



Mestrado em Engenharia e Gestão Industrial

---

**Otimização de processos logísticos na Roca Leiria:  
desenvolvimento de um *software* baseado em filas  
de espera**

Relatório de Estágio para a obtenção do grau de Mestre em Engenharia e  
Gestão Industrial

**Autor**

**Daniela Patrícia Silva Neves**

**Orientador**

**Professor Jorge Alexandre Caldeira Gonçalves de Almeida**

Instituto Superior de Engenharia de Coimbra

**Supervisor**

**Eng<sup>o</sup> Vitor Alexandre Ferreira Rodrigues**

Roca, S.A

**Coimbra, junho 2017**



*“O Homem é do tamanho do seu sonho.”*

Fernando Pessoa



## Agradecimentos

A realização do estágio curricular no âmbito do Mestrado em Engenharia e Gestão Industrial foi um marco para mim, não só profissionalmente, mas também a nível emocional. Este desafio permitiu-me crescer numa realidade diferente, no qual foram postos à prova não só conhecimentos, como a atitude face ao mundo de trabalho.

Contei com a confiança e ajuda de inúmeras pessoas, não apenas por parte da empresa, mas também da família, amigos e de docentes que fizeram parte do meu percurso académico no Instituto Superior de Engenharia de Coimbra. Estou certa de que sem a paciência, a compreensão e a amabilidade tal não seria possível.

Quero agradecer ao Professor Jorge Almeida, pela orientação, apoio, disponibilidade e auxílio nos problemas e dúvidas que foram surgindo, e pelo total incentivo na realização desta dissertação.

Ao Engenheiro Vítor Rodrigues, supervisor na empresa Roca, S.A, agradeço o apoio, a partilha do conhecimento e a valiosa contribuição para o meu trabalho, prescindindo do seu tempo para responder às minhas necessidades. Obrigada por estimular o meu interesse pelo conhecimento, pela melhoria contínua e por continuar a acompanhar-me nesta nova etapa.

Nas instalações da Roca, S.A, pude participar de um grupo íntegro de colegas, o que representou uma oportunidade ímpar de crescimento académico e também pessoal. A todos, obrigada pela integração e pelo carinho.

Aos meus pais, à minha irmã e ao Luís, obrigada pela paciência e pela força. Sem vocês, nada disto seria possível.

Mais uma vez, sou grata a todos os meus familiares, aos meus amigos, à loja de artigos do Santuário de Fátima, pelo incentivo que recebi ao longo do meu percurso. O meu profundo agradecimento a todas as pessoas que contribuíram para a realização deste relatório, e que estimularam a minha pessoa de forma intelectual e emocional.



## Resumo

No cenário atual, o setor industrial necessita, cada vez mais, de maximizar a sua produção e os seus fluxos, sem sofrer perdas, e com isso, minimizar os custos associados. A otimização surge como uma potencial solução.

A crescente procura de metodologias de otimização de rotas e de processos, no tecido fabril, permitiu que a simulação computacional ganhasse um lugar de destaque. O conceito de simulação agregado à modelação permite o estudo de um sistema real. Em virtude desse estudo, é possível a melhoria dos seus *outputs*, sem efetuar alterações no sistema real.

Esta dissertação enquadra um estudo, dentro de uma empresa, sobre a Teoria de Filas de espera e o caminho mais curto, na ótica da simulação computacional. Relaciona-se com a capacidade de produção e com os fluxos logísticos.

De um modo geral, no âmbito do estágio curricular na empresa Roca, S.A., o principal foco foi a otimização de processos e fluxos logísticos. Como tal, recorreu-se à elaboração e ao desenvolvimento de um *software* em *Matlab*. Numa primeira fase efetuou-se um estudo ao estado atual do sistema, e numa fase posterior analisaram-se diferentes e possíveis cenários. Este *software* permitiu a otimização da cadeia de processo, baseando-se em Teoria de Filas de espera e no problema do trajeto mais curto.

Palavras-chave: otimização, simulação, trajeto mais curto, filas de espera, processos logísticos.



## Abstract

In the current scenario, the industrial sector increasingly needs to maximize production and flows without suffering losses. Optimization emerges as a potential solution to minimize the associated costs.

The increasing demand for route and process optimization methodologies in the industry allowed the computational simulation to gain a prominent place. The concept of simulation added to the modeling allows the study of a certain real system. By virtue of this study, it is possible to improve its outputs without making changes in the real system.

The main issue of this dissertation is a study of queues, as well as the shortest paths, from the point of view of computational simulation. It is directed to the organizational aspect and relates to the capacity of production and the flows inherent to it.

The main focus of this internship at Roca, S.A was the optimization of industry processes and logistic flows. As such, using software developed with Matlab. In a first phase, a study was made to understand the current state of the system and then different and possible scenarios were analyzed. This *software* allowed the optimization of the process chain, based on Theory of Queues and the problem of the shortest path.

Keywords: optimization, simulation, shortest path, logistic flow, queues.



# Índice

1.	Introdução .....	1
1.1	Principais objetivos e metodologias.....	1
1.2	Organização da tese.....	2
2.	Enquadramento .....	3
2.1	A História da Roca Corporação Empresarial, S.A. – Um negócio Global.....	3
2.1.1	Resultados económicos do ano de 2014.....	5
2.1.2	A Roca em Portugal .....	6
2.1.3	A Roca, S.A. – Leiria .....	7
2.1.4	Descrição do processo produtivo .....	11
2.2	O estado da arte.....	24
2.2.1	Definição de Simulação .....	24
2.2.2	As características de um modelo de simulação.....	25
2.2.3	As vantagens da simulação .....	26
2.2.4	As desvantagens da simulação .....	28
2.2.5	O desenvolvimento de um modelo de simulação.....	29
2.2.6	As áreas de aplicação da simulação .....	33
2.2.7	A simulação na indústria.....	34
3.	Desenvolvimento .....	37
3.1	A Teoria de Filas de Espera .....	37
3.1.1	A estrutura de um modelo de filas de espera .....	38
3.1.2	A terminologia e a notação utilizada.....	40
3.1.3	As relações fundamentais entre L, W, Lq e Wq.....	41
3.1.4	A distribuição das chegadas e do atendimento.....	42

3.1.5	A caracterização da distribuição exponencial .....	42
3.1.6	O modelo M/M/S .....	47
3.1.7	O modelo M/M/S (s=1).....	48
3.1.8	O modelo M/M/S (s>1).....	49
3.2	Algoritmo de Dijkstra .....	52
3.3	O modelo proposto.....	56
3.3.1	Caraterização da fila de espera.....	60
3.3.2	Caraterização do percurso .....	61
3.4	Recolha de dados .....	62
3.5	Dados para a caracterização da fila de espera e do percurso.....	62
4.	Implementação.....	77
4.1	Tratamento de dados .....	77
5.	Demonstração e discussão de resultados .....	85
5.1	A fila de espera .....	85
5.2	O trajeto mais curto.....	91
6.	Conclusão .....	99
	Bibliografia.....	103
	Anexo 1 .....	107
	Anexo 3.....	135
	Anexo 4.....	139
	Anexo 5.....	143
	Anexo 6.....	145

## Índice de Figuras

Figura 1 – Instalações da Roca, S.A. ....	8
Figura 2 – Etapas do processo produtivo da Roca, S.A. ....	11
Figura 3 – Passadeira com lavatórios para posterior inspeção. ....	13
Figura 4 – Mesa de medição de lavatórios. ....	13
Figura 5 – Elevador contendo uma paleta de peças de porcelana sanitária. ....	15
Figura 6 – Elevador observado da secção “Fornos”. ....	15
Figura 7 – Paleta à espera para ser inspecionada e embalada. ....	16
Figura 8 – Inspeção de tanques de porcelana sanitária na passadeira um. ....	16
Figura 9 – Inspeção e rastreabilidade de tanques de porcelana sanitária na passadeira um. ....	17
Figura 10 – Bacia de porcelana sanitária na passadeira 2. ....	17
Figura 11 - Bacia de porcelana sanitária na passadeira dois a ser inspecionada. ....	18
Figura 12 – Paleta com lavatórios de porcelana na plataforma. ....	18
Figura 13 – Lavatórios de porcelana na passadeira três. ....	19
Figura 14 – Plataforma com tanques inspecionados. ....	20
Figura 15 – Paletes na passadeira do forno retrátil. ....	20
Figura 16- Colocação de lâmina retrátil na paleta. ....	21
Figura 17 - Colocação de plástico retrátil na paleta. ....	21
Figura 18 – Paletização referente aos tanques de porcelana sanitária. ....	22
Figura 19 - Paletização referente aos lavatórios de porcelana sanitária. ....	22
Figura 20 – Passadeira afeta ao forno retrátil. ....	22
Figura 21 – Forno retrátil. ....	23
Figura 22 – Entrada de peças na passadeira através do robot. ....	24
Figura 23 – Saída de peça da passadeira através do robot. ....	24
Figura 24 – Etapas de um modelo de simulação, segundo (Anu Maria, 1997). ....	29
Figura 25 – Fluxograma do processo que envolve a Simulação, segundo Banks (Banks, 1998). ....	32
Figura 26 – Estrutura básica de um sistema de Fila de espera (fonte: (Hillier & Lieberman, 1990). ....	38
Figura 27 – Gráfico da função densidade para a distribuição exponencial negativa (Fonte: (Hillier & Lieberman, 1990)) ....	43
Figura 28 – Diagrama de estado para o modelo M/M/S, com um único servidor (s=1) (adaptado: (Hillier & Lieberman, 1990)) ....	48

Figura 29 - Diagrama de estado para o modelo M/M/S, com s servidores (adaptado: (Hillier & Lieberman, 1990)).	50
Figura 30 – Fluxograma do percurso da Sanita.	57
Figura 31 – Fluxograma do percurso do lavatório.	58
Figura 32 – Fluxograma do percurso da tanque.	59
Figura 33 – Zona de <i>stock</i> das peças (antes de inspeção).	60
Figura 34 – Cadência de chegada de peças de porcelana ao departamento Escolha Final.	63
Figura 35 – Saída gráfica do ERP SAP.	64
Figura 36 – Dados das peças conformes, em Janeiro, Fevereiro, Março e Abril.	64
Figura 37 – Fluxograma do percurso do produto tanque.	70
Figura 38 – Fluxograma do percurso do produto sanita.	72
Figura 39 – Fluxograma do percurso do produto sanita.	75
Figura 40 – Desenvolvimento do layout da interface gráfica que corresponde à ferramenta de simulação computacional.	78
Figura 41 – <i>Layout</i> da interface gráfica que corresponde à ferramenta de simulação computacional.	79
Figura 42 – <i>Layout</i> da interface gráfica que corresponde à seleção de dados históricos, para a caracterização da fila de espera.	80
Figura 43 – <i>Layout</i> da interface gráfica que corresponde à seleção do tipo de produto, para a caracterização da fila de espera.	80
Figura 44 – <i>Layout</i> da interface gráfica que corresponde à seleção de dados de uma determinada data, para a caracterização da fila de espera.	81
Figura 45 – Seleção do ficheiro em Microsoft Excel (Dados.xlsx).	82
Figura 46 – <i>Layout</i> da interface gráfica que corresponde à parametrização da fila de espera.	82
Figura 47 – <i>Layout</i> da interface gráfica que corresponde à caracterização da fila de espera.	83
Figura 48 – <i>Layout</i> da interface gráfica que corresponde à caracterização do percurso.	84
Figura 49 – <i>Layout</i> da interface gráfica que corresponde à obtenção do trajeto mais curto de cada peça.	84
Figura 50 – Parametrização do dia 1 de abril de 2016, para o produto sanita.	86
Figura 51 – Medidas de desempenho obtidas do dia 1 de abril de 2016, para o produto Sanita.	87
Figura 52 – Trajeto mais curto, do produto Sanita.	93
Figura 53 – Trajeto mais curto, do produto tanque.	95
Figura 54 – Trajeto mais curto, do produto tanque.	97
Figura 55 – Percentagem de peças boas embaladas durante o mês de janeiro.	108
Figura 56 – Percentagem de peças boas embaladas durante o mês de fevereiro.	109
Figura 57 – Percentagem de peças boas embaladas durante o mês de março.	110

Figura 58 – Percentagem de peças boas embaladas durante o mês de abril. ....	111
Figura 59 – Percentagem do tipo de produto sanita, embalado durante o mês de janeiro.....	112
Figura 60 – Percentagem do tipo de produto sanita, embalado durante o mês de fevereiro. ....	113
Figura 61 – Percentagem do tipo de produto sanita, embalado durante o mês de março.....	114
Figura 62 – Percentagem do tipo de produto sanita, embalado durante o mês de abril.....	115
Figura 63 – Percentagem do tipo de produto lavatório, embalado durante o mês de janeiro. ....	116
Figura 64 – Percentagem do tipo de produto lavatório, embalado durante o mês de fevereiro.....	117
Figura 65 – Percentagem do tipo de produto lavatório, embalado durante o mês de março. ....	118
Figura 66 – Percentagem do tipo de produto lavatório, embalado durante o mês de abril.....	119
Figura 67 – Percentagem do tipo de produto tanque, embalado durante o mês de janeiro. ....	120
Figura 68 – Percentagem do tipo de produto tanque, embalado durante o mês de fevereiro. ....	121
Figura 69 – Percentagem do tipo de produto tanque, embalado durante o mês de março.....	122
Figura 70 – Percentagem do tipo de produto tanque, embalado durante o mês de abril. ....	123



## Índice de tabelas

Quadro 1 – Empresas constituintes do Grupo Roca (Roca, 2016). .....	4
Quadro 2 – Simbologia e respetiva descrição da teoria de filas de espera (Hillier & Lieberman, 1990).....	41
Quadro 3 – Tipo de produto e posto de trabalho correspondente.....	56
Quadro 4 – Definição das zonas de trabalho. ....	61
Quadro 5 - Correspondência de cada posto de trabalho para o tipo de peça a produzir. ....	61
Quadro 6 – Definição e descrição dos nós, para o produto Sanita. ....	63
Quadro 7 – Dados relativos às peças boas, correspondentes ao tipo de produto sanita. ....	65
Quadro 8 – Dados relativos às peças boas, correspondentes ao tipo de produto tanques.....	66
Quadro 9 – Dados relativos às peças boas, correspondentes ao tipo de produto lavatórios. ....	67
Quadro 10 – Definição e descrição dos nós, para o produto tanque.....	68
Quadro 11 – Distância entre os diferentes nós, para o produto tanque. ....	68
Quadro 12 – Dados referentes ao Cenário Atual, Cenário 1 e Cenário 2 do percurso do tanque.....	69
Quadro 13 – Distância Total, para cada cenário, do produto Tanque. ....	69
Quadro 14 – Definição e descrição dos nós, para o produto Sanita. ....	70
Quadro 15 – Distância entre os diferentes nós, para o produto sanita.....	71
Quadro 16 – Dados referentes ao Cenário Atual, Cenário 1 e Cenário 2 do percurso da Sanita.....	71
Quadro 17 – Distância Total, para cada cenário, do produto Sanita. ....	72
Quadro 18 – Definição e descrição dos nós, para o produto Lavatório.....	73
Quadro 19 – Distância entre os diferentes nós, para o produto lavatório. ....	73
Quadro 20 – Dados referentes ao Cenário Atual, Cenário 1 e Cenário 2 do percurso do lavatório. ....	74
Quadro 21 – Distância Total, para cada cenário, do produto lavatório. ....	74
Quadro 22 – Parte integral de código em Matlab para obter o tempo de chegada. ....	82
Quadro 23 – Parte integral de código em Matlab correspondentes à caracterização da fila de espera. ....	83
Quadro 24 – Obtenção de cinco medidas de desempenho de fila de espera do produto sanita, para 10 dias..	85
Quadro 25 – Obtenção de seis medidas de desempenho de fila de espera do produto Sanita, para dez dias. .	86
Quadro 26 – Obtenção de cinco medidas de desempenho de fila de espera do produto tanque, para dez dias. .....	88
Quadro 27 – Obtenção de seis medidas de desempenho de fila de espera do produto Tanque, para 10 dias..	89
Quadro 28 – Obtenção de cinco medidas de desempenho de fila de espera do produto Lavatório, para 10 dias. .....	90

Quadro 29 – Obtenção de seis medidas de desempenho de fila de espera do produto Lavatório, para 10 dias. .....	91
Quadro 30 – Distância Total, para cada cenário, do produto sanita. ....	92
Quadro 31 – Distância Total, para cada cenário, do produto Sanita, em função do número de peças embaladas. .....	92
Quadro 32 – Diferença obtida entre o Estado Atual e o Trajeto mais curto, do produto sanita. ....	93
Quadro 33 – Distância Total, para cada cenário, do produto tanque. ....	94
Quadro 34 – Distância Total, para cada cenário, do produto tanque, em função do número de peças embaladas. .....	94
Quadro 35 – Diferença obtida entre o Estado Atual e o Trajeto mais curto, do produto tanque. ....	95
Quadro 36 – Distância Total, para cada cenário, do produto lavatório. ....	96
Quadro 37 – Distância Total, para cada cenário, do produto lavatório, em função do número de peças embaladas. ....	96
Quadro 38 – Diferença obtida entre o Estado Atual e o Trajeto mais curto, do produto lavatório. ....	97

## Simbologia

$\lambda_n$  – Ritmo médio de chegadas de novos clientes/peças por unidade de tempo.

$\frac{1}{\lambda}$  – Tempo esperado entre chegadas de novos clientes/peças.

$\mu_n$  – Ritmo médio de serviço global do sistema – n° médio de clientes/peças que terminam o serviço por unidade de tempo.

$\mu_n = s\mu$ ,  $n \geq s$  (Todos os servidores estão ocupados).

$\frac{1}{\mu}$  - Tempo esperado de serviço.

$\rho = \frac{\lambda}{s\mu}$  - Taxa de ocupação (fator de utilização da infraestrutura de serviço).

$s$  – Número de servidores.

$P_n$  – Probabilidade de estarem exatamente  $n$  clientes/peças no sistema.

$L$  – Número de clientes no sistema.

$L_q$  – Comprimento da fila de espera (excluindo as peças que estão a ser servidas na cinta).

$W$  – Tempo de espera no sistema (inclui tempo de serviço) para cada cliente/peça.

$W_q$  – Tempo de espera no sistema (exclui tempo de serviço) de cada cliente



# 1. Introdução

A presente dissertação descreve as atividades desenvolvidas no âmbito do estágio curricular realizado na Empresa Roca S.A, em Leiria. O estágio curricular teve duração de oito meses e decorreu na secção Escolha Final. O objetivo desta secção é a triagem e verificação de peças conformes e respetiva embalagem, para posteriormente serem enviadas ao cliente final.

Este estágio teve como foco a análise e a otimização da secção de embalagem e paletização – Escolha Final. Em virtude da realização deste estágio, o objetivo foi essencialmente a elaboração e o desenvolvimento de um *software* para a cadeia de produção, baseado na Teoria de Filas de espera. Através deste *software*, pretendeu-se a otimização dos processos e fluxos logísticos desta secção da Roca Leiria, para que num breve horizonte temporal o mesmo seja validado e implementado.

## 1.1 Principais objetivos e metodologias

Um dos principais objetivos e motivação da realização deste estágio curricular na Empresa Roca, S.A., foi o enriquecimento pessoal e o desenvolvimento de competências. O estágio curricular é uma importante componente na formação académica e profissional que permite o conhecimento da realidade industrial e a integração dentro de uma equipa de trabalho. Pretendeu-se, portanto, a adaptação de conteúdos teóricos em problemas reais dentro de um contexto de trabalho.

Ao nível de conteúdo programático e plano de estudos, no seguimento deste estágio, pretenderam-se adquirir capacidades de resolução de problemas complexos, como os elevados *stocks* analisados e o *layout* do armazém deste departamento, que necessita de um plano de ações de melhoria. Como uma das grandes necessidades da empresa é a otimização e melhoria do fluxo do processo, decidiu-se efetuar um estudo ao nível da Teoria de Filas de espera e dos percursos das peças.

Para a realização deste relatório, a metodologia de abordagem foi a seguinte:

- Pesquisa e estudo bibliográfico;
- Observação e estudo do modo de funcionamento do departamento Escolha Final;
- Estudo e acompanhamento do percurso de peças, desde a secção dos “Fornos” até à secção da Escolha Final;
- Estudo de modelação matemática aplicável ao caso de estudo;
- Estudo de modelos comerciais existentes;
- Aplicação a nível computacional, do estudo de modelação matemática realizado no departamento.

