sub

Sub Paral_Prog()
Dim maior As Integer
Dim imaior As Integer
Dim col As Integer
Dim lin As Integer

lin = 16
col = 3
TT = 0
i = 1

If seq = 0 Then
  Do Until Cells(5, i + 2).Value = False And Cells(6, i +
2).Value = False
    ordjobs(1, i) = Cells(5, i + 2).Value
    ordjobs(2, i) = Cells(6, i + 2).Value
    TT = TT + ordjobs(2, i)
    i = i + 1
  Loop
  njobs = i - 1
  i = 1
  Do Until i = nmach + 1
    dispRCT(i) = RCT
    i = i + 1
  Loop
End If

nmach = Range("C9")

RCT = TT / nmach

i = 1

Do Until i + seq >= njobs + 1
  If i = 1 Then
    imaior = ordjobs(1, i + seq)
    maior = ordjobs(2, i + seq)
  ElseIf ordjobs(2, i + seq) > maior Then
    imaior = ordjobs(1, i + seq)
    maior = ordjobs(2, i + seq)
  End If
  i = i + 1
Loop

i = 1
"Do Until seq = njobs
  Do Until i > nmach

    If maior <= dispRCT(i) And seq + 1 < njobs Then
      Cells(lin + 2 + seqmach(i), col + (i - 1) * 3).Value =
imaior
      cor = Cells(2, 2 + imaior).Interior.ColorIndex
      Cells(lin + 2 + seqmach(i), col + (i - 1) *
3).Interior.ColorIndex = cor
      Cells(lin + 2 + seqmach(i), col + (i - 1) *
3).HorizontalAlignment = xlCenter
      Cells(lin + 2 + seqmach(i), col + 1 + (i - 1) * 3).Value
= maior

```

```

Cells(lin + 2 + seqmach(i), col + 1 + (i - 1) *
3).HorizontalAlignment = xlCenter
dispRCT(i) = dispRCT(i) - maior
Cells(lin + 2 + seqmach(i), col + 2 + (i - 1) * 3).Value
= dispRCT(i)
Cells(lin + 2 + seqmach(i), col + 2 + (i - 1) *
3).HorizontalAlignment = xlCenter
If i = 1 Then
Cells(lin + 2 + seqmach(i), col + 2 + (i - 1) *
3).Borders(xlEdgeRight).LineStyle = xlContinuous
ElseIf i > 1 And i < nmach Then
Cells(lin + 2 + seqmach(i), col + 2 + (i - 1) *
3).Borders(xlEdgeRight).LineStyle = xlContinuous
Cells(lin + 2 + seqmach(i), col + (i - 1) *
3).Borders(xlEdgeLeft).LineStyle = xlContinuous
Else
Cells(lin + 2 + seqmach(i), col + (i - 1) *
3).Borders(xlEdgeLeft).LineStyle = xlContinuous
End If
seqmach(i) = seqmach(i) + 1
seq = seq + 1
i = nmach + 1
ElseIf i = nmach And seq < njobs Then
k = 1
Do Until k >= nmach + 1
If k = 1 Then
maiorRCT = dispRCT(k)
imaiorRCT = k
ElseIf maiorRCT < dispRCT(k) Then
maiorRCT = dispRCT(k)
imaiorRCT = k
End If
k = k + 1
Loop
"MsgBox (maiorR)
"MsgBox (imaiorRCT)
Cells(lin + 2 + seqmach(imaiorRCT), col +
(imaiorRCT - 1) * 3).Value = imaior
cor = Cells(2, 2 + imaior).Interior.ColorIndex
Cells(lin + 2 + seqmach(imaiorRCT), col +
(imaiorRCT - 1) * 3).Interior.ColorIndex = cor
Cells(lin + 2 + seqmach(imaiorRCT), col +
(imaiorRCT - 1) * 3).HorizontalAlignment = xlCenter
Cells(lin + 2 + seqmach(imaiorRCT), col + 1 +
(imaiorRCT - 1) * 3).Value = maior
Cells(lin + 2 + seqmach(imaiorRCT), col + 1 +
(imaiorRCT - 1) * 3).HorizontalAlignment = xlCenter
dispRCT(imaiorRCT) = dispRCT(imaiorRCT) - maior
Cells(lin + 2 + seqmach(imaiorRCT), col + 2 +
(imaiorRCT - 1) * 3).Value = dispRCT(imaiorRCT)
Cells(lin + 2 + seqmach(imaiorRCT), col + 2 +
(imaiorRCT - 1) * 3).HorizontalAlignment = xlCenter
If imaiorRCT = 1 Then
Cells(lin + 2 + seqmach(imaiorRCT), col + 2 +
(imaiorRCT - 1) * 3).Borders(xlEdgeRight).LineStyle =
xlContinuous
ElseIf imaiorRCT > 1 And imaiorRCT < nmach Then
Cells(lin + 2 + seqmach(imaiorRCT), col + 2 +
(imaiorRCT - 1) * 3).Borders(xlEdgeRight).LineStyle =
xlContinuous
Cells(lin + 2 + seqmach(imaiorRCT), col +
(imaiorRCT - 1) * 3).Borders(xlEdgeLeft).LineStyle =
xlContinuous
Else
Cells(lin + 2 + seqmach(imaiorRCT), col +
(imaiorRCT - 1) * 3).Borders(xlEdgeLeft).LineStyle =
xlContinuous
End If
seqmach(imaiorRCT) = seqmach(imaiorRCT) + 1
seq = seq + 1
i = nmach + 1

```