## 1.2 Organização da tese

A presente tese encontra-se organizada em 7 capítulos, que abrange:

No Capítulo Um apresenta-se a introdução do trabalho, nomeadamente os principais objetivos, as principais metodologias e a motivação para a realização deste estágio curricular.

No Capítulo Dois é apresentado o enquadramento do trabalho, nomeadamente o enquadramento da Empresa Roca, no Mundo e em Portugal, sendo a empresa Roca um negócio global. Este capítulo compreende o processo produtivo da empresa e a descrição da secção Escolha Final, departamento onde foi realizado o estágio curricular. Por outro lado, é apresentado o Estado da Arte, tendo como foco a Simulação, a sua importância e como esta é aplicada, atualmente, nas mais diversas áreas e na Indústria. Por último, são abordados os principais objetivos e metodologias necessárias à realização deste trabalho.

No Capítulo Três é apresentado o desenvolvimento do trabalho. Este compreende a Teoria de Filas de Espera, aplicada à secção a analisar. É enunciado e descrito o modelo referente ao departamento Escolha Final, são analisados e estudados os dados deste e posteriormente, a aplicação do algoritmo de *Dijkstra* ao processo logístico.

No Capítulo Quatro é descrito o desenvolvimento do modelo em *Matlab*. Apresenta-se não só como foi desenvolvida esta implementação, mas também, como foi realizado o tratamento de dados no seguimento desta implementação.

No Capítulo Cinco são demonstrados os resultados da implementação e a sua posterior análise e discussão.

No Capítulo Seis são apresentadas as principais conclusões do desenvolvimento do trabalho, assim como propostas de melhoria, em diferentes cenários.

Finalmente é apresentada a Bibliografia, que traduz o suporte para a análise, o estudo e a concretização deste trabalho efetuado durante o estágio curricular.

## 2. Enquadramento

A Roca é uma empresa multinacional espanhola, com sede em Barcelona, líder de mercado, que produz e comercializa uma vasta gama de produtos para espaços de banho e instalações sanitárias. Apresenta soluções de lavatórios e móveis, torneiras, banheiras e duchas, acessórios, complementos e divisórias, bem como sistemas de instalação (Roca, 2016).

A missão da Roca visa a oferta de uma grande diversidade de produtos cerâmicos, dotados de inovação e qualidade superior, de forma a pensar no máximo conforto e bem-estar dos clientes e assim, satisfazer as suas necessidades. O objetivo crucial da organização é alcançar e manter a liderança mundial, no setor cerâmico, sendo uma referência na qualidade e no desenvolvimento dos seus produtos, no serviço ao cliente e na preocupação pelo meio ambiente (Roca, 2016).

De um modo geral, a Roca privilegia os seguintes aspetos:

- Inovação contínua ao nível dos seus processos e produtos;
- Elevada qualidade do produto e levar, deste modo, a satisfação aos clientes nacionais e internacionais;
- Formação contínua de todos os colaboradores (Roca, 2016).

### 2.1 A História da Roca Corporação Empresarial, S.A. – Um negócio Global

Em 1917, a Roca Corporação Empresarial, S.A começou a sua atividade em Gavà – província de Barcelona – e o seu foco seria a produção de radiadores de ferro fundido para aquecedores domésticos. No entanto, dada a rápida introdução de novos produtos no mercado e o desejo crescente de expansão, orientou-se para outras áreas de negócio, com o intuito de diversificar as suas atividades.

Assim, em 1925 foram produzidas as primeiras caldeiras de fundição e em 1929 iniciou-se a produção de banheiras de ferro fundido.

Em 1936, entrou no mercado de louças sanitárias e em 1954 iniciou a fabricação de metais sanitários. Enquanto isso, em 1962 foi construída uma segunda fábrica, em Alcalá de Henares (Madrid).

Em 1963, a Roca transforma-se numa das primeiras empresas de Espanha a produzir e comercializar sistemas de ar-condicionado, com a inauguração de uma fábrica em Sabadell.

Em 1968 foi construída a terceira fábrica de porcelana, em Alcalá de Guadaira (Sevilha), e em 1974 uma fábrica de banheiras de aço.

Em janeiro de 2002, o Grupo Roca inicia um processo de reestruturação nas empresas, no sentido de dividir as diferentes áreas de negócio em empresas independentes. Esta estratégia conduziu à criação da Roca Corporação Empresarial, S.A, que compreende quatro empresas, uma para cada uma das áreas de negócio: revestimentos cerâmicos, aquecedores, ar-condicionado e porcelana sanitária.

Na década de noventa do século passado, o Grupo Roca inicia uma fase de expansão internacional. Além de Portugal e França, a presença comercial começa a difundir-se por países como o Reino Unido, Alemanha, Itália, Rússia, Turquia, Marrocos, Argentina, Brasil e China. O ponto de viragem no processo de internacionalização ocorre em 1999, com a aquisição da empresa suíça Keramik Holding Laufen.

Com a compra da empresa Laufen, o Grupo reforça a sua presença em mercados estratégicos em que tinha menor influência como a Europa de Leste, o Brasil e os Estados Unidos da América. Em 2005 alcança a liderança através de um plano estratégico com foco em espaços de banho.

Nos últimos anos, apresenta uma notável expansão internacional, com atividade em 135 mercados. Possui 69 unidades de produção em quatro continentes, sendo constituída por cerca de 20000 colaboradores (Roca, 2016).

Quadro 1 – Empresas constituintes do Grupo Roca (Roca, 2016).

	<b>Localização</b>	<b>Empresa</b>
<b>Alemanha</b>	Staudt	Roca GmbH
<b>Argentina</b>	Buenos Aires	Roca Argentina, S.A.
<b>Austrália</b>	Sydney	Roca Bathroom Products Australia Pty Ltd
	Botany	Gemini Industries Pty Ltd
<b>Austria</b>	Wilhelmsburg	Laufen Austria AG
<b>Brasil</b>	Campo Largo-Paraná	Incepa Revestimentos Cerâmicos Ltda
	Jundiaí-São Paulo	Roca Sanitários Brasil Ltda
<b>Bulgária</b>	Kaspichan	Roca Bulgaria AD
<b>China</b>	Jiangmen	Jiangmen Giessdorf Bathroom Products Co. Ltd.
	Foshan	Roca (China), Co. Ltd.
	Foshan	Xinle Bathroom Products (Foshan), Co. Ltd.
	Foshan	Cosmic Bathroom Accessories, Co. Ltd
	Shanghai	Roca Sanitaryware Trading, Co. Ltd.
	Suzhou	Roca Sanitaryware (Suzhou), Co. Ltd.
	Tangshan	Tangshan Ying Bathroom Products, Co. Ltd.
<b>Croácia</b>	Zapresic	Inker, Indstrija Keramike i Porculana D.D.
<b>Dinamarca</b>	Køge	Laufen Nordic ApS
<b>Egipto</b>	Cidade 6 de Outubro	National industrial & Trading Co “Gravena” S.A.E.
	Cidade 6 de Outubro	National Sanitaryware Company S.A.E.
<b>Emiratos Árabes Unidos</b>	Dubai	Roca Sanitario, S.A. (R.O.)
<b>Eslováquia</b>	Bratislava	Laufen SK s.r.o.

<b>EUA</b>	Miami/Florida	Roca USA Inc. Keramika
<b>Rússia</b>	Cheboksary Davidovo Kaluga Tosno	Aquaton Rus Ugrakeram Roca Rus
<b>França</b>	Paris	Roca S.A.R.L.
<b>Grécia</b>	Atenas	Roca Hellas, A.E.E.
<b>Holanda</b>	Eindhoven	Laufen Benelux B.V.
<b>Hong Kong</b>	Hong Kong Hong Kong	Roca Asia Ltd. Roca Sanitaryware Ltd.
<b>Índia</b>	Chennai	Roca Bathroom Products Pvt Ltd.
<b>Itália</b>	Milán	Laufen Italia S.R.L.
<b>Malásia</b>	Selangor Darul Ehsan Selangor Darul Ehsan Selangor Darul Ehsan	Roca Malaysia Sdn Bhd Johnson Suisse Sdn Bhd Econax Sdn Bhd
<b>Marrocos</b>	Settat	Roca Maroc, S.A.
<b>Noruega</b>	Drammen	VVS Marketing AS
<b>Polónia</b>	Gliwice Gliwice	ZWS Silesia, Sp. z.o.o. Roca Polska, Sp. z.o.o.
<b>Portugal</b>	Águeda Anadia Cantanhede Leiria	BLB – Indústrias Metalúrgicas, S.A Sanitana, S.A. Roca Torneiras, Lda. Roca, S.A.
<b>Reino Unido</b>	Birmingham Worcestershire	Roca, Ltd. Laufen, Ltd.
<b>República Checa</b>	Praga	Laufen CZ, s.r.o.
<b>Roménia</b>	Bucarest	Roca Obiecte Sanitare SRL.
<b>Singapura</b>	Singapur	Roca Bathroom Products Singapore Pte Ltd
<b>Suiça</b>	Laufen Laufen Laufen Laufen	Keramik Holding AG Keramik Laufen AG Roca Finance AG Similor AG
<b>Tailândia</b>	Changhua	Roca Taiwan Co Ltd

### 2.1.1 Resultados económicos do ano de 2014

No ano de 2014, a economia mundial cresceu 3,4%, sendo que os principais mercados, os Estados Unidos de América e a China cresceram a uma taxa de 2,4% e de 7,4% respetivamente. Pelo contrário, os mercados emergentes e em desenvolvimento contribuíram para o crescimento mundial, mas a um ritmo inferior comparativamente com o ano de 2013. Com uma evolução lenta, sustentada pela diminuição de preços do petróleo, a zona euro depara-se com uma política monetária menos rígida, mas com um grau de incerteza em relação ao mercado de trabalho. É de notar que no ano de 2014, verificou-se uma valorização do Euro face a outras moedas, especialmente em países com grande influência nos resultados do Grupo Roca.

Neste panorama de incerteza, os dados económicos mais relevantes de 2014, do Grupo Roca foram:

- O volume de negócios consolidado foi de 1.646 milhões de euros, que traduz um aumento de 4,7% em relação ao ano anterior.
- O EBITDA consolidado, que reflete o resultado operacional antes de juros, impostos, depreciação e amortização, foi de 221 milhões de euros e representa 13,4% do volume de negócios.
- O resultado líquido, depois de serem imputados os impostos, foi de 53 milhões de euros, montante similar ao do ano anterior.
- Os capitais próprios do grupo foram de 1.216 milhões de euros no final do ano de 2014. Por sua vez, a dívida financeira líquida em 31 de dezembro de 2014 foi de 378 milhões de euros.
- Os investimentos em ativos tangíveis e intangíveis, durante este ano, foram de 122 milhões de euros. Este facto traduz um aumento face ao ano de 2013 em que foram investidos 80 milhões de euros.

Durante o ano 2014, os principais investimentos aplicados foram a instalação de uma nova linha de produção de cerâmica plana, no Brasil, e o aumento de capacidade de produção nas fábricas localizadas na Índia, na Malásia e na China, no sentido de atender às necessidades dos seus respetivos mercados.

Para além disso, o Grupo Roca aumentou a sua presença comercial em mercados no norte da Europa e na Austrália e procedeu à aquisição de um fornecedor na Índia.

Em Portugal, durante o ano de 2014, verificou-se um crescimento económico em 0,9% e uma redução gradual da taxa de desemprego. O mercado imobiliário continuou a decrescer, embora de forma menos acentuada relativamente a anos anteriores. Apesar da estagnação do setor da construção, o Grupo mantém um aumento nas vendas, resultado esse do mercado interno, assim como das exportações para países africanos de língua Portuguesa, e para o Médio Oriente (Roca, 2016).

### **2.1.2 A Roca em Portugal**

Em 1972, a política de expansão do Grupo Roca catalisou a afirmação da Roca em Portugal – Roca Portugal Lda. – através da produção de equipamentos para aquecimento e de banheiras de ferro fundido (Roca, 2016).

Em 1986, o Grupo cresce e aumenta a sua capacidade de produção com a aquisição da fábrica da Madalena, em Leiria (Roca, 2016).

Em 1995, a fábrica de banheiras de aço esmaltado, em Águeda, é inaugurada e em 1996 foi construída uma segunda unidade de produção em Leiria.

A inovação aliada às novas tecnologias são vistas pelo Grupo Roca, como fatores fundamentais para a qualidade dos seus produtos. Nesse sentido, foi inaugurada, em 1999, a unidade de produção em Cantanhede.

Esta seria considerada uma das unidades europeias com o equipamento tecnológico mais avançado e moderno.

Em 2001, foi inaugurado o Centro de formação para apoio e formação a profissionais do setor.

Dado o seu compromisso com o mercado na oferta de produtos com qualidade, a Roca continuou a manter a sua liderança num mercado em constante evolução, com grande ênfase no domínio da qualidade e *design*. Em 2007, lançou no mercado português a marca Laufen. Esta marca oferece uma panóplia de produtos de elevada qualidade para os clientes mais exigentes.

Em 2011, um novo espaço em Lisboa é inaugurado – o Roca Lisboa Gallery. Este espaço situa-se na Praça dos Restauradores e apresenta as últimas novidades em produtos Roca. Por outro lado, também pretende proporcionar a descoberta da história da Marca e a sua importância no mundo dos espaços de banho.

O projeto foi desenvolvido pelo estúdio de *design* de interiores Ferruz Decoradors. Os valores de liderança, *design*, inovação, sustentabilidade e bem-estar estão bem presentes neste espaço.

Atualmente, existem cerca de 100 pontos de venda por todo o país, e 4 unidades fabris em Anadia, Águeda, Cantanhede e em Leiria (Roca, 2016).

### **2.1.3 A Roca, S.A. – Leiria**

No sentido da internacionalização, o Grupo Roca decidiu investir em Portugal, com vista a uma parceria estratégica vantajosa, dada a sua proximidade.

Através da delegação comercial, localizada em Odivelas, foi realizado um estudo do mercado português, analisando-se os catalisadores positivos e negativos para a construção e instalação do novo complexo fabril. A construção da nova unidade fabril foi na zona de Leiria. Para isso, em 1987, o Grupo Roca comprou uma empresa existente e com dificuldades económicas – a Nova Cerâmica da Madalena (N.C.M.).

A Nova Cerâmica da Madalena transformou-se na Roca Cerâmica e Comércio, S.A, integrando o conhecimento técnico, a organização e a gestão que o grupo espanhol projetou ao longo dos últimos anos.

No início da década de 90 do século passado, foi realizada uma reorganização de processos e recursos humanos, investindo-se em novas tecnologias, conferindo uma maior e melhor capacidade de produção ao processo.

Neste enquadramento, o Grupo decidiu catalisar forças e apostar em Portugal, sendo que em 1995, iniciou-se a construção de uma nova fábrica – Leiria 2 – que foi concebida com um *layout* inovador, caracterizando-se por ser tecnologicamente mais desenvolvida.

O complexo fabril fica a ser, deste modo, constituído por duas unidades fabris: Leiria 1 e Leiria 2. A fábrica de Leiria 2 entra em laboração em abril de 1996. A sua atividade iniciou com reduzidas séries de louça sanitária, ao invés da fábrica de Leiria 1 que aumentava a sua diversificação, bem como o processo produtivo.



Figura 1 – Instalações da Roca, S.A.

No ano de 1998 é realizado um aumento de capacidade do processo produtivo na fábrica de Leiria 2, o que permitiu um aumento de diversidade de modelos. Como tal, a empresa investiu na compra de mais dois fornos e à admissão de um número elevado de novos colaboradores.

De forma a dotar a empresa com uma capacidade logística extremamente eficiente e eficaz, procedeu-se à construção de um novo armazém de expedição, no ano de 2001.

Nos últimos anos, a Roca Cerâmica e Comércio, S.A tem apostado nas novas tecnologias, nomeadamente em máquinas de pressão, *robots*, fornos e essencialmente na formação dos seus colaboradores.

Apesar da constante inovação tecnológica, a Roca S.A continua a dar ênfase e a apostar na mão-de-obra, na medida em que existem etapas do processo produtivo de grande complexidade e de difícil automatização. Deste modo, são obrigatoriamente efetuadas de forma manual, conferindo à produção uma qualidade superior.

Em 1996, foi desenvolvido o Departamento de Qualidade, integrando o Controlo de Qualidade até aí existente e deste modo, ampliou a sua área de intervenção. A intervenção deste departamento incidiu no apoio à área comercial, através da análise de reclamações dos clientes, assim como na inspeção diária de todos os modelos Roca existentes.

Em 1998, a Roca obteve a sua primeira certificação como empresa de Qualidade, com a Norma ISO 9002. Por sua vez, em 2003, a Empresa obteve a certificação ISO 9001.

O funcionamento da empresa está organizado de forma a cumprir os requisitos da Norma ISO 9001. De acordo com a norma, estão estipulados procedimentos e instruções de trabalho para os setores da fábrica. Para além disso, existem outras implicações no funcionamento da empresa, nomeadamente:

- O equipamento de medida tem de estar calibrado e operacional;
- É primordial a formação dos colaboradores e a sua integração no posto de trabalho;
- Os produtos necessitam de ser analisados e medidos durante o processo produtivo, com vista à verificação da sua conformidade de acordo com as normas de funcionamento implementadas pela empresa;
- Análise e correção dos defeitos, e por isso é fundamental o registo de ações;
- Implementação de ações corretivas, com o intuito de corrigir situações problemáticas ou não conformidades que surgem.
- Registrar as ações tomadas, de forma a garantir salvaguarda de dados históricos importantes para a melhoria contínua;
- Cumprimento das necessidades dos clientes.

A entidade APICER efetua, todos os anos, uma auditoria, que permite saber se a empresa continua a funcionar corretamente ou não, e se são cumpridos os objetivos e requisitos da norma;

A certificação da Norma ISO 9001:2000 é válida por 3 anos, renovável ou não, tendo em consideração os resultados da auditoria da APICER.

No que concerne ao sistema de gestão ambiental, que está intimamente integrado com a Qualidade, foi definido e implementado uma Política de Qualidade e Ambiente, que reflete determinados e específicos compromissos, tais como:

- Satisfazer as necessidades dos clientes e as suas expectativas, através da melhoria contínua;
- A melhoria contínua que é alcançada através do “fazer bem à primeira”;
- A melhoria contínua do desempenho ambiental e a prevenção da poluição na sua origem;

- Controlar e reduzir os impactos ambientais resultantes da atividade da empresa;
- Prevenir a ocorrência de desastres ambientais;
- Cumprimento com a legislação ambiental em vigor;
- Satisfazer os requisitos das Normas NP EN ISO 9001:2000 e NP EN ISO 14001:2004.

Assim sendo, ao nível ambiental, como foi referido anteriormente, é imprescindível para a empresa a melhoria contínua do desempenho ambiental e a prevenção da poluição na sua origem. Como exemplo de uma medida de prevenção realizada pela empresa é a colocação de filtros de retenção de poeiras, para que estas não sejam expelidas para o meio ambiente externo.

Por outro lado, a empresa dá ênfase ao controlo e redução dos impactos ambientais resultantes da sua atividade, nomeadamente ao nível de resíduos gerados, à utilização de energia e de água, numa perspetiva de racionalização e de melhoria contínua.

As normas de qualidade da Roca são elaboradas pela sede do Grupo Roca em Barcelona e distribuídas a todas as fábricas da Roca existentes. Estas referem dez grupos de requisitos:

- Defeitos de aspeto;
- Medidas;
- Instalação e Montagem;
- Resistência química do vidro;
- Resistência mecânica da peça;
- Resistência térmica (a variações de temperaturas);
- Defeitos de cor;
- Estanquicidade;
- Funcionamento;
- Paletização.

A empresa Roca Cerâmica e Comércio, S.A. compromete-se com os seus colaboradores, fornecedores e clientes a proporcionar todos os meios necessários para assegurar e garantir a saúde e segurança destes. Como tal é necessário cumprir a legislação sobre a segurança e saúde no trabalho em vigor, e aplicável à organização. Do mesmo modo, na prevenção, a empresa compromete-se a ter em consideração os riscos da atividade, no sentido de reduzir ou eliminar os acidentes de trabalho e as doenças profissionais.

A administração é responsável por proceder à revisão periódica da política de segurança da empresa, com a finalidade de a manter atualizada e adaptada à organização.

A legislação aplicável, ao nível de segurança e saúde no trabalho é a DL. 441/91, DL. 26/94, DL. 109/00 e o Novo Código de Trabalho.

Para além da legislação referida anteriormente existem outras para riscos específicos, assim como regulamentos internos implementados pela empresa, nomeadamente: os Regulamentos Internos de estacionamento, de consumo de álcool, utilização de vestuário e calçado de proteção. Estão também implementados procedimentos de segurança, fichas de segurança de máquinas e produtos e instruções de segurança para a condução de veículos de transporte e elevação de cargas.

### 2.1.4 Descrição do processo produtivo

O fluxograma seguinte apresenta, de forma simplificada, o processo de produção da louça sanitária. O produto resultante deste processo poderá ser uma sanita, um bidé, um lavatório ou outros produtos de porcelana sanitária.

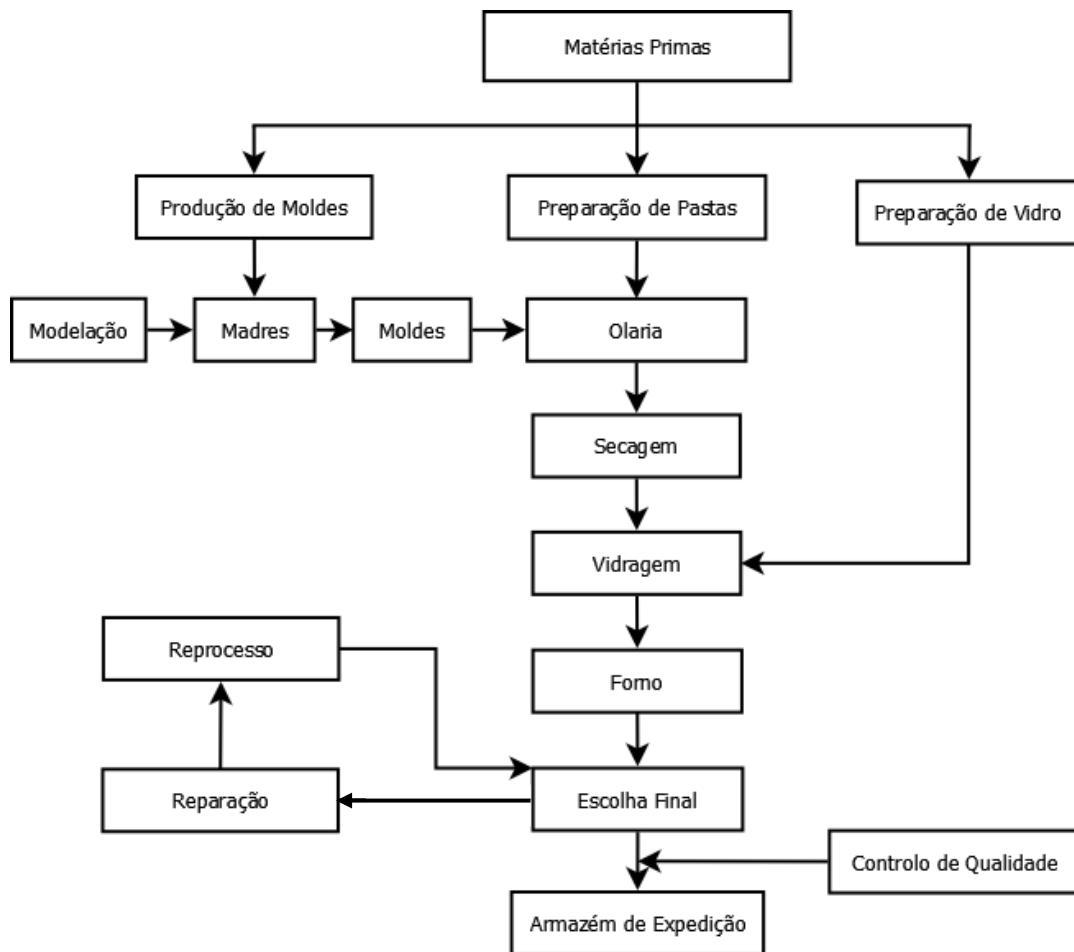


Figura 2 – Etapas do processo produtivo da Roca, S.A.

Este processo divide-se em etapas distintas, mas contínuas, tal como se pode observar na Figura 2. De um modo sucinto, estas etapas compreendem:

- A preparação de pastas em que são utilizadas determinadas matérias-primas, tais como a areia, a argila, o feldspato e a água. A pasta, após um processo de moagem e de diluição com caulino e argilas, é bombeada para peneiros, para serem retiradas as impurezas. Posteriormente, é colocada em tanques de distribuição, para ser enviada para os poços de enchimento da olaria.
- Numa etapa paralela, na preparação de vidro, as matérias-primas como a areia, o feldspato, o caulino, assim como a alumina, constituem um líquido viscoso de cor branca, cujo objetivo é o abastecimento dos carrosséis, onde as peças serão, posteriormente, vidradas. Antes de entrarem no processo produtivo, são analisadas, em laboratório, determinadas propriedades físico-químicas das matérias-primas, nomeadamente a densidade, a viscosidade, no sentido de assegurar que estas propriedades estão dentro dos parâmetros definidos pelas especificações técnicas.
- Na modelação são concebidas e elaboradas as “madres” (“mãe dos moldes”), para produzir moldes de gesso em série. Estes são os moldes que dão forma à peça de porcelana cerâmica.
- Na Olaria efetua-se o enchimento e escorrimento, o processo de desmoldar e procede-se ao primeiro acabamento em cru. No processo de enchimento, a pasta é colocada dentro do molde através de bombagem ou por gravidade. Quando o molde atinge a máxima capacidade de pasta, posteriormente a peça retém espessura, através da sucção de água, que é exercida pelo molde de gesso. Esta diminuição de humidade permite a obtenção de uma parede com uma determinada resistência, que assume as formas do molde. Após este tempo de espessura, ocorre o enchimento, com o objetivo de escoar a pasta líquida que não foi aproveitada. No processo de “desmoldar”, existe a separação da peça e do molde. Para o primeiro acabamento em cru, é necessário retirar as rebarbas e efetuar o acabamento de modo a que a peça fique totalmente lisa.
- No processo de secagem, ocorre a extração de humidade das peças, pela ação do calor. Esta etapa é efetuada em estufas.
- Numa etapa posterior, ocorre a escolha das peças em cru e a sua vidragem. Esta fase de acabamento é realizada no carrossel e integra um conjunto de tarefas de acabamento, polimento, inspeção e vidragem. Após inspeção, as peças são direcionadas para o carrossel ou para a recuperação em cru (Olaria), no caso de apresentarem defeitos.
- Após a aplicação da camada de vidro sobre as peças e a sua devida secagem, as peças são colocadas em vagonas, que através de um sistema automático, entram no forno. O forno é constituído por três zonas distintas: pré-aquecimento, cozedura e arrefecimento.
- Após a cozedura, numa etapa final do processo produtivo, as peças são inspecionadas na secção Escolha Final, cujo intuito é a verificação e deteção de eventuais defeitos nas peças em cozido, de acordo com as normas de qualidade da Roca.

### 2.1.5 A Secção Escolha Final

O estágio curricular foi realizado na Escolha Final. Esta secção visa a finalização do processo produtivo que se inicia na conceção do produto.

Nesta secção, é efetuada a inspeção visual das peças, bem como as inspeções de medidas, no sentido de detetar defeitos, pelo que na Escolha Final o rigor na execução das tarefas é um fator crucial para garantir a qualidade final do produto.

Durante o processo de escolha é importante o cumprimento de um conjunto de procedimentos, nomeadamente:

- O colaborador ou o *robot* efetua a colocação de peças de louça sanitária nas passadeiras, nas mesas de medição de lavatórios e nas mesas de escolha individual.



Figura 3 – Passadeira com lavatórios para posterior inspeção.



Figura 4 – Mesa de medição de lavatórios.

- A peça é carimbada, de forma a ser-lhe conferida rastreabilidade da Escolha Final.
- O registo da peça é efetuado em TPI (ecrã para visualização dos registos da produção) e em suporte de papel, pelo que é atribuído o resultado da inspeção. A inspeção aplicada à louça sanitária é de extrema

- importância, para a deteção de defeitos da porcelana sanitária. De acordo com esta inspeção, a peça de porcelana sanitária pode ser classificada como:
  - Peça boa – A peça está em conformidade, dentro dos parâmetros especificados pelas normas da Roca.
  - Peça para reprocesso – A peça apresenta um ou mais defeitos, que não são admissíveis segundo as normas da Roca, no entanto pode ser recuperada.
  - Peça inútil – A peça apresenta um ou mais defeitos que não são admissíveis segundo as normas da Roca, pelo que não poderá ser recuperada. A peça é enviada para um local específico de resíduos.
  - Peça para amola – A peça apresenta um ou mais defeitos, que não são admissíveis segundo as normas da Roca. No entanto pode ser amolada, isto é ser recuperada.

Durante a inspeção e classificação das peças, em relação aos defeitos que desencadeiem dúvidas em relação ao seu estado de admissibilidade, a peça deverá ser colocada no chão, e inspecionada a “olho nu”, a uma distância aproximada de 0,75 metros e na posição de instalação. A iluminação do local é crucial para observação, pelo que deverá ser de  $1000 \pm 100$  lux.

- Caso a peça seja classificada como boa, se se verificarem gretas superficiais, a sua reparação é realizada com um silicone apropriado.
- Caso a peça apresente pequena rebarbas, recorre-se ao punho pneumático para a sua reparação.
- Caso a peça escolhida seja uma bacia, procede-se ao ensaio de estanquicidade. O objetivo deste ensaio é a deteção e a verificação de fugas por injeção de ar. A máquina automática de estanquicidade necessita de ser calibrada periodicamente, com as peças padrões existentes (uma sanita “rota” e uma sanita “boa”).
- As peças são embaladas e paletizadas pelo *robot* ou pelo colaborador afeto a esta tarefa. As fitas e os batentes são fundamentais na paletização, uma vez que conferem proteção às peças.

Assim, de um modo geral, este departamento é constituído por 13 postos de trabalho, para a inspeção e a embalagem destas peças. O horário de funcionamento estipulado é das 8 horas da manhã até às 24 horas. Estão afetos 2 turnos, de 8 horas cada um. O primeiro turno inicia-se às 8 horas da manhã, tendo um intervalo entre as 9 horas e 20 minutos e as 9 horas e 45 minutos. O horário do almoço é entre as 13 horas e as 13 horas e 45 minutos. Este turno acaba às 16 horas e 11 minutos. Por outro lado, o segundo turno inicia às 16 horas e 11 minutos.

O elevador é um dos meios de transporte mais importantes para o Departamento Escolha Final. Através do elevador, cada palete de peças de porcelana sanitária é transportada da secção dos “Fornos” para sofrer inspeção e respetiva embalagem e paletização. Tal como se pode observar nas figuras 5 e 6, o elevador apresenta uma capacidade de 2 paletes por cada carga e descarga. Na figura 7, observa-se uma palete na zona de stock, com peças para serem verificadas e embaladas.



Figura 5 – Elevador contendo uma palete de peças de porcelana sanitária.



Figura 6 – Elevador observado da secção “Fornos”.



Figura 7 – Palete à espera para ser inspecionada e embalada.

Relativamente aos postos de trabalho e às passadeiras, estas obedecem a um conjunto de procedimentos, tais como, a rastreabilidade, a inspeção, o embalamento e a paletização. Cada posto de trabalho apresenta uma equipa de trabalho e cada colaborador apresenta uma tarefa específica.

A passadeira 1 é um posto de trabalho afeto à escolha de tanques de porcelana sanitária. Para a escolha de peças, nesta passadeira, a equipa é constituída por 4 colaboradores: um colaborador para efetuar a rastreabilidade, um colaborador que é responsável pela inspeção e avaliação de conformidade da peça, um colaborador que procede à colocação de etiquetas na peça e auxilia o quarto colaborador na colocação dos acessórios e mecanismos corretos, na embalagem e paletização das peças.

Na figura 8, observa-se a passadeira com tanques e com os respetivos acessórios, que posteriormente são embalados.



Figura 8 – Inspeção de tanques de porcelana sanitária na passadeira um.

Na figura 9, observa-se a peça de porcelana, antes da inspeção, com o material necessário para a sua rastreabilidade.



Figura 9 – Inspeção e rastreabilidade de tanques de porcelana sanitária na passadeira um.

A passadeira 2 tem o pressuposto de inspecionar e embalar sanitas de porcelana sanitária. Esta é constituída por duas cintas. Neste posto de trabalho, a equipa é constituída por 3 colaboradores: um colaborador para efetuar a rastreabilidade e verificar a estanquicidade das peças, um colaborador que é responsável pela inspeção e avaliação de conformidade da peça e um colaborador que procede à colocação de etiquetas na peça, à colocação dos acessórios e mecanismos corretos e à embalagem e paletização das peças. Nas figuras 10 e 11, observam-se as peças neste posto de trabalho.



Figura 10 – Bacia de porcelana sanitária na passadeira 2.



Figura 11 - Bacia de porcelana sanitária na passadeira dois a ser inspecionada.

Por outro lado, a passadeira 3 tem como pressuposto a inspeção e a verificação de conformidade ou não conformidade de lavatórios de porcelana sanitária.

Neste posto de trabalho, a equipa é constituída por 4 colaboradores: um colaborador responsável pela colocação das peças na passadeira e que efetua a rastreabilidade, um colaborador que é responsável pela inspeção e avaliação de conformidade da peça, um colaborador que procede à colocação de etiquetas na peça e colocação de sílica, caso esta seja necessária. Este colaborador auxilia o quarto colaborador na realização da embalagem e paletização das peças. Na figura 12, observa-se uma palete com lavatórios, numa plataforma, à espera de serem colocadas na passadeira, para posteriormente se obter o produto acabado.



Figura 12 – Palete com lavatórios de porcelana na plataforma.

Na figura 13, verificam-se as peças colocadas na passadeira, antes de serem avaliadas pelo colaborador responsável pela inspeção de qualidade das peças.



Figura 13 – Lavatórios de porcelana na passadeira três.

Relativamente ao posto de trabalho 5, este é um posto individual, pelo que a equipa é constituída apenas por um colaborador. Este colaborador é responsável pela rastreabilidade, inspeção, embalagem e paletização das peças. Os tipos de peças de porcelana sanitária a serem escolhidas, neste posto individual, são os lavatórios, de menor dimensão, os tanques e as colunas (peça integrante dos lavatórios).

No posto de escolha 6 são inspecionadas as colunas, sendo que a equipa que colabora na avaliação da sua conformidade e na paletização é constituída somente por um colaborador.

Assim como os postos de trabalho 5 e 6, o posto de trabalho 8 é constituído, geralmente, por um colaborador. No entanto, mediante o número de peças a serem produzidas, poderão estar afetos 2 colaboradores: um colaborador para a inspeção das peças e um segundo colaborador a auxiliar na paletização das mesmas. Neste posto são avaliados os lavatórios de menores dimensões e os urinóis.

No posto de trabalho 10 são inspecionadas as bacias, bem como os tanques. A equipa é constituída por 3 colaboradores: um colaborador responsável pela colocação de peças na passadeira e a sua respetiva rastreabilidade, um colaborador responsável pela avaliação de conformidade das peças e um terceiro colaborador responsável pelo embalamento e paletização das peças. Para a inspeção de sanitas, o colaborador responsável pela colocação das peças na passadeira, é responsável também pela verificação de estanquicidade destas.

Na figura 14, observa-se a plataforma com uma palete de tanques previamente inspecionados e embalados.



Figura 14 – Plataforma com tanques inspecionados.

O forno retrátil é uma plataforma onde é colocado plástico retrátil nas paletes com as peças previamente inspecionadas e embaladas, para posteriormente entrarem para o armazém de expedição.

Neste posto, a equipa é constituída por um colaborador responsável pelo funcionamento deste equipamento, que permite a embalagem das paletes com o plástico retrátil, assim como, a sua posterior identificação de produto acabado.

Na figura 15, observam-se as paletes na passadeira do forno retrátil, para *a posteriori* serem embaladas com o plástico retrátil. Nas figuras 16 e 17, observa-se a colocação do plástico retrátil na paleta, para ser enviada para o armazém de expedição.



Figura 15 – Paletes na passadeira do forno retrátil.



Figura 16- Colocação de lâmina retrátil na paleta.



Figura 17 - Colocação de plástico retrátil na paleta.



Figura 18 – Paletização referente aos tanques de porcelana sanitária.



Figura 19 - Paletização referente aos lavatórios de porcelana sanitária.

- A palete completa prossegue para o forno retrátil para o ajustamento do plástico retrátil à louça embalada previamente.



Figura 20 – Passadeira afeta ao forno retrátil.



Figura 21 – Forno retrátil.

Relativamente ao transporte de louça sanitária, na secção Escolha Final, este processo compreende as seguintes fases:

- Retirar a louça sanitária dos elevadores, provenientes da secção dos fornos. Os elevadores deverão permanecer ocupados com as peças, paletes ou cartão o tempo estritamente necessário.
- Deslocar as peças para a zona de *stock* de louça que está à espera de ser inspecionada e escolhida.
- Organizar as peças na zona de *stock*, tendo em consideração o modelo da peça, o forno em que a peça foi cozida, a cor, a olaria, e as experiências a serem estudadas.
- Abastecer e programar os *robots* de alimentação de peças nas passadeiras, assim como é importante para esta função abastecer a zona dos *robots* com paletes, cartão e deste modo, programá-los para estas situações.
- Abastecer a passadeira referente aos tanques com os mecanismos que pertencem ao modelo a escolher.
- Abastecer as passadeiras com paletes de *picking* de peças boas ou para reprocesso, de forma a serem finalizadas.
- As peças de porcelana sanitária que foram classificadas como peças para reprocesso e peças para resíduos, no fim da inspeção de cada modelo, devem ser colocadas nos elevadores. As peças para reprocesso prosseguem para a secção de Retoque, ao invés das peças classificadas como resíduos que são redirecionadas para um local específico, designado de tulha.

Por outro lado, para reduzir o esforço aplicado pelos colaboradores em determinadas tarefas, para aumentar a produtividade da própria secção e para reduzir o número de lesões provocadas no trabalho, existiu a necessidade de instalar *robots* nas secções da Escolha Final de Leiria 1, assim como de Leiria 2.

A instalação destas máquinas permitiu um aumento da produção, tendo em conta a complexidade de modelos das peças de porcelana, que é cada vez maior, e que para além disso, permitiu diminuir uma das maiores preocupações da administração da Roca: os acidentes de trabalho. É de referir que a manipulação dos *robots* é realizada por colaboradores com formação teórica e prática.



Figura 22 – Entrada de peças na passadeira através do *robot*.



Figura 23 – Saída de peça da passadeira através do *robot*.

## 2.2 O estado da arte

### 2.2.1 Definição de Simulação

Em latim, a palavra simulação é *Simulare* que significa fingir ou imitar. Este termo, tem tido um grande impacto no tecido empresarial.

No contexto atual e dada a crescente globalização da economia, a necessidade por parte das empresas do setor industrial, de alcançar as suas metas, de forma a criar valor e de aumentar a competitividade no mercado, tem aumentado significativamente. Como tal, a utilização da simulação computacional ganha um papel preponderante na análise e na previsão de resultados.

Segundo Williams e Narayanaswamy (Williams & Narayanaswamy, 1997), a “simulação é uma abordagem poderosa na modelação de sistemas de produção, que permitem a representação de sistemas complexos”. Segundo Anu Maria, a simulação é uma abordagem que permite avaliar o desempenho de um sistema, existente ou proposto, na ótica de diferentes configurações, e ao longo de determinados períodos de tempo (Anu Maria, 1997).

O processo de simulação é previamente modelado antes do sistema existente ou de um novo sistema, de forma a reduzir ou eliminar alternativas de insucesso e restrições imprevistas, ou até mesmo evitar a utilização em excesso de recursos, com incapacidade de atender às especificações requeridas. Com o objetivo de otimizar o desempenho do sistema, a simulação procura responder a perguntas como: “Qual o melhor projeto para uma nova rede de telecomunicações?” ou “Quais os recursos a utilizar e as suas especificações?” (Anu Maria, 1997). Para Banks (Banks, 1998), a simulação é uma técnica que permite a reprodução de uma operação de um determinado processo inerente a um sistema real ao longo do tempo. Esta é uma metodologia indispensável na resolução de situações problemáticas.