```

Else
i = i + 1
End If

Loop
End Sub

Sub Paral_Testes()
Dim maior As Integer
Dim imaior As Integer
Dim col As Integer
Dim lin As Integer

lin = 16
col = 3
TT = 0
i = 1

If seq = 0 Then
Do Until Cells(5, i + 2).Value = False And Cells(6, i +
2).Value = False
ordjobs(1, i) = Cells(5, i + 2).Value
ordjobs(2, i) = Cells(6, i + 2).Value
TT = TT + ordjobs(2, i)
i = i + 1
Loop
njobs = i - 1
i = 1
Do Until i = nmach + 1
dispRCT(i) = RCT
i = i + 1
Loop
End If

nmach = Range("C9")

RCT = TT / nmach

Do Until seq = njobs
i = 1

Do Until i + seq >= njobs + 1
If i = 1 Then
imaior = ordjobs(1, i + seq)
maior = ordjobs(2, i + seq)
ElseIf ordjobs(2, i + seq) > maior Then
imaior = ordjobs(1, i + seq)
maior = ordjobs(2, i + seq)
End If
i = i + 1
Loop

i = 1
"Do Until seq = njobs

Do Until i > nmach

If maior <= dispRCT(i) And seq + 1 < njobs Then
Cells(lin + 2 + seqmach(i), col + (i - 1) * 3).Value =
imaior
cor = Cells(2, 2 + imaior).Interior.ColorIndex
Cells(lin + 2 + seqmach(i), col + (i - 1) *
3).Interior.ColorIndex = cor
Cells(lin + 2 + seqmach(i), col + (i - 1) *
3).HorizontalAlignment = xlCenter
Cells(lin + 2 + seqmach(i), col + 1 + (i - 1) *
3).Value = maior

```

```

Cells(lin + 2 + seqmach(i), col + 1 + (i - 1) *
3).HorizontalAlignment = xlCenter
dispRCT(i) = dispRCT(i) - maior
Cells(lin + 2 + seqmach(i), col + 2 + (i - 1) *
3).Value = dispRCT(i)
Cells(lin + 2 + seqmach(i), col + 2 + (i - 1) *
3).HorizontalAlignment = xlCenter
If i = 1 Then
Cells(lin + 2 + seqmach(i), col + 2 + (i - 1) *
3).Borders(xlEdgeRight).LineStyle = xlContinuous
ElseIf i > 1 And i < nmach Then
Cells(lin + 2 + seqmach(i), col + 2 + (i - 1) *
3).Borders(xlEdgeRight).LineStyle = xlContinuous
Cells(lin + 2 + seqmach(i), col + (i - 1) *
3).Borders(xlEdgeLeft).LineStyle = xlContinuous
Else
Cells(lin + 2 + seqmach(i), col + (i - 1) *
3).Borders(xlEdgeLeft).LineStyle = xlContinuous
End If
seqmach(i) = seqmach(i) + 1
seq = seq + 1
i = nmach + 1
ElseIf i = nmach And seq < njobs Then
k = 1
Do Until k >= nmach + 1
If k = 1 Then
maiorRCT = dispRCT(k)
imaiorRCT = k
ElseIf maiorRCT < dispRCT(k) Then
maiorRCT = dispRCT(k)
imaiorRCT = k
End If
k = k + 1
Loop
"MsgBox (maiorR)
"MsgBox (imaiorRCT)
Cells(lin + 2 + seqmach(imaiorRCT), col +
(imaiorRCT - 1) * 3).Value = imaior
cor = Cells(2, 2 + imaior).Interior.ColorIndex
Cells(lin + 2 + seqmach(imaiorRCT), col +
(imaiorRCT - 1) * 3).Interior.ColorIndex = cor
Cells(lin + 2 + seqmach(imaiorRCT), col +
(imaiorRCT - 1) * 3).HorizontalAlignment = xlCenter
Cells(lin + 2 + seqmach(imaiorRCT), col + 1 +
(imaiorRCT - 1) * 3).Value = maior
Cells(lin + 2 + seqmach(imaiorRCT), col + 1 +
(imaiorRCT - 1) * 3).HorizontalAlignment = xlCenter
dispRCT(imaiorRCT) = dispRCT(imaiorRCT) -
maior
Cells(lin + 2 + seqmach(imaiorRCT), col + 2 +
(imaiorRCT - 1) * 3).Value = dispRCT(imaiorRCT)
Cells(lin + 2 + seqmach(imaiorRCT), col + 2 +
(imaiorRCT - 1) * 3).HorizontalAlignment = xlCenter
If imaiorRCT = 1 Then
Cells(lin + 2 + seqmach(imaiorRCT), col + 2 +
(imaiorRCT - 1) * 3).Borders(xlEdgeRight).LineStyle =
xlContinuous
ElseIf imaiorRCT > 1 And imaiorRCT < nmach
Then
Cells(lin + 2 + seqmach(imaiorRCT), col + 2 +
(imaiorRCT - 1) * 3).Borders(xlEdgeRight).LineStyle =
xlContinuous
Cells(lin + 2 + seqmach(imaiorRCT), col +
(imaiorRCT - 1) * 3).Borders(xlEdgeLeft).LineStyle =
xlContinuous
Else
Cells(lin + 2 + seqmach(imaiorRCT), col +
(imaiorRCT - 1) * 3).Borders(xlEdgeLeft).LineStyle =
xlContinuous
End If
seqmach(imaiorRCT) = seqmach(imaiorRCT) + 1
seq = seq + 1
i = nmach + 1
Else
i = i + 1
End If
Loop
Loop
End Sub

```