A simulação é uma tecnologia aplicada à análise e resolução de problemas, que requer uma definição bem clara do problema, as razões para a simulação e os resultados esperados. Ao utilizar-se a simulação, deve existir um equilíbrio entre a compreensão, o conhecimento do problema e o conhecimento subjacente à aplicação da simulação – o *software* a aplicar, as metodologias de análise e os conceitos teóricos da simulação (Beaverstock, Greenwood, Lavery, & Nordgren).

A sua abordagem permite o desenvolvimento de um modelo de um sistema real para um sistema computacional, de forma rápida e menos dispendiosa, e desta forma, analisar e estudar diferentes cenários e decisões. Consoante esta análise de sensibilidade, a simulação permite, na prática, a implementação do sistema (Seleima, Azaba, & Algeddwy, 2012). É fulcral na compreensão do comportamento do sistema, na identificação de oportunidades de melhoria, na avaliação de possíveis estratégias e alternativas a aplicar e na sua validação.

Desta forma, a simulação é uma ferramenta importante quando o risco envolvido para a sua manipulação e modelação é elevado, comparativamente com uma abordagem empírica, de tentativa e erro, aplicado a um sistema real (Reeb & Leavengood, 2003).

### **2.2.2 As características de um modelo de simulação**

Atualmente, existem diversos conceitos relacionados com a simulação (Banks, 1998).

Os modelos de simulação são caracterizados por serem estáticos ou dinâmicos, determinísticos ou estocásticos e contínuos ou discretos (Reeb & Leavengood, 2003).

O modelo estático representa um sistema num determinado horizonte temporal. A simulação de Monte Carlo é uma das ferramentas caracterizada por ser estática, uma vez que utiliza números aleatórios para a resolução de problemas estocásticos. Neste caso, o horizonte temporal é desprezável. Contudo, o modelo de simulação dinâmica representa um sistema, tendo em consideração a unidade de tempo.

Relativamente aos modelos de simulação determinísticos e estocásticos, nos modelos determinísticos, os parâmetros do modelo não assumem variabilidade, e assim, não contém variáveis aleatórias. No entanto, em ambientes fabris com processos produtivos, as variáveis são aleatórias ou imprevisíveis. Por isso, o modelo de simulação utilizado para estas situações é o modelo estocástico, que permite, deste modo, modelar sistemas com uma variabilidade elevada e com características aleatórias, tais como: os tempos de funcionamento, os ciclos de máquina, as taxas de chegada e os tempos de paragem das máquinas (Reeb & Leavengood, 2003).

Relativamente à análise de variáveis de estado, o modelo de simulação é classificado como simulação contínua ou simulação discreta.

Na simulação contínua, as variáveis alteram de forma contínua. Tem-se o exemplo do fluxo de água ao longo de um canal ou o voo de um avião. As variáveis de posição e de velocidade variam continuamente, como se verifica em simuladores de meteorologia (Reeb & Leavengood, 2003).

Um sistema é definido por um conjunto de entidades, geralmente de pessoas e máquinas, que atuam e interagem, com o pressuposto de realizar múltiplas operações. A simulação de eventos discretos é uma ferramenta que ajuda a compreender, a analisar, a gerir processos produtivos complexos e a analisar medidas de desempenho. É importante para ambientes de produção uma vez que é necessário proceder-se a estudos de aquisição e instalação de novos equipamentos, à atualização de máquinas existentes, bem como à alteração do *layout* da fábrica.

Os sistemas referentes a processos produtivos são, na maioria, modelados como simulação de eventos dinâmicos e discretos. Como são de natureza estocástica, as variáveis aleatórias possuem a capacidade de modelar tempos entre filas de espera e processos.

Estas características são notórias na indústria automóvel, militar e eletrónica (Reeb & Leavengood, 2003).

### **2.2.3 As vantagens da simulação**

A concorrência no setor da indústria de computadores provocou avanços tecnológicos que permitem às empresas uma melhor eficiência e eficácia na sua produção. Nos últimos anos tem-se verificado um aumento exponencial na utilização da simulação por parte das empresas (Banks, 1998).

A simulação permite testar todos os aspetos de uma proposta, sem adquirir recursos, que posteriormente poderão não ser utilizados. Para além disso, a simulação possibilita comprimir ou expandir o

tempo, isto é, permite a aceleração dos eventos, ou o contrário, de forma a que sejam investigados e estudados (Banks, 1998).

Um dos aspetos mais vantajosos na utilização de um *software* de simulação é a possibilidade de explorar alternativas. Após ter-se desenvolvido um modelo de simulação válido, é exequível a exploração de novas políticas, de procedimentos operacionais, ou de métodos, sem custos e sem tempos de paragem, que com um sistema real não seria praticável (Banks, 1998).

Segundo Banks (Banks, 1998), o diagnóstico de problemas é também um benefício proveniente da utilização de um *software* de simulação. O diagnóstico de problemas, dificuldades e a realização de uma análise sobre a importância dessas variáveis aumenta a compreensão dos seus efeitos sobre o desempenho do sistema global. A simulação permite uma compreensão clara em relação às interações entre as variáveis constituintes do sistema.

Numa unidade fabril, os “pontos de gargalo”, isto é, os pontos de restrição são problemáticos para o processo produtivo. É importante a sua identificação, sabendo que estes são um efeito e não uma causa. Ao utilizar-se a simulação, no sentido de proceder a uma análise de restrições inerentes no processo, é possível reconhecer a causa de determinados atrasos no processo, de informações ou de utilização específica de materiais (Banks, 1998).

A simulação auxilia e desenvolve a compreensão sobre o sistema e de que modo ele funciona. No entanto, não indica previsões de como o sistema irá operar (Banks, 1998).

Por outro lado, a simulação permite a visualização do plano, ou seja, das instalações, com o recurso de desenhos *CAD* e através de “*add-ins*” adicionais aos pacotes do *software* de simulação. De acordo com o *software*, permite visualizar, de forma bidimensional e tridimensional, as operações intrínsecas ao sistema, a partir de diversos ângulos e de níveis de ampliação (Banks, 1998).

Ao proceder-se à simulação, constrói-se uma opinião objetiva, uma vez que os resultados deste processo foram previamente modelados, testados, validados e representados visualmente (Banks, 1998).

No contexto da simulação é necessário existir uma predisposição para a mudança e para a conceção de novos sistemas e reformulação de sistemas já existentes. É importante a interação com todos os envolvidos no projeto de simulação, durante a fase da formulação do problema, para uma melhor análise dos cenários em causa e, assim, construir um modelo que responda às necessidades relativas desses cenários (Banks, 1998).

Em termos de custos, um estudo de simulação é substancialmente menos dispendioso em comparação com os custos inerentes à instalação de um sistema ou à sua modificação, sem um estudo prévio (Banks, 1998).

Um estudo de simulação permite, também, o desenvolvimento de competências ao nível de trabalho de equipa. À medida que o modelo de simulação progride, a equipa fornece variáveis de entrada para este. Com isto, a equipa de trabalho aprende com os erros e as falhas e, deste modo, aprende a trabalhar com maior eficácia e eficiência. Para a melhoria da eficácia e eficiência, é necessário estabelecer especificações. A simulação pode ser utilizada para especificar requisitos para uma determinada máquina. Estes requisitos podem ser estabelecidos, ao simular-se diferentes capacidades para a máquina em questão (Banks, 1998).

Chaturvedi (Chaturvedi, 2010) escreveu que a simulação é um processo de apoio à tomada de decisão, dada as seguintes razões:

- Melhoria na capacidade de compreensão das consequências das decisões e os seus efeitos nas organizações;
- Identificação dos pontos fortes e fracos;
- Identificação de estratégias políticas que vão de encontro aos objetivos da organização;
- Desenvolvimento e comunicação de diferentes cenários e perspetivas;
- Melhoria na velocidade e eficiência de compreensão e resolução de estratégias. A simulação permite analisar de que forma é que os recursos disponíveis estão a ser aplicados. Assim, auxilia na identificação de problemas como a excessiva capacidade e as restrições dentro de um sistema;
- Construção de ambientes virtuais, tratando-se de um processo que é viável em termos financeiros;
- Permite a elaboração de protótipos relativos a uma ideia, estratégia e decisão num horizonte temporal reduzido.

#### **2.2.4 As desvantagens da simulação**

São algumas as desvantagens da utilização da simulação, nomeadamente:

1. A construção do modelo de simulação requer um estudo exigente. A simulação requer uma aprendizagem contínua, ao longo do tempo, através da experiência.
2. Os resultados de um estudo de simulação são, por vezes, difíceis de interpretar. A maior parte das simulações são variáveis aleatórias, baseadas em dados aleatórios. Com este facto, é difícil determinar se uma observação é um resultado de uma relação do sistema ou de aleatoriedade.
3. A modelação da simulação é, em certas circunstâncias, um processo caro e demorado.
4. A simulação pode ser interpretada, analisada e utilizada de forma inadequada.

Em defesa do processo de simulação, as desvantagens acima referidas são compensadas do seguinte modo:

1. Simuladores. Os fornecedores de *software* de simulação têm vindo a desenvolver ativamente os modelos dos quais só necessitam de *inputs* para o seu funcionamento.

2. Análise de *outputs*. A maioria dos fornecedores de *software* de simulação têm desenvolvido determinadas capacidades de análise: adicionar recursos para a realização de uma análise mais rigorosa e muito mais extensa. Isto permite reduzir os requisitos computacionais por parte do usuário. No entanto, o usuário deve compreender o processo que está a analisar.
3. A velocidade de simulação. A simulação pode ser realizada de forma mais rápida hoje do que ontem, e ainda mais rápido amanhã. Isto deve-se aos avanços em *hardware* que permitem a rápida execução de cenários.

### 2.2.5 O desenvolvimento de um modelo de simulação

Os modelos de simulação são constituídos por determinados componentes, tais como: os elementos do sistema, as medidas de desempenho, as variáveis de entrada e as relações funcionais.

De acordo com Anu Maria (Anu Maria, 1997), no desenvolvimento de um modelo de simulação, as etapas que permitem a sua projeção e análise são:

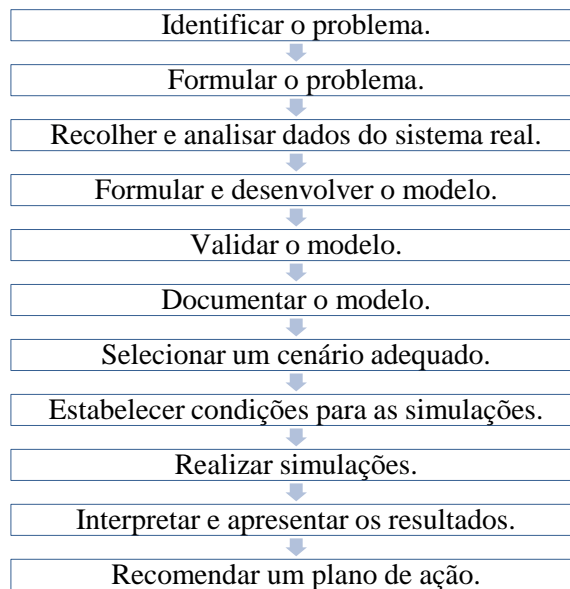


Figura 24 – Etapas de um modelo de simulação, segundo (Anu Maria, 1997).

Segundo Banks (Banks, 1998), um estudo de simulação compreende 12 etapas, tal como se observa no seguinte fluxograma.

1. Formulação do problema.

Cada estudo de simulação necessita de um conjunto de pressupostos e de uma transparente formulação do problema. Como tal, é imprescindível que durante esta análise, o problema seja claramente compreendido. À medida que o estudo de simulação progride, eventualmente, o problema terá de ser reformulado.

2. Definição de objetivos.

Esta etapa consiste na realização de uma proposta. O objetivo é, essencialmente, traçar questões capazes de responder ao estudo de simulação a proceder. O plano do projeto deve incluir uma visão sobre os diferentes cenários a estudar.

3. Concetualização do modelo.

O sistema real que é analisado e estudado é reportado para um modelo, que traduz relações matemáticas e lógicas sobre os componentes e a estrutura do sistema. No início, a modelação é realizada de forma simples, até que, à medida do estudo e da análise de cenários, desenvolve-se um modelo de complexidade apropriada.

4. Recolha de dados.

Para a recolha de dados, é crucial um cronograma de requisitos para estes. Durante a construção do modelo, é normal que ocorra uma progressão e renovação de dados recolhidos.

5. Tradução do modelo.

Durante esta etapa, o modelo construído é codificado em suporte informático.

6. O modelo é verificado?

Na verificação do modelo pretende-se saber se o modelo está a funcionar adequadamente. A verificação do modelo é um processo contínuo. Desse modo, é imprescindível a verificação, mesmo que o modelo de simulação não esteja completo.

7. O modelo é válido?

Esta etapa traduz se o modelo conceptual é uma representação precisa do sistema real em questão. Neste sentido, procede-se a uma análise de sensibilidade.

8. Plano de experiências.

É importante proceder a um plano de experiências no sentido de, para cada cenário que se está a simular, se tome uma decisão, nomeadamente no número de simulações a realizar, na duração de cada simulação e nos “*inputs*” a introduzir, de acordo com o que é realmente necessário.

#### 9. Execução do modelo de simulação.

A execução do modelo, bem como a sua análise, é importante no sentido de estimar índices de desempenho para os diferentes cenários que estão a ser simulados.

De acordo com os resultados e análise das simulações executadas, é fundamental determinar se será necessário proceder a um maior número de simulações, e se será necessário adicionar e enquadrar outros cenários, tendo em consideração aos índices de desempenho obtidos previamente (Banks, 1998).

#### 10. Documentação dos resultados de simulação.

A documentação é necessária na medida em que caso o modelo de simulação seja utilizado para replicação, é importante compreender como é que este funciona. O resultado da análise de um modelo de simulação deve ser documentado em relatório, de forma objetiva, clara e consistente. A sua documentação permite fazer uma revisão à formulação do problema e às alternativas que estão anexadas. Permite também uma revisão aos critérios utilizados para comparação com sistemas alternativos e aos resultados das experiências e recomendações (Banks, 1998).

#### 11. Implementação do modelo.

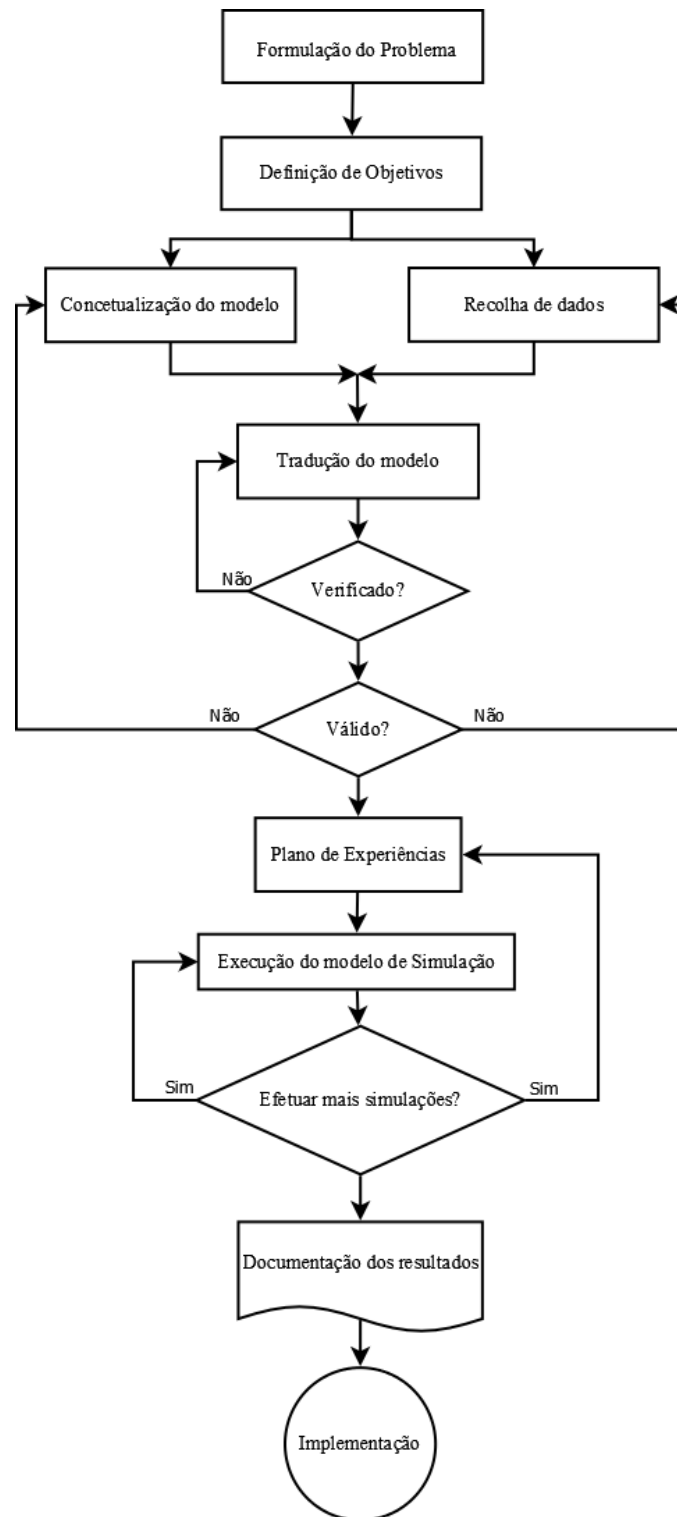


Figura 25 – Fluxograma do processo que envolve a Simulação, segundo Banks (Banks, 1998).

### 2.2.6 As áreas de aplicação da simulação

Nos dias de hoje existe uma panóplia de aplicações no campo da Simulação, verificando-se ao nível da indústria (essencialmente na produção), dos sistemas públicos e na prestação de serviços.

A *Winter Simulation Conference (WSC)* é uma forma de apreender os avanços tecnológicos na simulação, a sua progressão e as mais recentes aplicações desta (Banks, 1998).

No que respeita à produção e utilização de materiais, as aplicações da simulação são:

- Minimização de tempos de atrasos de peças pré-fabricadas, antes da montagem;
- Modelação flexível e análise de sistemas;
- Análise do fluxo de material relativo à montagem de automóveis;
- Análise e avaliação do custo da qualidade do produto.

Ao nível de sistemas públicos, as aplicações da simulação centram-se nomeadamente em:

- Sistemas de saúde, verificando-se uma elevada afluência da simulação na avaliação de centros cirúrgicos e das áreas inerentes aos cuidados intensivos, bem como na avaliação de equipamentos automatizados, que estão afetos a um laboratório clínico, ou também, na avaliação de uma equipa de enfermeiros e na análise de diferentes cenários relativamente à população de pacientes (Banks, 1998);
- Sistemas militares, com uma elevada importância para o desenvolvimento e implementação de medidas de eficiência e de eficácia;
- Recursos naturais, nomeadamente na análise de pontos de poluição, assim como na recolha de dados e tomada de decisão no controlo de plantas daninhas e na avaliação de índices de qualidade de água;
- Prestação de serviços públicos, analisando-se o sistema de emergência.

Por outro lado, segundo Banks (Banks, 1998) nos sistemas e prestação de serviços, existem aplicações ao nível de:

- Transporte, mais concretamente na avaliação de procedimentos de controlo de tráfego nas estradas, em situações de obras, na análise de sistemas para veículos inteligentes, na modelação e avaliação de trânsito rápido com controlos automáticos e manuais;
- Desempenho de sistemas informáticos, nomeadamente avaliação de protocolos de gestão de base de dados e avaliação de modelos de filas de espera;
- Transporte aéreo, tendo em consideração as operações do aeroporto e do avião;
  - Sistemas de comunicação, tal como a avaliação do processo de serviços de telefone.

Para Chaturvedi (Chaturvedi, 2010), a simulação tem aplicações no campo da:

- Projeção e análise de sistemas;

- Avaliação dos requisitos de *hardware* e *software* para um sistema informático;
- Determinação de políticas de encomenda para um sistema de inventário;
- Elaboração de sistemas de comunicação;
- Elaboração de canais de transportes;
- Avaliação de projetos ao nível da prestação de serviços, tais como em hospitais, ou restaurantes;
- Análise de sistemas financeiros e económicos.

### 2.2.7 A simulação na indústria

Segundo Ingalls e Preston White (Preston White & Ingalls, 2015), o campo de aplicação da simulação é amplo e pode ser dividido em duas categorias: a simulação “*man-in-loop*” e a simulação para o design de processos.

A simulação “*man-in-loop*” é direcionada para a formação, nomeadamente para profissionais que pretendam melhorar os seus conhecimentos, como exemplo, aprender determinados procedimentos de emergência. Na área da aviação, os pilotos treinam em simuladores de voo, com o objetivo de experimentar o “*cockpit*” do avião. Até na área dos jogos de computador, a simulação encontra-se presente.

Relativamente à simulação para o *design* de processos, considera-se, por exemplo a simulação de fenómenos aerodinâmicos ou de operações de um processo de uma unidade fabril (Preston White & Ingalls, 2015).

A versatilidade, a flexibilidade e a dinâmica dos métodos de simulação caracterizam a sua aplicabilidade em processos de tomada de decisão, tendo ênfase no setor industrial.

No setor industrial é crucial que os gestores e os engenheiros possuam uma visão ampla do efeito que determinadas alterações desencadeiam no sistema. Se é realizada uma alteração, numa dada localização, o seu impacto poderá ser previsível, no entanto, sem uma ferramenta de simulação, poderá ser impossível determinar o efeito desta alteração no desempenho global do sistema (Law & Kelton, 2000).

A simulação tem sido aplicada a diferentes níveis no setor industrial, como em atividades de planeamento, na logística e gestão da cadeia de abastecimento (Seleima, Azaba, & Algeddwy, 2012). A sua aplicação é extrapolada ao nível de otimização de *layouts* de unidades fabris, no estudo de linhas de montagem, e nos recursos de produção, incluindo a aquisição de novos equipamentos.

O método com maior incidência e com maior utilização na compreensão e gestão de processos é a simulação de eventos discretos, em sistemas de produção, sendo que os *softwares ProModel, ARENA e Flexsim* são os mais utilizados. O *software ProModel* é um dos métodos mais adequados em termos de fluxos de

trabalho e na engenharia de processos. A funcionalidade deste *software* é a modelação de sistemas de produção, desde sistemas flexíveis de fabrico até sistemas de cadeia de abastecimento.

O *ProModel* possibilita a oportunidade de testar novas ideias ou por outro lado, a melhoria contínua do processo. Foca-se na utilização de recursos, na capacidade do sistema, na melhoria dos processos, no rendimento, na análise de restrições e na logística e cadeia de abastecimento (Promodel, 2016).

Em alternativa, o *ARENA* é outro *software* de simulação que possibilita às empresas aumentar a sua produção, ao analisar o fluxo do processo produtivo (os sistemas de embalagem, o controlo do *stock*) e ao avaliar as potenciais mudanças neste processo (Arena Simulation, 2016).

O *software Flexsim* é uma ferramenta de análise, desenvolvimento e suporte à simulação, orientada para o desenvolvimento, a modelação, a simulação, a visualização e a monitorização das atividades do processo. Este *software* permite a construção de um modelo tridimensional e computacional, que traduz um sistema real. Através de animações, gráficos e relatórios de desempenho, possibilita a identificação de problemas e a avaliação de soluções alternativas, num reduzido horizonte temporal.

Paralelamente, o *software* Matlab também permite o estudo de um sistema real, dado ser possível construir aplicações para a indústria, no entanto, apresenta um reduzido impacto no tocante à sua utilização no setor industrial cerâmico.

A construção de uma aplicação com uma interface gráfica no Matlab permite automatizar uma determinada função, que se utiliza diversas vezes e, para além disso, permite partilhar a terceiros uma demonstração interativa. De um modo geral, a interface gráfica em Matlab é dotada de componentes estruturados de eventos, como os controlos de interface (*uicontrols*) e os menus (*uimenu*), que facilmente possibilitam o desenvolvimento da interface gráfica. Uma das vantagens da sua utilização é o facto de conferir comunicação entre os diferentes componentes, sem ter que utilizar comandos de programação. Estes componentes têm sido desenvolvidos e conferem ao programa, um conjunto de ferramentas intuitivo. Estas ferramentas, por sua vez, conferem um aumento da produtividade por parte do utilizador. Por outro lado, o conjunto de componentes de interface fornecidos pelo Matlab permitem a formação e o desenvolvimento de interfaces gráficas (GUIs) correspondentes e equivalentes às que são utilizadas em pacotes de *software* mais sofisticados (Marchand & Holland, 2003).

No desenvolvimento deste relatório recorreu-se ao *Matlab-Guide*, para a elaboração e a construção de uma aplicação, com capacidade de simular computacionalmente o sistema em causa.



## 3. Desenvolvimento

### 3.1 A Teoria de Filas de Espera

A Teoria de filas de Espera é o resultado do trabalho de A.K. Erlang, cientista dinamarquês que, no início do século XX, dedicou-se ao estudo de uma tecnologia recente na época - o telefone, onde se verifica uma grande relevância em relação às filas de espera (Tavares, Oliveira, Themido, & Correia, 1996).

A Teoria de Filas de Espera compreende o estudo matemático de filas de espera. Verifica-se a existência destas sempre que a procura no momento presente é superior à capacidade atual, para fornecer o serviço em questão. Contudo, em determinadas situações é difícil efetuar uma previsão precisa, em termos de cadência de chegada, bem como em termos do tempo necessário para prestar o serviço (Hillier & Lieberman, 1990).

A Teoria de Filas de Espera apresenta um grande número de modelos matemáticos alternativos para descrever um caso específico de fila de espera, sendo que um dos objetivos é alcançar um equilíbrio económico entre o custo do atendimento e os custos intrínsecos no tempo de espera para esse atendimento (Hillier & Lieberman, 1990).

Um problema de Teoria de Filas de Espera é caracterizado por existir uma relação de necessidade de um serviço por parte de clientes – pessoas, veículos ou entidades físicas – que compreende o tempo de espera numa fila (Tavares, Oliveira, Themido, & Correia, 1996).

O modelo de fila de espera é utilizado na tomada de decisão para o dimensionamento do serviço ou da configuração do sistema. É importante o seu estudo para otimizar o funcionamento do sistema e encontrar soluções equilibradas e viáveis, de forma a colmatar situações de congestionamento e de rarefação.

Em situações de congestionamento, para serem atendidos, os clientes esperam na fila períodos de tempo elevados. Deste modo, como a taxa de ocupação dos servidores é elevada, pressupõe uma valorização elevada do custo dos servidores, comparativamente ao custo que é atribuído ao tempo de espera dos clientes. Relativamente ao tempo de serviço, se este for elevado incorre a custos elevados, uma vez que a capacidade de serviço não é suficiente e como consequência, o comprimento da fila de espera é elevado (Hillier & Lieberman, 1990).

Contudo, em situações de rarefação, o mesmo não acontece. Os servidores permanecem inativos durante uma elevada percentagem de tempo, verificando-se uma elevada taxa de inatividade (Tavares, Oliveira, Themido, & Correia, 1996).

Na otimização de sistemas de filas de espera, persistem dificuldades na valorização do tempo de espera dos clientes e o estabelecimento de *trade-offs* entre o custo do serviço e o custo do tempo perdido pelos clientes. A informação importante para esta otimização é analisada quando se dispõe de dados sobre os custos de espera dos clientes, para além de indicadores de qualidade do serviço prestado aos utentes e de medidas de desempenho (Tavares, Oliveira, Themido, & Correia, 1996).

### 3.1.1 A estrutura de um modelo de filas de espera

O modelo de fila de espera é um processo básico que consiste numa fonte de entrada, da qual comparecem, ao longo do tempo, clientes que necessitam de atendimento. Estes clientes entram no sistema e aderem à fila de espera. Consoante o critério previamente estabelecido para a disciplina da fila, um membro da fila é selecionado para ser atendido. Após o atendimento solicitado pelo cliente estar terminado, este sai do sistema (Hillier & Lieberman, 1990).

O sistema de fila de espera é constituído por três elementos fundamentais – a fonte (população), a fila e o serviço. Na figura seguinte observa-se a estrutura básica de um modelo de filas de espera.

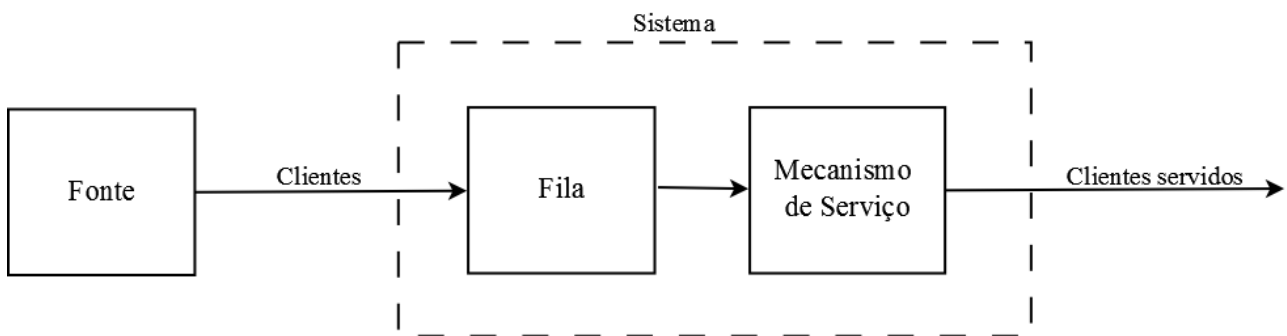


Figura 26 – Estrutura básica de um sistema de Fila de espera (fonte: (Hillier & Lieberman, 1990).

A fonte, ou população, origina os clientes do sistema. Com isto, o processo de chegada dos clientes é caracterizado, de acordo com a:

- Dimensão da população:

Em termos de dimensão da população, esta caracteriza-se por ser infinita ou, por outro lado, finita. Quando o número de elementos no sistema, num horizonte temporal, é uma percentagem elevada, trata-se de uma população infinita. Ou seja, é a probabilidade de ocorrer uma nova chegada, sabendo que não é

influenciada pelo número de clientes que já se encontravam no sistema. Ao invés, uma população finita é constituída por um número de clientes no sistema caracterizado por ser uma parte significativa da população.

- Dimensão da chegada:

Esta pode ser classificada como simples – os clientes chegam ao sistema, um de cada vez – ou em grupo.

- Controlo das chegadas:

A classificação do controlo das chegadas é feita consoante dois grupos: as chegadas controláveis e as chegadas incontroláveis.

- Distribuição das chegadas:

Descreve a distribuição do intervalo de tempo entre duas chegadas sucessivas. Por outras palavras, é a distribuição do número de chegadas por unidade de tempo. O padrão da chegada pode ser aleatório ou constante. A aleatoriedade nas chegadas é traduzida pelos intervalos de tempo entre as chegadas sucessivas, que não podem ser previstos com certeza.

Por sua vez, o padrão de chegadas é constante uma vez que os intervalos de tempo entre as chegadas sucessivas são valores fixos.

- Taxa de chegada:

Esta medida de desempenho traduz o número médio de clientes que pretendem o serviço por unidade de tempo. Quando a taxa de chegada não varia, considera-se independente do estado do sistema. Numa situação em que se verifique a existência de muitos elementos no sistema, a taxa de chegada diminui.

- Atitude dos clientes:

A atitude dos clientes é classificada em paciente e impaciente. Quando os clientes permanecem na fila até serem atendidos, seja qual for o comprimento da fila, estes designam-se de clientes pacientes. Pelo contrário, são classificados de clientes impacientes, em situações de desistência, em que o tempo de espera vai ser superior ao que consideram razoável (Tavares, Oliveira, Themido, & Correia, 1996).

A fila, um dos elementos deste sistema, tem como características:

- Número de filas de espera:

O número de filas de espera é classificado em fila simples – uma única fila para todos os servidores – ou em fila múltipla – uma fila para cada servidor, quando se verifica mais do que um servidor no sistema.

- Comprimento de filas de espera:

Este atributo pode ser considerado infinito, quando a sua capacidade máxima é muito superior, relativamente ao número de elementos que a podem constituir. Em contrapartida, é denominada de fila finita, quando a fila só pode suportar, no máximo, um número reduzido de clientes.

- **Disciplina da fila:**

No que respeita à disciplina da fila, esta obedece a um dos três critérios apresentados: ou segue a disciplina FIFO (*First In First Out*), em que os clientes são atendidos pela ordem de chegada; ou por outro lado segue a disciplina LIFO (*Last In First Out*) - a disciplina rege-se por um conjunto de prioridades. O terceiro critério a adotar pode ser aleatório.

Por último, o Serviço possui aspetos fundamentais, nomeadamente a:

- **Configuração do Serviço:**

É caracterizada em função do número de servidores em paralelo e do número de fases do atendimento e exprime o modo como o atendimento está estruturado.

- **Dimensão do Serviço:**

A dimensão do atendimento pode ser simples – os clientes são atendidos um de cada vez - ou múltiplo – quando os clientes são atendidos em grupo.

- **Distribuição do tempo de serviço:**

O serviço pode ser caracterizado em função da distribuição do tempo de serviço ou da distribuição do número de clientes potencialmente atendidos por unidade de tempo. O tempo de serviço é classificado como constante ou aleatório.

- **Taxa de serviço:**

A taxa de atendimento indica o número médio de clientes atendidos por cada servidor e por unidade de tempo. Dito por outras palavras, a taxa de serviço representa a capacidade média de atendimento de cada servidor. Esta medida pode ser classificada como independente e dependente do estado do sistema. Uma taxa de serviço dependente do estado do sistema varia em função do número de elementos no sistema (Tavares, Oliveira, Themido, & Correia, 1996).

### **3.1.2 A terminologia e a notação utilizada**

Segundo Hillier e Lieberman, a terminologia utilizada para o desenvolvimento e resolução de um problema de filas de espera encontra-se explícita no seguinte quadro (Hillier & Lieberman, 1990).

Quadro 2 – Simbologia e respetiva descrição da teoria de filas de espera (Hillier &amp; Lieberman, 1990).

Símbolo	Descrição
$\lambda_n$	Ritmo médio de chegadas de clientes por unidade de tempo.
$1/\lambda$	Tempo entre chegadas de clientes.
$\mu_n$	Ritmo médio de serviço do sistema.
$1/\mu$	Tempo esperado de serviço.
$S$	Número de servidores.
$\rho$	Taxa de ocupação (fator de utilização da infraestrutura de serviço).
$P_n$	Probabilidade de estarem $n$ clientes no sistema.
$L$	Número de clientes no sistema.
$L_q$	Comprimento da fila de espera (exclui os clientes que estão a ser servidos).
$W$	Tempo de espera no sistema (inclui tempo de serviço) para cada cliente.
$W_q$	Tempo de espera no sistema (exclui tempo de serviço) de cada cliente.

### 3.1.3 As relações fundamentais entre $L$ , $W$ , $L_q$ e $W_q$

Assume-se que  $\lambda_n$  é a constante  $\lambda$  para qualquer  $n$ . Este facto foi comprovado num sistema de fila de espera, em estado estacionário. Deste facto resultam a equação (1) mencionada como a Fórmula de *Little* e a equação (2) (Hillier & Lieberman, 1990).

$$L = \lambda W \quad (1)$$

$$L_q = \lambda W_q \quad (2)$$

Se  $\lambda_n$  não é constante, então  $\lambda$  é substituído pela taxa média de chegada,  $\bar{\lambda}$ . Assume-se que a taxa média de serviço é constante,  $\frac{1}{\mu}$ , para qualquer  $n \geq 1$ . Segue-se, deste modo a equação (3). Esta traduz o tempo de espera no sistema e exclui o tempo de serviço.

$$W = W_q + \frac{1}{\mu} \quad (3)$$

Estas relações de *Little* são de extrema importância, dado que permitem, de forma direta, a determinação das medidas de desempenho relativamente a  $L$ ,  $W$ ,  $L_q$  e  $W_q$  (Hillier & Lieberman, 1990).

### 3.1.4 A distribuição das chegadas e do atendimento

As características operacionais de sistemas de filas de espera são determinadas por duas propriedades estatísticas: a distribuição dos tempos entre chegadas e a distribuição dos tempos de serviço (Hillier & Lieberman, 1990).

Em sistemas reais, estas distribuições assumem qualquer forma. A única restrição é a não ocorrência de valores negativos. Contudo, para formular um problema de Teoria de Filas de Espera de um sistema real e representá-lo, é necessário especificar a forma de cada uma destas distribuições. A forma assumida deve ser realista, de modo a que o modelo forneça previsões verossímeis e, ao mesmo tempo, simples, no sentido de o modelo ser matematicamente tratável (Hillier & Lieberman, 1990).

Posto isto, é importante a caracterização das distribuições das chegadas e do atendimento, no estudo de modelos de fila de espera. E como tal, procede-se à análise de informação concreta e real de chegadas e tempos de atendimento, através de técnicas estatísticas elementares – estatística descritiva e inferência. Esta metodologia consiste em descrever a informação recolhida e os parâmetros das amostras; inferir através dos parâmetros das amostras, os parâmetros da população e ajustar uma distribuição teórica ao histograma experimental (Tavares, Oliveira, Themido, & Correia, 1996).

Os modelos analíticos pressupõem que a distribuição de probabilidades dos tempos entre chegadas siga uma distribuição de Poisson. O tempo de atendimento segue uma distribuição exponencial negativa.

Com base nas constatações anteriores, a distribuição de probabilidade mais importante na teoria das filas de espera é a distribuição exponencial.

### 3.1.5 A caracterização da distribuição exponencial

A distribuição exponencial negativa tem como função densidade de probabilidade da variável aleatória  $T$  (Hillier & Lieberman, 1990):

$$f(t) = \begin{cases} \alpha e^{-\alpha t} & \text{se } t \geq 0 \\ 0 & \text{se } t < 0 \end{cases} \quad (4)$$

A variável  $\alpha$  é o parâmetro de distribuição e a variável  $t$  representa o tempo de serviço ou tempo de chegada (Tavares, Oliveira, Themido, & Correia, 1996).

Para esta distribuição, as probabilidades cumulativas são (Hillier & Lieberman, 1990):

$$P\{T \leq t\} = 1 - e^{-\alpha t}, t \geq 0 \quad (5)$$

$$P\{T > t\} = e^{-\alpha t}, t \geq 0 \quad (6)$$

As equações que traduzem o valor esperado e a variância de  $T$  são (Hillier & Lieberman, 1990):

$$E(T) = \frac{1}{\alpha} \quad (7)$$

$$\text{var}(T) = \frac{1}{\alpha^2} \quad (8)$$

Para um modelo de filas de espera, a variável  $T$  segue uma distribuição exponencial, de acordo com seis propriedades específicas. Estas propriedades são:

1. “A função  $f(t)$  é estritamente decrescente, quando  $t \geq 0$ ”, tal como se observa na figura 27 (Hillier & Lieberman, 1990).

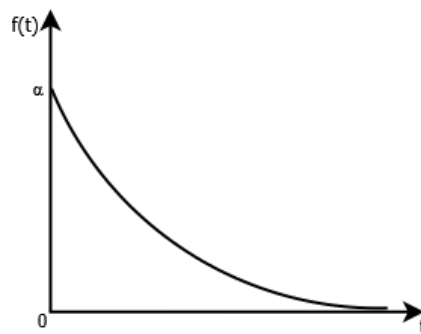


Figura 27 – Gráfico da função densidade para a distribuição exponencial negativa (Fonte: (Hillier & Lieberman, 1990))

Desta propriedade, obtém-se para valores estritamente positivos, as seguintes probabilidades:

$$P\{0 \leq T \leq \Delta t\} > P\{t \leq T \leq t + \Delta t\} \quad (9)$$

As probabilidades anteriormente enunciadas correspondem à área relativa à curva da função  $f(t)$ , dentro do intervalo de tempo,  $\Delta t$ . A altura média da curva é menor para a segunda probabilidade do que para a primeira probabilidade (Hillier & Lieberman, 1990).

Esta propriedade exclui situações de potenciais clientes que se aproximam do sistema de filas, e que pretendam adiar a sua entrada, ao constatarem que existe outro cliente à sua frente (Hillier & Lieberman, 1990).

2. “A distribuição Exponencial Negativa não tem memória” (Hillier & Lieberman, 1990).

Esta propriedade é suportada pelo facto de a probabilidade de um acontecimento ser independente do instante em que ocorreu o acontecimento anterior (Tavares, Oliveira, Themido, & Correia, 1996). Por outras palavras, significa que a distribuição de probabilidades do intervalo de tempo até ao próximo evento – chegada de um novo cliente – é constante independentemente de há quanto tempo é que ocorreu o último evento – chegada do último cliente. Esta propriedade é aplicada através da seguinte equação (Tavares, Oliveira, Themido, & Correia, 1996).

$$\begin{aligned}
 P[T > t + \Delta t \mid T > \Delta t] &= \frac{P[T > \Delta t, T > t + \Delta t]}{P[T > \Delta t]} \\
 \Leftrightarrow P[T > t + \Delta t \mid T > \Delta t] &= \frac{P[T > t + \Delta t]}{P[T > \Delta t]} \\
 \Leftrightarrow P[T > t + \Delta t \mid T > \Delta t] &= \frac{e^{-\alpha(t+\Delta t)}}{e^{-\alpha\Delta t}} \\
 \Leftrightarrow P[T > t + \Delta t \mid T > \Delta t] &= e^{-\alpha t} \\
 \Leftrightarrow P[T > t + \Delta t \mid T > \Delta t] &= P[T > t] \tag{10}
 \end{aligned}$$

Contudo, para o tempo de serviço e de atendimento, esta propriedade não se aplica em situações que se verifiquem diferentes tipos de serviço. O tempo de serviço segue uma distribuição exponencial negativa quando o serviço é distinto de cliente para cliente (Tavares, Oliveira, Themido, & Correia, 1996).

3. “O mínimo de variáveis aleatórias exponenciais independentes segue uma distribuição exponencial” (Hillier & Lieberman, 1990).

Com esta propriedade, matematicamente supõe-se que  $T_1, T_2, \dots, T_n$  são variáveis exponenciais negativas com parâmetros  $\alpha_1, \alpha_2, \dots, \alpha_n$ , respectivamente. A variável aleatória que assume o mínimo dos valores de  $T_1, T_2, \dots, T_n$  é:

$$U = \text{mínimo}\{T_1, T_2, \dots, T_n\} \tag{11}$$

Esta variável representa o instante em que o primeiro dos  $n$  eventos ocorre, sabendo que  $T_i$  representa o instante em que ocorre cada evento (Tavares, Oliveira, Themido, & Correia, 1996).

Deste modo, para qualquer  $t \geq 0$ , sabe-se que:

$$\begin{aligned}
 P = \{u > t\} &= P \{T_1 > t, T_2 > t, \dots, T_n > t\} \\
 \Leftrightarrow P &= P \{T_1 > t\} \times P \{T_2 > t\} \times \dots \times P \{T_n > t\} \\
 \Leftrightarrow P &= e^{-\alpha_1 t} \times e^{-\alpha_2 t} \times \dots \times e^{-\alpha_n t} \\
 \Leftrightarrow P &= e^{-(\alpha_1 + \alpha_2 + \dots + \alpha_n)t} \\
 \Leftrightarrow P &= e^{-\sum_{i=1}^n (\alpha_i)t} \tag{12}
 \end{aligned}$$

A partir de um determinado instante, o tempo que falta até à chegada de um cliente,  $U$ , tem uma distribuição exponencial negativa de parâmetro (Hillier & Lieberman, 1990):

$$\alpha = \sum_{i=1}^n \alpha_i \tag{13}$$

Esta propriedade tem determinadas implicações relativamente aos tempos entre chegadas.

Ao supor-se que existem diferentes tipos de clientes ( $n$ ), assume-se que os tempos entre chegadas para cada tipo (tipo  $i$ ) seguem uma distribuição exponencial com parâmetro  $i$  ( $i = 1, 2, \dots, n$ ).

A propriedade 3 considera que o tempo entre chegadas ( $U$ ) segue uma distribuição exponencial de parâmetro  $\alpha$ , definido na última equação. No entanto, é mais relevante a informação sobre o tempo de serviço em modelos de filas com mais de um servidor do que para tempos entre chegadas (Hillier & Lieberman, 1990).

4. “Relação com a distribuição de Poisson” (Hillier & Lieberman, 1990).

Supõe-se que o tempo entre consecutivas ocorrências de um determinado evento (isto é, cadência de chegada ou serviços concluídos por um servidor continuamente ocupado) segue uma distribuição exponencial de parâmetro  $\alpha$ . A propriedade 4 resulta da distribuição de probabilidades do número de ocorrências de um determinado tipo de evento ao longo de um horizonte temporal. Assume-se que  $X(t)$  é o número de ocorrências por tempo  $t$  ( $t > 0$ ), onde o tempo zero designa o instante inicial. Posto isto, obtém-se que (Hillier & Lieberman, 1990):

$$P\{X(t) = n\} = \frac{(\alpha t)^n e^{-\alpha t}}{n!} \text{ para } n = 0, 1, 2, \dots, n \quad (14)$$

Da equação (14), a variável  $X(t)$  segue uma distribuição de Poisson, com parâmetro  $\alpha t$ . Se  $n=0$ , obtém-se a probabilidade da distribuição exponencial, que corresponde ao primeiro evento que ocorre após a variável  $t$ , tal como se pode verificar na equação (15) (Hillier & Lieberman, 1990):

$$P\{X(t) = 0\} = e^{-\alpha t} \quad (15)$$

A média desta distribuição de Poisson é dada pela equação (16) (Hillier & Lieberman, 1990):

$$E\{X(t)\} = \alpha t \quad (16)$$

A variável  $\alpha$  traduz o número esperado de eventos por unidade de tempo, sendo a taxa média a que os eventos ocorrem. Quando os eventos são estudados de forma contínua, o processo segue uma distribuição de Poisson com parâmetro  $\alpha$  (Hillier & Lieberman, 1990).

Esta propriedade fornece informações úteis sobre a finalização de um determinado serviço, quando os tempos de serviço seguem uma distribuição exponencial com parâmetro  $\mu$ . Esta informação é obtida através da definição de  $X(t)$  com o número de serviços terminados por um servidor que está continuamente ocupado, num horizonte temporal específico,  $t$ , em que  $\alpha = \mu$ . Para múltiplos servidores, num sistema de filas de espera,  $X(t)$  também pode ser definido como o número de serviços concluídos por  $n$  servidores continuamente ocupados, onde  $\alpha = n\mu$  (Hillier & Lieberman, 1990).

Esta propriedade é bastante útil para descrever o comportamento probabilístico de chegadas quando os tempos entre estas seguem uma distribuição exponencial com parâmetro  $\lambda$ . Nesta situação,  $X(t)$  trata-se do número de chegadas de clientes, num horizonte temporal, com variável  $t$ , em que  $\alpha = \lambda$  é a taxa média de chegada. As chegadas seguem uma distribuição de Poisson, de parâmetro  $\lambda$  (Hillier & Lieberman, 1990).

5. “Para todos os valores positivos de  $t$ ,  $P\{T \leq t + \Delta t \mid T > t\} \approx \alpha \Delta t$ ” (Hillier & Lieberman, 1990).

A propriedade 5 revela que quando o valor de  $\Delta t$  é pequeno, a constante de probabilidade é aproximadamente  $\alpha \Delta t$ . Ao considerar-se valores pequenos de  $\Delta t$ , a probabilidade é proporcional ao  $\Delta t$ , com factor de proporcionalidade  $\alpha$ . A variável  $\alpha$  é a taxa média que traduz a ocorrência dos eventos, pelo que o número esperado de eventos no intervalo de tempo  $\Delta t$  é exatamente  $\alpha \Delta t$ . A probabilidade de ocorrência de um evento é ligeiramente diferente deste valor, dado que existe a possibilidade de ocorrer mais do que um evento, que conduz a uma probabilidade desprezável, quando o valor de  $\Delta t$  é pequeno (Hillier & Lieberman, 1990).

Assim, o valor da probabilidade para um valor fixo do intervalo de tempo,  $\Delta t$  (com  $\Delta t > 0$ ) é traduzido pela equação (17) (Hillier & Lieberman, 1990):

$$P\{T \leq t + \Delta t \mid T > t\} = P\{T \leq \Delta t\} = 1 - e^{-\alpha \Delta t}, t \geq 0 \quad (17)$$

Posto isto, uma vez que a série de  $e^x$ , para qualquer expoente  $x$  é traduzida pela equação (18), obtém-se a equação (19) que traduz a propriedade 5.

$$e^x = 1 + x + \sum_{n=2}^{\infty} \frac{x^n}{n!} \quad (18)$$

$$P\{T \leq t + \Delta t \mid T > t\} = 1 - 1 + \alpha \Delta t - \sum_{n=2}^{\infty} \frac{(-\alpha t)^n}{n!} \approx \alpha \Delta t \quad (19)$$

Na equação (19) os termos de soma são desprezáveis para pequenos valores da variável  $\Delta t$ .

A variável  $T$  representa os tempos de serviço ou os tempos entre chegadas de clientes, nos modelos de filas de espera. Esta propriedade consiste numa aproximação da probabilidade de ocorrência do evento em causa, no próximo intervalo de tempo ( $\Delta t$ ) (Hillier & Lieberman, 1990).

6. “Insensível à agregação e à desagregação” (Hillier & Lieberman, 1990).

Esta propriedade é importante principalmente para verificar e confirmar que o processo de entrada segue uma distribuição de Poisson, embora também seja aplicado diretamente à distribuição exponencial dada a propriedade 4, anteriormente citada.

Supõe-se que existem diferentes tipos de clientes ( $n$ ), dos quais chegam segundo um processo de Poisson, com parâmetros  $\lambda_i$  ( $i=1,2,\dots,n$ ). Ao considerar-se que estes são processos de Poisson independentes, esta propriedade evidencia o facto de que o processo de chegada agregada (chegada de todos os clientes sem ter em consideração o tipo destes) também segue uma distribuição de Poisson, com parâmetro (taxa de chegada)  $\lambda = \lambda_1 + \lambda_2 + \dots + \lambda_n$  (Hillier & Lieberman, 1990). Portanto, um processo de Poisson é insensível à agregação.

Esta propriedade revela estar diretamente relacionada com as propriedades 3 e 4. A propriedade 4 implica que os tempos entre chegadas de clientes de um determinado tipo,  $i$ , sigam uma distribuição exponencial com parâmetro  $\lambda_i$ . Com a propriedade 3 constatou-se que os tempos entre chegadas dos clientes também devem seguir uma distribuição exponencial, com o parâmetro  $\lambda = \lambda_1 + \lambda_2 + \dots + \lambda_n$  (Hillier & Lieberman, 1990).

A propriedade 6 indica que um processo de Poisson é insensível à desagregação, onde o processo de chegada segue uma distribuição de Poisson com parâmetro  $\lambda$ . No entanto, o ponto importante é a natureza dos processos de chegada desagregados, para os diferentes tipos de clientes. Sabendo que cada cliente tem como probabilidade,  $p_i$ , com  $i = 1,2,\dots,n$ , então seguem-se as equações (20) e (21) (Hillier & Lieberman, 1990).

$$\lambda_i = p_i \lambda \quad (20)$$

$$\sum_{i=1}^n p_i = 1 \quad (21)$$

Existem dois tipos de clientes - aqueles que se recusam a entrar no sistema e aqueles que entram no sistema. Esta propriedade diz que cada tipo de cliente chega de acordo com um processo de Poisson, com parâmetros  $p\lambda$  e  $(1-p)\lambda$ , respectivamente (Hillier & Lieberman, 1990).

### 3.1.6 O modelo M/M/S

Na análise de prestação de serviços, em que se verifica um número de servidores superiores a um, o modelo mais utilizado é o modelo marcoviano M/M/S.

Neste modelo, as chegadas de clientes caracterizam-se por seguirem uma distribuição de Poisson. Por outro lado, a distribuição exponencial caracteriza o tempo de serviço, pelo que o número de servidores do sistema é independente entre si. Assume-se que os clientes formam uma fila única.

No modelo M/M/S, as variáveis relativas à taxa de chegada ( $\lambda$ ) e à taxa de serviço ( $\mu$ ) traduzem o facto dos tempos de chegadas e de serviço apresentarem distribuições exponenciais. Assim sendo, para a taxa de chegada, a distribuição exponencial apresenta, para  $x>0$ , uma densidade de  $\lambda e^{-\lambda x}$ . Na taxa de serviço, a distribuição exponencial apresenta uma densidade de  $\mu e^{-\mu x}$ , igualmente para  $x>0$ .

No processo de nascimento e morte, inicialmente é estabelecido os valores para  $\lambda_n$  e  $\mu_n$ , quando estão  $n$  clientes no sistema. Neste sistema de filas, a taxa média de chegada e a taxa média de serviço por servidor ocupado são constantes, independentemente do estado do sistema. Quando o sistema tem um único servidor ( $s=1$ ), os parâmetros do processo de nascimento e morte são  $\lambda_n = \lambda$  ( $n = 0,1,2, \dots$ ) e  $\mu_n = \mu$  ( $n = 0,1,2, \dots$ ) (Hillier & Lieberman, 1990).

A taxa de serviço é constante para todos os servidores. A taxa de chegada não varia com o número de clientes no sistema, sendo constante (Tavares, Oliveira, Themido, & Correia, 1996).

### 3.1.7 O modelo M/M/S ( $s=1$ )

Quando se trata única e exclusivamente de um servidor, consideram-se determinados aspetos, no ritmo médio de chegadas e de serviços, como indicam as equações (22) e (23) (Hillier & Lieberman, 1990).

$$\lambda_n = \lambda, \text{ para } n = 0, 1, 2 \dots \tag{22}$$

$$\mu_n = n\mu, \text{ para } n \leq s \tag{23}$$

Na figura seguinte encontra-se o diagrama de estado para o modelo M/M/S com um único servidor.

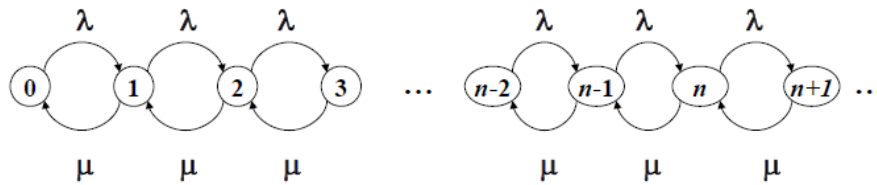


Figura 28 – Diagrama de estado para o modelo M/M/S, com um único servidor ( $s=1$ ) (adaptado: (Hillier & Lieberman, 1990))

Para um modelo de filas de espera M/M/S, com  $s=1$ , o fator  $C_n$  para os processos de nascimento e morte é reduzido para (Hillier & Lieberman, 1990):

$$C_n = \left(\frac{\lambda}{\mu}\right)^n = \rho^n \text{ (para } n=1,2,\dots) \tag{24}$$

A probabilidade no estado  $n$  é traduzida pela seguinte equação (Hillier & Lieberman, 1990):

$$P_n = \rho^n P_0 \text{ (para } n=1,2,\dots) \tag{25}$$

A probabilidade no estado zero é obtida segundo a seguinte equação (Hillier & Lieberman, 1990):

$$P_0 = \frac{1}{1 + \sum_{n=1}^{\infty} \rho^n} = (\sum_{n=0}^{\infty} \rho^n)^{-1} = \left(\frac{1}{1-\rho}\right)^{-1} = 1 - \rho \tag{26}$$

A probabilidade no estado  $n$ , sabendo a probabilidade no estado zero, é dada pela seguinte equação (Hillier & Lieberman, 1990):

$$P_n = (1 - \rho)\rho^n \text{ (para } n=0,1,2,\dots) \quad (27)$$

Consequentemente, o número de clientes no sistema,  $L$ , é dado pela equação seguinte:

$$L = \sum_{n=0}^{\infty} n (1 - \rho)\rho^n = (1 - \rho)\rho \sum_{n=0}^{\infty} \frac{d}{dp} (\rho^n) = (1 - \rho)\rho \frac{d}{dp} (\sum_{n=0}^{\infty} \rho^n) = (1 - \rho)\rho \frac{d}{dp} \left( \frac{1}{1-\rho} \right) = \frac{\rho}{1-\rho} = \frac{\lambda}{\mu-\lambda} \quad (28)$$

De forma similar, o comprimento da fila de espera (que exclui os clientes que estão a ser servidos),  $L_q$  é definido pela seguinte equação (Hillier & Lieberman, 1990):

$$L_q = \sum_{n=0}^{\infty} (n - 1) P_n = L - 1(1 - P_0) = \frac{\lambda^2}{\mu(\mu-\lambda)} \quad (29)$$

Quando  $\lambda \geq \mu$ , isto significa que a taxa média de chegada é superior à taxa de serviço. Verifica-se que a fila de espera irá “explodir” e irá crescer, tendencialmente, sem limites. Ao contrário, quando  $\lambda < \mu$ , é possível derivar a distribuição da probabilidade do tempo de espera no sistema (inclui tempo de serviço),  $W$ , para cada cliente, segundo o critério FIFO (*first in, first out*). Como consequência, obtém-se a equação (30) (Hillier & Lieberman, 1990).

$$P[W > t] = \sum_{n=0}^{\infty} P_n P\{S_{n+1} > t\} = e^{-\mu(1-\rho)t}, \text{ para } t \geq 0 \quad (30)$$

O tempo de espera no sistema (que inclui tempo de serviço),  $W$ , segue uma distribuição exponencial de parâmetro  $\rho(1 - \rho)$ . A equação que traduz o tempo de espera no sistema,  $W$  é (Hillier & Lieberman, 1990):

$$W = E(W) = \frac{1}{\mu(1-\rho)} = \frac{1}{\mu-\lambda} \quad (31)$$

Para  $t \geq 0$ , se  $n > 0$  clientes no sistema, o tempo de espera no sistema (excluindo o tempo de serviço),  $W_q$ , segue uma distribuição exponencial e traduz o tempo de espera até o atendimento começar. A probabilidade de  $W_q$  é definida pela seguinte equação (Hillier & Lieberman, 1990):

$$P[W_q > t] = \sum_{n=1}^{\infty} P_n P\{S_n > t\} = \sum_{n=1}^{\infty} (1 - \rho)\rho^n P\{S_n > t\} = \rho \sum_{n=1}^{\infty} P_n P\{S_{n+1} > t\} = \rho e^{-\mu(1-\rho)t} \quad (32)$$

Quando se deriva a média desta distribuição, obtém-se que (Hillier & Lieberman, 1990):

$$W_q = E(W_q) = \frac{\lambda}{\mu(\mu-\lambda)} \quad (33)$$

### 3.1.8 O modelo M/M/S ( $s > 1$ )

Para o modelo M/M/S, quando se trata de múltiplos servidores ( $s > 1$ ), são considerados determinados aspetos, como indicam as equações (34), (35) e (36) (Hillier & Lieberman, 1990).

$$\lambda_n = \lambda, \text{ para } n = 0, 1, 2 \dots \quad (34)$$

$$\mu_n = n\mu, \text{ para } n \leq s \quad (35)$$

$$\mu_n = s\mu, \text{ para } n > s \quad (36)$$

Na figura seguinte, encontra-se o diagrama de estado para o modelo M/M/S para múltiplos servidores.

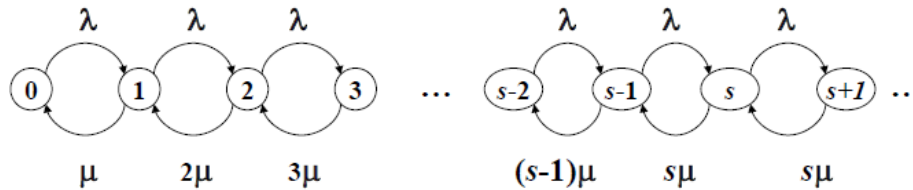


Figura 29 - Diagrama de estado para o modelo M/M/S, com s servidores (adaptado: (Hillier & Lieberman, 1990)).

Quando o número de servidores é superior a um, o fator  $C_n$  é definido como (Hillier & Lieberman, 1990):

$$C_n = \begin{cases} \frac{1}{n!} \left(\frac{\lambda}{\mu}\right)^n & \text{para } n = 1, 2, \dots, s \\ \frac{1}{s!} \left(\frac{\lambda}{\mu}\right)^s \left(\frac{\lambda}{s\mu}\right)^{(n-s)} = \frac{\left(\frac{\lambda}{\mu}\right)^n}{s! s^{(n-s)}} & \text{para } n = s, s + 1, \dots \end{cases} \quad (37)$$

Como consequência, se  $\lambda < s\mu$ , a probabilidade no estado zero é (Hillier & Lieberman, 1990):

$$P_0 = \left( \sum_{n=0}^{\infty} C_n \right)^{-1} = \left( \sum_{n=0}^{s-1} \frac{1}{n!} \left(\frac{\lambda}{\mu}\right)^n + \sum_{n=s}^{\infty} \frac{1}{s!} \left(\frac{\lambda}{\mu}\right)^s \left(\frac{\lambda}{\mu}\right)^{(n-s)} \right)^{-1} = \left( \sum_{n=0}^{s-1} \frac{1}{n!} \left(\frac{\lambda}{\mu}\right)^n + \frac{1}{s!} \left(\frac{\lambda}{\mu}\right)^s \sum_{m=0}^{\infty} \left(\frac{\lambda}{s\mu}\right)^m \right)^{-1}$$

Considera-se que  $m = n - s$ .

$$P_0 = \left( \sum_{n=0}^{s-1} \frac{1}{n!} \left(\frac{\lambda}{\mu}\right)^n + \frac{1}{s!} \left(\frac{\lambda}{\mu}\right)^s \sum_{m=0}^{\infty} \left(\frac{1}{1-\lambda/s\mu}\right)^m \right)^{-1} \quad (38)$$

A probabilidade no estado  $n$ ,  $P_n$  é o resultado entre a multiplicação do factor  $C_n$  e a probabilidade no estado zero,  $P_0$  (Hillier & Lieberman, 1990), sabendo que o estado de ocupação,  $\rho$  é definido como:

$$\rho = \frac{\lambda}{s\mu} < 1 \quad (39)$$

A probabilidade  $P_n$  é traduzida segundo as equações 40 e 41.

$$P_n = C_n P_0 \quad (40)$$

$$P_n = \begin{cases} \frac{1}{n!} \left(\frac{\lambda}{\mu}\right)^n P_0 & \text{para } 0 \leq n \leq s \\ \frac{(\frac{\lambda}{\mu})^n}{s! s^{(n-s)}} P_0 & \text{para } n \geq s \end{cases} \quad (41)$$

O comprimento da fila de espera (que exclui os clientes que estão a ser servidos) é o resultado do produto entre o ritmo médio de chegada de novos clientes por unidade de tempo,  $\lambda$ , e o tempo de espera no sistema (excluindo o tempo de serviço) de cada cliente,  $W_q$ , tal como se verifica na seguinte equação.

$$L_q = \lambda W_q \quad (42)$$

Para a obtenção da equação do número de clientes na fila de espera (excluindo os clientes que estão a ser servidos),  $L_q$ , procede-se às seguintes etapas de simplificação (Hillier & Lieberman, 1990):

$$L_q = E(n - s) = \sum_{n=s}^{\infty} (n - s) P_n \quad (43)$$

Considera-se que  $j = n - s$ ;  $n = s + j$ .

$$\begin{aligned} L_q &= \sum_{j=0}^{\infty} j P_{s+j} = \sum_{j=0}^{\infty} \frac{(\lambda/\mu)^s}{s!} \rho^j P_0 = P_0 \frac{(\lambda/\mu)^s}{s!} \sum_{j=0}^{\infty} j \rho^j = P_0 \frac{(\lambda/\mu)^s}{s!} \rho \sum_{j=0}^{\infty} \frac{d}{d\rho} (\rho^j) \\ \Leftrightarrow L_q &= P_0 \frac{(\lambda/\mu)^s}{s!} \rho \frac{d}{d\rho} \left( \sum_{j=0}^{\infty} \rho^j \right) = P_0 \frac{(\lambda/\mu)^s}{s!} \rho \frac{d}{d\rho} \frac{1}{1-\rho} = P_0 \frac{(\lambda/\mu)^s \rho}{s!(1-\rho)^2} \end{aligned} \quad (44)$$

O tempo de espera no sistema (excluindo o tempo de serviço) de cada cliente,  $W_q$  é dado por (Hillier & Lieberman, 1990):

$$W_q = \frac{L_q}{\lambda} \quad (45)$$

O tempo de espera no sistema (incluindo o tempo de serviço) de cada cliente,  $W$  é dado por (Hillier & Lieberman, 1990):

$$W = W_q + \frac{1}{\mu} \quad (46)$$

O número de clientes no sistema,  $L$ , é dado pela seguinte equação (Hillier & Lieberman, 1990) :

$$L = L_q + \frac{\lambda}{\mu} = \lambda W = \lambda \left( W_q + \frac{1}{\mu} \right) \quad (47)$$

A probabilidade do tempo de espera no sistema (incluindo o tempo de serviço) de cada cliente ser superior a um determinado horizonte temporal,  $t$ , é definido por (Hillier & Lieberman, 1990):

$$P[W > t] = e^{-\mu t} \left( \frac{1 + P_0 (\lambda/\mu)^s}{s!(1-\rho)} \left( \frac{1 - e^{\mu t (s-1-\lambda/\mu)}}{s-1-\lambda/\mu} \right) \right) \quad (48)$$

A probabilidade do tempo de espera no sistema (excluindo o tempo de serviço) de cada cliente ser superior a um determinado horizonte temporal,  $t$ , é definido por (Hillier & Lieberman, 1990):

$$P[W_q > t] = (1 - P[W_q = t])e^{-s\mu(1-\rho)t} \quad (49)$$

A probabilidade do tempo de espera no sistema (excluindo o tempo de serviço) de cada cliente ser igual a zero é traduzido pela seguinte equação (Hillier & Lieberman, 1990):

$$P[W_q = 0] = \sum_{n=0}^{s-1} P_n \quad (50)$$

### 3.2 Algoritmo de Dijkstra

O conceito de rede refere-se a um conjunto de nós, que estabelecem ligações entre si. Essa interligação permite a transferência de dados entre estes. Este problema é um processo de seleção do melhor caminho entre o nó de origem e o nó destino, de acordo com os dados transmitidos pelos nós (Miglani, Khera, & Aggarawal, 2016).

No ano de 1956, o cientista holandês Edsger Dijkstra desenvolveu um algoritmo, com o fim de resolver o problema de caminho mais curto. Publicado em 1959 e mais conhecido como o Algoritmo de Dijkstra, este algoritmo é fulcral na otimização de percursos, sendo de fácil implementação computacional (Goldbarg & Luna, 2000).

O algoritmo de Dijkstra é um dos principais algoritmos no processo de decisão do trajeto mais curto, isto é, na determinação do trajeto com a distância mínima entre o nó de origem e o nó de destino. Como exemplo, este algoritmo permite selecionar o caminho mais curto entre uma cidade, e todas as outras cidades, tendo em consideração que os nós da rede representam cada cidade (Miglani, Khera, & Aggarawal, 2016).

O algoritmo de Dijkstra consiste na determinação de trajetos ótimos do nó de origem para outros nós. Considera-se, neste algoritmo, que a rede ( $G$ ) é constituída por um conjunto de nós ou vértices,  $V$ , e por um conjunto de arcos positivos,  $A$ . O intervalo de nós, em que as distâncias mínimas ( $z_j$ ) são conhecidas, na iteração  $k$ , traduz-se por  $S$ . Inicialmente, sabe-se que o nó de origem se encontra no intervalo  $S$ , ( $S = \{1\}$ ). No ponto de origem é atribuído uma distância nula de  $z_1 = 0$  e aos nós precedentes é atribuído  $p_1 = 0$  (Dantzig & Thapa, 1997).

Por outras palavras, e com uma simbologia equivalente, o algoritmo de Dijkstra determina o caminho mais curto entre o nó  $m$  e os restantes nós da rede  $G = (V, A)$  de  $m$  nós. Para a obtenção desse objetivo, são considerados dois conjuntos  $S$  e  $\bar{S}$ , pelo que  $S \cap \bar{S} = \emptyset$  e  $S \cup \bar{S} = V - \{m\}$  que se referem aos conjuntos de nós com rótulos definitivos e com rótulos provisórios, respetivamente (Martins, Pascoal, Santos, & Joaquim J., 2006).

Este algoritmo tem aplicação em redes cujos arcos são dotados de comprimentos não negativos. É baseado num processo de rotulação dos nós da rede e na classificação dos respetivos rótulos (Barrico, 2016).

No seguimento desse processo de rotulação, a cada nó  $i$  é atribuído um rótulo  $[\xi_i, \pi_i]$ . Este rótulo pode ser permanente ou temporário. Isto traduz que:

1.  $[\xi_i, \pi_i]$  é rótulo permanente que representa o caminho mais curto de  $S$  para  $i$ . Assim sendo,  $\xi_i$  é o nó que antecede  $i$  no caminho mais curto de  $S$  para  $i$ , ao que o  $\pi_i$  indica o valor do caminho mais curto de  $S$  para  $i$ .
2.  $[\xi_i, \pi_i]$  é rótulo temporário que representa o caminho mais curto de  $S$  para  $i$ . Desta forma,  $\xi_i$  é o nó que antecede  $i$  no melhor caminho, até ao momento, de  $S$  para  $i$ , e  $\pi_i$  indica o valor do melhor caminho até ao momento, de  $S$  para  $i$ .

O rótulo temporário de um nó traduz um limite superior correspondente à distância mais curta de  $S$  a esse nó, dado que o caminho que lhe está associado não é obrigatoriamente o mais curto.

Este algoritmo resume o processo de rotulação dos nós da rede. De forma ordenada, inicia em  $S$ , mediante as distâncias de cada nó a  $S$ . Assim, o nó selecionado é o nó com rótulo temporário e com o menor valor de  $\pi$ . Este converte-se em nó permanente, para que os seus nós adjacentes sejam eliminados, de forma a atualizar os rótulos destes (temporários). No início do algoritmo, apenas  $S$  é permanente, pelo que o algoritmo só é finalizado quando não existirem rótulos temporários.

Em seguida são descritas, matematicamente, as etapas do algoritmo de Dijkstra:

- Etapa 1:

$[\xi_S, \pi_S] = [S, 0]$ , o que indica que o caminho mais curto para o nó  $S$  custa zero, e não possui nós intermédios.

$[\xi_i, \pi_i] = [S, C_{Si}]$ ,  $\forall i \in N - \{S\}$  e  $(S, i) \in A$

$[\xi_i, \pi_i] = [-, \infty]$ ,  $\forall i \in N - \{S\}$  e  $(S, i) \notin A$

Posto isto, os nós temporários traduzem o conjunto de nós com rótulos temporários, sendo que:

$$\text{Nós temporários} = N - \{S\}$$

Por outro lado, os nós permanentes são o conjunto de nós com rótulos permanentes, pelo que:

$$\text{Nós permanentes} = \{S\}$$

- Etapa 2:

Se os nós temporários =  $\emptyset$ , traduz que todos os nós têm rótulos permanentes.

Então STOP

$$K = \text{nós temporários tal que } \pi_k \text{ é mínimo (k: } \pi_k = \min \{ \pi_x, x \in \text{Temporários} \} )$$

$$\text{Nós temporários} = \text{temporários} - \{k\}$$

Nós permanentes = permanentes  $\cup \{k\}$ , em que k passou a nó permanente.

- Etapa 3:

Para todo o  $j \in N$  tal que  $(k, j) \in \text{Temporários}$  Fazer

$$\text{Se } \pi_k + C_{kj} < \pi_j$$

Então

$$\pi_j = \pi_k + C_{kj}$$

$$\xi_j = k$$

Posto isto regressar à Etapa 2.

O principal objetivo deste algoritmo é determinar o caminho mais curto entre um nó específico (S) e todos os outros nós da rede.

Sendo assim, na etapa final do algoritmo, para verificar se existe um caminho entre S e qualquer nó k, analisa-se o valor de  $\pi_k$ . Não há caminho, caso  $\pi_k = \infty$ .

Caso exista caminho mais curto de S para k, este pode ser determinado ao percorrer (no sentido inverso) a primeira parte dos rótulos dos nós ( $\xi$ ) de k até S, do seguinte modo:

Caminho =  $\{k\}$ ,  $i = k$

Enquanto  $i \neq S$  Fazer

$i = \xi_i$

Caminho  $\leftarrow$  Caminho  $\cup \{i\}$  (Barrico, 2016)

### 3.3 O modelo proposto

O modelo a desenvolver tem como base o estudo e a análise na secção de inspeção e embalagem final, designada Escolha Final, da fábrica de Leiria 1. Em virtude dessa análise, o objetivo é a representação deste sistema.

A implementação do modelo, através do Matlab, leu ao desenvolvimento de uma interface gráfica, de forma a simular no sentido computacional, o meio em causa. Esta interface gráfica compreende não só caracterização da fila de espera das peças, mas também a caracterização dos seus percursos e a sua simulação baseada no algoritmo de Dijkstra.

Para esta implementação foi necessário, numa etapa primordial, compreender o funcionamento desta secção, desde a chegada dos vários tipos de peças, até à verificação de conformidade, e no final, a obtenção do produto final.

Existe uma diversidade de peças de cerâmica sanitária que são produzidas nesta fábrica. Desde as sanitas até aos lavatórios, são, no total, oito tipos:

- Sanita;
- Lavatório;
- Tanque;
- Tampa (parte integrante do tanque);
- Urinol;
- Bidé;
- Coluna (parte integrante do lavatório);
- Base de Chuveiro.

Neste estudo de simulação são analisados os três principais tipos de peças: a sanita, o lavatório e o tanque. Cada tipo de produto apresenta uma linha de processo, isto é, um conjunto de ações e tarefas associadas a ele. São inspecionados no devido posto de trabalho, tal como se verifica na tabela seguinte.

Quadro 3 – Tipo de produto e posto de trabalho correspondente.

<b>Tipo de Produto</b>	<b>Posto de trabalho</b>
Tanque	Passadeira 1
Sanita	Passadeira 2
Lavatório	Passadeira 3

O fluxograma seguinte corresponde ao percurso realizado pelo produto sanita.

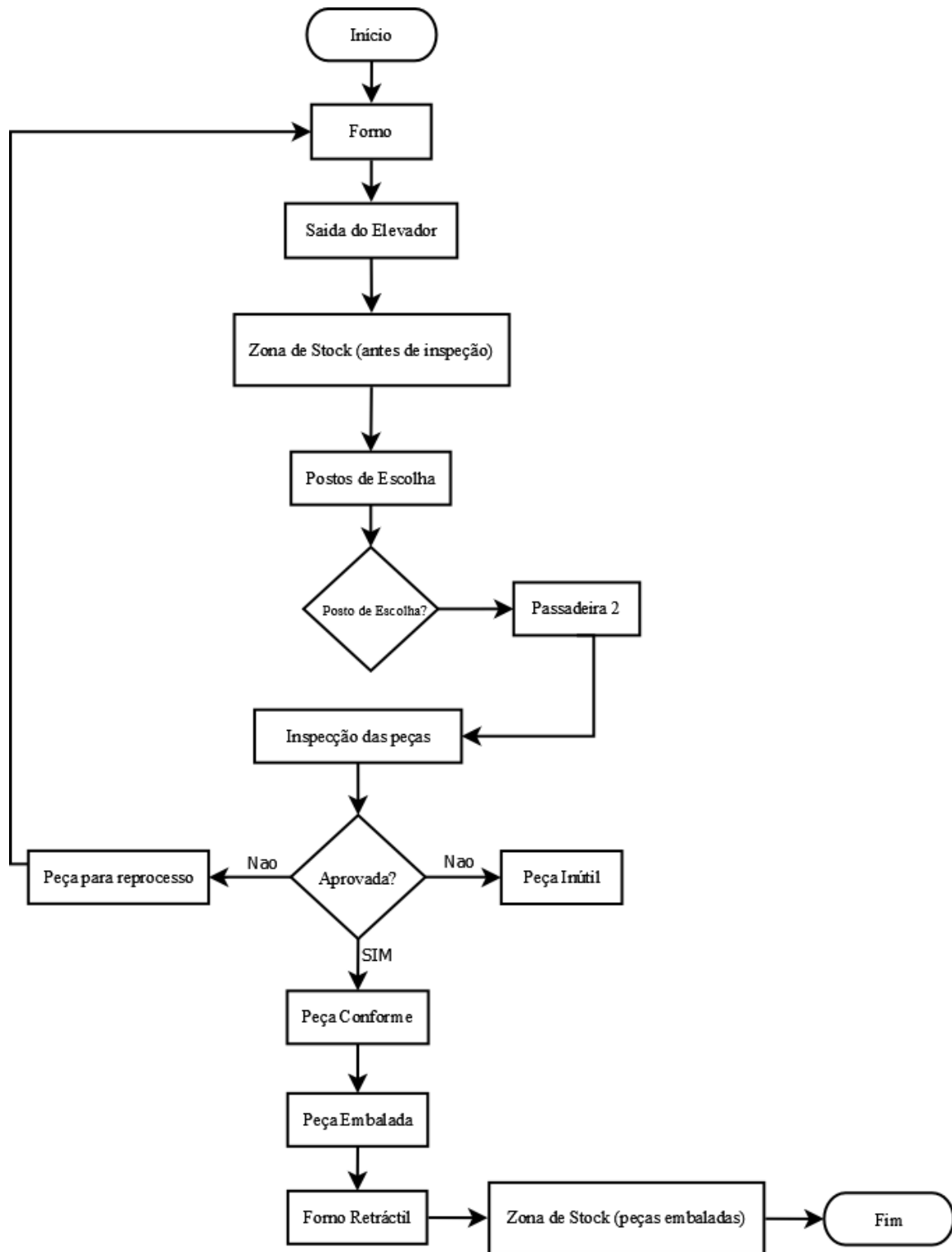


Figura 30 – Fluxograma do percurso da Sanita.

O fluxograma apresentado, de seguida, corresponde ao percurso realizado pelo produto lavatório.

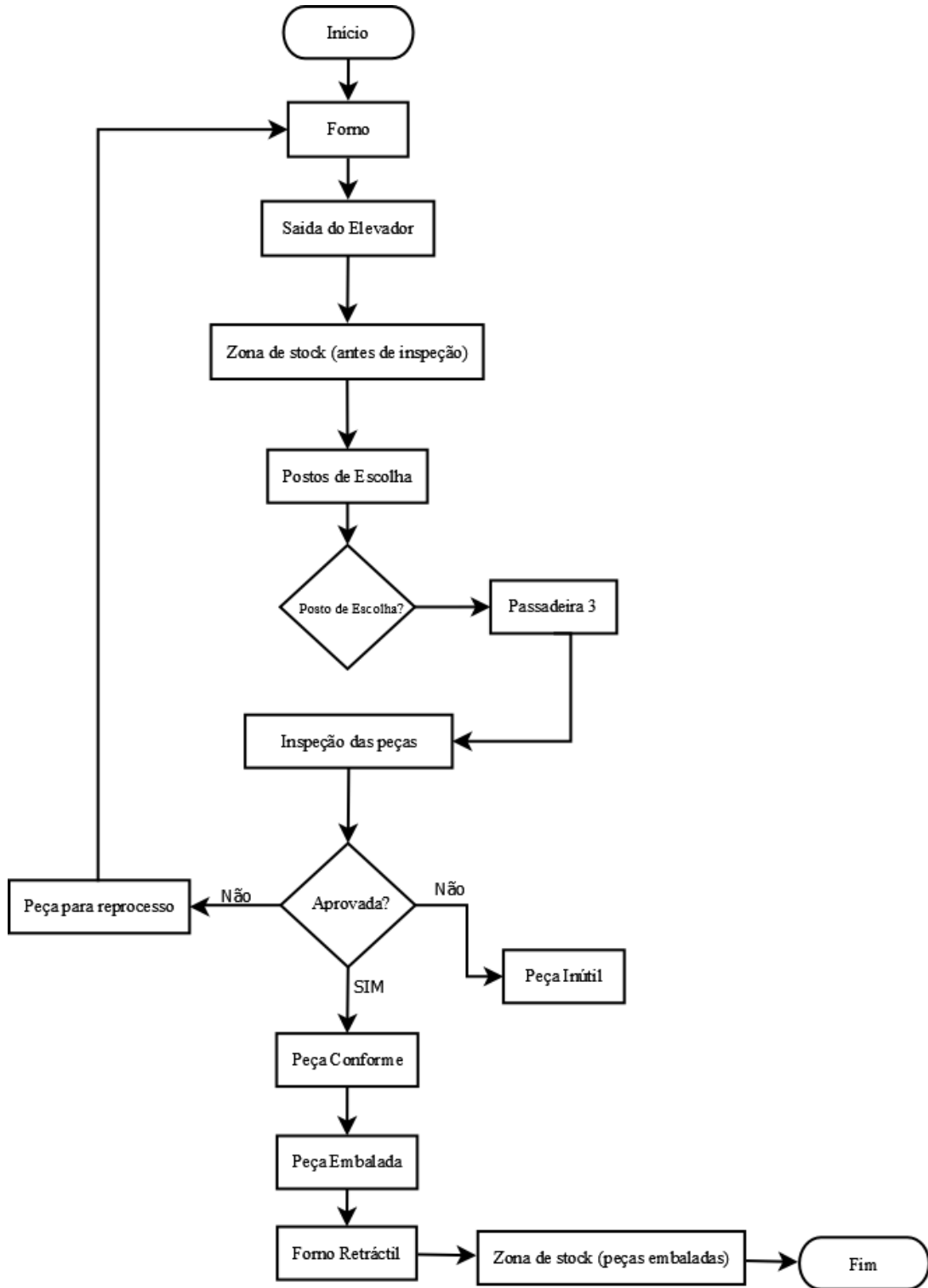


Figura 31 – Fluxograma do percurso do lavatório.

O fluxograma seguinte corresponde ao percurso realizado pelo produto tanque.

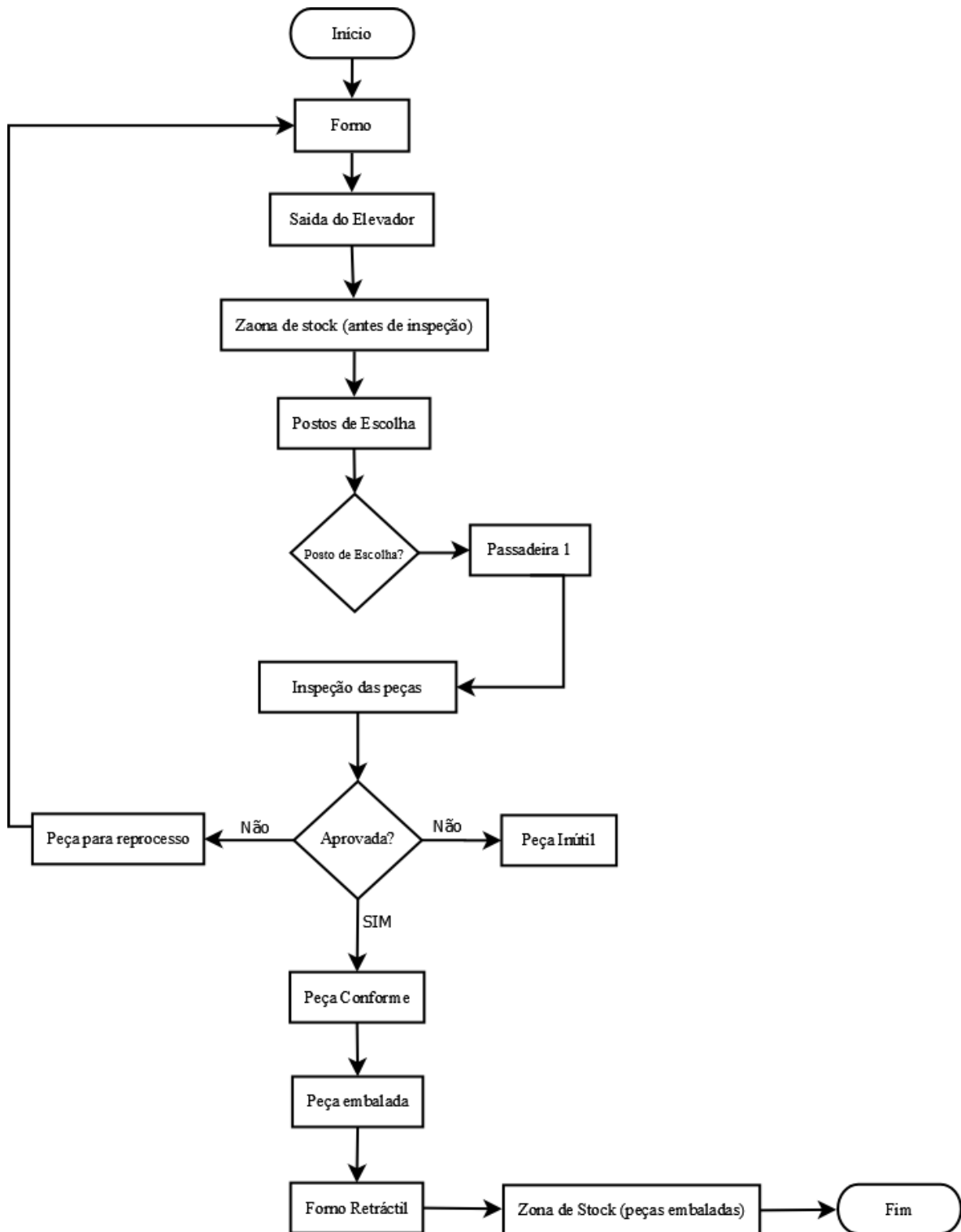


Figura 32 – Fluxograma do percurso da tanque.

De um modo geral e de forma resumida, as peças chegam até à secção Escolha Final, após o processo de cozedura. Uma vez que o forno se situa no piso superior, estas são transportadas através do elevador. Posteriormente, seguem para uma zona onde aguardam até serem inspecionadas – zona de stock (Figura 33). São inspecionadas e embaladas na passadeira estipulada, pelo que depois são transportadas até ao forno retrátil, e seguem para uma zona de stock, onde aguardam até à chegada do camião. Este meio de transporte envia as peças para o armazém de expedição.



Figura 33 – Zona de *stock* das peças (antes de inspeção).

### 3.3.1 Caracterização da fila de espera

A dimensão da população é o número total de peças que requisitam serviços do sistema. Na secção Escolha Final, a dimensão da população caracteriza-se por ser infinita, dado que o número de peças na secção é uma percentagem reduzida relativamente às restantes peças que estão inerentes ao processo produtivo.

No que toca à dimensão da chegada, observa-se que as peças chegam à secção em grupos de paletes, sendo que se caracterizam por serem chegadas incontroláveis.

O padrão estatístico, segundo o qual as peças se apresentam para inspeção e posterior paletização, segue uma distribuição de Poisson, pelo que a cadência de chegada de peças ao sistema é independente do número de peças presentes – população infinita. Isto é, a cadência de chegada das peças a um posto de trabalho segue uma distribuição de Poisson, se se verificar uma taxa média constante e independente entre estas.

Por outro lado, o tempo entre chegadas de peças de porcelana ao sistema segue uma distribuição exponencial.

Assume-se que a dimensão da fila de espera é infinita, sendo que a disciplina desta respeita o critério aleatório – a primeira peça a chegar, poderá não ser a primeira a ser inspecionada.

Para cada servidor especifica-se a distribuição de probabilidades dos tempos de serviço. Assim sendo, cada posto de trabalho ou passadeira segue uma distribuição de probabilidades exponencial.

Em virtude dos tópicos referidos anteriormente, o modelo de fila de espera a aplicar é o modelo marcoviano M/M/S, uma vez que as distribuições de tempos entre chegadas e de tempos de serviços seguem uma distribuição exponencial (Marcoviana). É aplicado, portanto, um modelo M/M/1 com um servidor. No quadro 4 encontram-se definidos os postos de trabalho, enquanto no quadro 5 encontra-se a correspondência de cada posto de trabalho para o tipo de peça a produzir.

Quadro 4 – Definição das zonas de trabalho.

Zona	Posto de trabalho na seção
S1	Passadeira 1
S2	Passadeira 2
S3	Passadeira 3
S4	Forno Retrátil
S5	Transportador de Abastecimento
S6	Transportador de saída
S7	Elevador

Quadro 5 - Correspondência de cada posto de trabalho para o tipo de peça a produzir.

	Sanita	Lavatório	Tanque
S1	○	○	●
S2	●	○	○
S3	○	●	○
S4	●	●	●
S5	●	●	●
S6	●	●	●
S7	●	●	●

● Sim      ○ Não

### 3.3.2 Caraterização do percurso

Na caracterização do percurso, observaram-se e estudaram-se os percursos reais de cada tipo de produto. O objetivo principal é a obtenção do caminho mais curto.

Em virtude desse objetivo, seguiu-se uma linha orientadora e metódica para a realização da simulação. Deste modo, a metodologia utilizada para a determinação do trajeto mais curto, com acesso à simulação em Matlab foi (Mohd Sabri, Hasan Basari, Husin, & Abu Samah, 2014):

1. Observar e analisar o *layout* em causa, isto é, estudar o plano a duas dimensões (2D):

Para tal, recorreu-se ao *software* Autocad, de fácil manipulação na análise dos nós e das distâncias entre os nós, bem como das coordenadas. Isto permitiu extrapolar o estudo real para um *layout* em 2D.

## 2. Construção da rede:

De acordo com o *layout* a duas dimensões analisado anteriormente, procede-se à construção da rede, constituída por nós e arcos (Anexo 2). Assim sendo, analisou-se o estado atual, isto é, o percurso atual para cada tipo de produto.

## 3. Implementação do Algoritmo de Dijkstra:

Este algoritmo foi incorporado no *software* Matlab, através da interface gráfica desenvolvida previamente e da qual incorporou-se um estudo e uma caracterização da fila de espera, como já foi referido anteriormente.

Recorreu-se à simulação para calcular a distância entre o nó de origem e o nó de destino, ou seja, de forma a calcular o trajeto mais curto entre os nós. Este algoritmo é utilizado como algoritmo padrão, uma vez que permite calcular e determinar de forma eficiente o trajeto mais curto. A implementação do algoritmo em Matlab compreende as seguintes etapas (Miglani, Khera, & Aggarawal, 2016)

1. Inserir o número de nós;
2. Proceder à elaboração da matriz de ligação entre os nós e da matriz dos valores de peso (nesta situação são os valores das distâncias);
3. Inserir o nó de origem;
4. Inserir o nó de destino;
5. Selecionar a matriz que contém os nós com o caminho direto ao nó de origem;
6. Calcular a distância do nó de origem para cada nó;
7. Calcular a distância mínima entre o nó de origem e o nó de destino.

Mediante estas etapas, procedeu-se, desta forma, à análise do sistema, através da recolha de dados.

### **3.4 Recolha de dados**

A recolha de dados, quer sejam registos históricos, quer sejam observações realizadas no sistema real carece de uma atenta análise para subsequente utilização na formulação do problema e do modelo. Os dados que existem na empresa estão no sistema de informação SAP e careceram de conversão para utilização no Matlab. Não existindo uma forma direta de migração de dados do SAP para o Matlab, optou-se por exportar para Microsoft Excel e posteriormente criação de procedimentos para acessibilidade pelo Matlab.

### **3.5 Dados para a caracterização da fila de espera e do percurso**

Para o estudo do sistema foi necessário obter determinados dados. Numa primeira etapa, recorreu-se à base de dados dos Fornos. Esta base de dados em Microsoft Access fornece informação relativamente às

peças que são colocadas no forno. Esta informação é uma previsão, uma vez que vinte e quatro horas depois de serem colocadas peças no forno, estas estarão no Departamento da Escolha Final. Conforme a figura seguinte, visualiza-se a cadência de chegada de peças do primeiro fogo, ao longo de 39 dias.

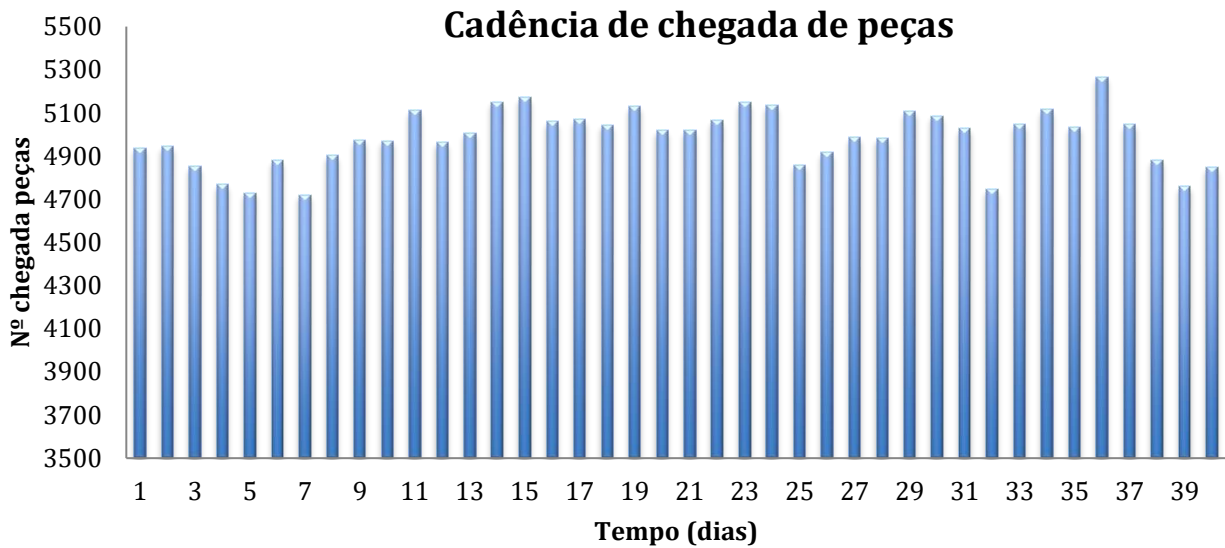


Figura 34 – Cadência de chegada de peças de porcelana ao departamento Escolha Final.

O valor de cadência de chegada é determinado através do cálculo da média, no horizonte temporal de 39 dias. Em virtude da cadência de chegada, tem-se em consideração que as peças são transportadas dos fornos, sendo que os seus colaboradores efetuam este transporte com destino à Escolha Final. Desse modo, os turnos da secção dos fornos são 3, pelo que cada turno corresponde a 7 horas e 15 minutos. A determinação média da cadência de chegadas, isto é, do ritmo médio de chegadas das peças, por unidade de tempo, depende do número de turnos estabelecidos previamente. A cadência de chegada corresponde ao  $\lambda$ .

É estabelecido o valor médio da cadência de chegada para os principais tipos de produto de cerâmica: a sanita, o tanque e o lavatório. Os dados encontram-se registados no quadro seguinte.

Quadro 6 – Definição e descrição dos nós, para o produto Sanita.

Tipo de produto	$\lambda$ (peças/dia)	$\lambda$ (peças/hora)
Sanita	725	33
Tanque	2185	100
Lavatório	1459	67

Através do ERP SAP obtiveram-se dados relativamente ao número de peças que são escolhidas por dia, assim como o número de peças que saem da secção para o armazém de expedição (armazém 0001), como se pode observar na figura seguinte. Com isto, o SAP permitiu, assim, a visualização do nível de produção diário.

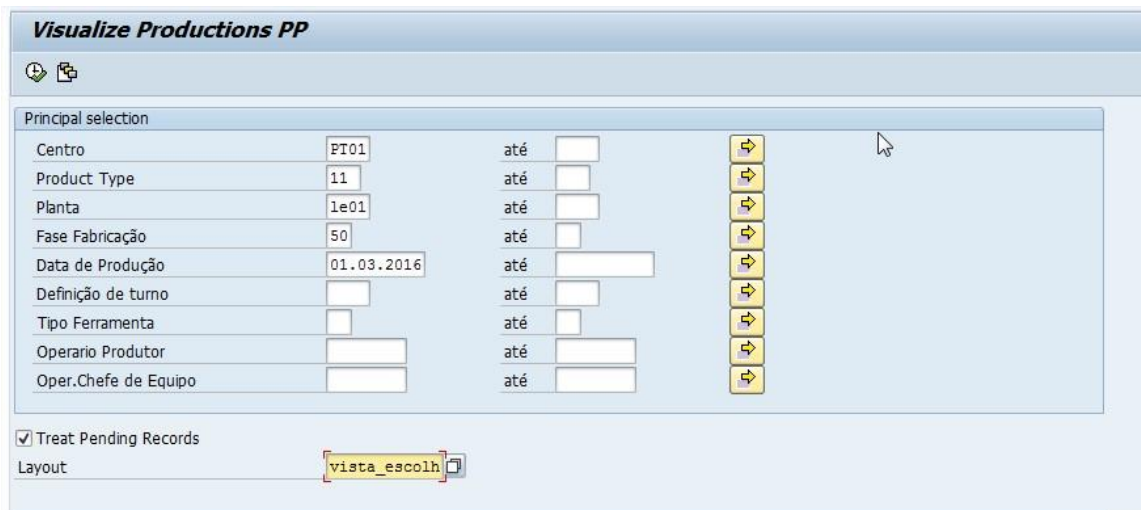


Figura 35 – Saída gráfica do ERP SAP.

Com o auxílio do ERP SAP, retiraram-se dados históricos referentes aos meses de Janeiro, Fevereiro, Março e Abril e 2016, do número de peças que foram escolhidas, avaliadas e classificadas como peças conformes (Anexo 1). No gráfico seguinte observam-se estes dados, em percentagem.

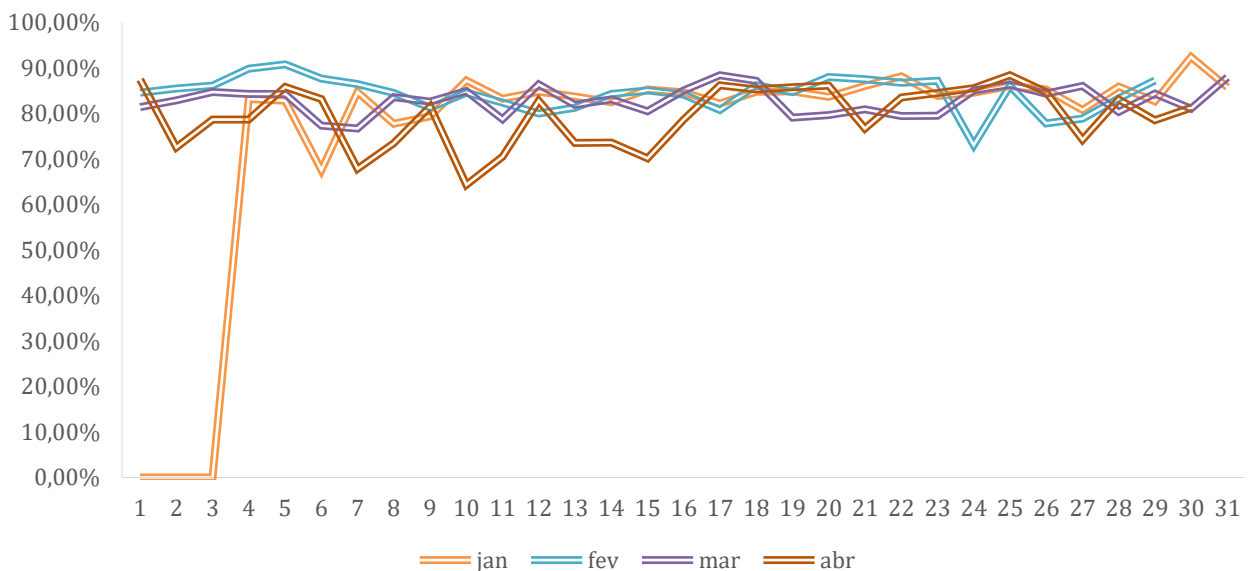


Figura 36 – Dados das peças conformes, em Janeiro, Fevereiro, Março e Abril.

De forma aleatória, e tendo em vista os dados históricos, foram selecionados 10 dias, para análise e caracterização da fila de espera e do percurso.

No Quadro 7, estão registados os dados relativos ao tipo de produto sanita e que correspondem às peças que foram inspecionadas, classificadas como conformes e embaladas, obtendo-se, deste modo o produto final. Os dados correspondem ao  $\mu_n$ , isto é, ao ritmo médio de serviço global do sistema (nº médio de peças que são escolhidas e avaliadas por unidade de tempo).

Quadro 7 – Dados relativos às peças boas, correspondentes ao tipo de produto sanita.

<b>Data</b>	<b><math>\mu</math>(peças/dia)</b>	<b><math>\mu</math> (peças/hora)</b>
12-01-2016	571	39
24-01-2016	670	46
07-02-2016	681	46
18-02-2016	515	35
21-03-2016	518	35
30-03-2016	605	41
04-03-2016	696	48
01-04-2016	805	55
27-04-2016	557	38
15-04-2016	661	45

No Quadro 8, estão registados os dados relativos ao tipo de produto tanque que correspondem às peças que foram inspecionadas, classificadas como conformes e embaladas, obtendo-se o produto final.

Quadro 8 – Dados relativos às peças boas, correspondentes ao tipo de produto tanques.

<b>Data</b>	<b><math>\mu</math> (peças/dia)</b>	<b><math>\mu</math> (peças/hora)</b>
12-01-2016	1962	135
24-01-2016	1786	123
07-02-2016	2718	187
18-02-2016	2039	140
21-03-2016	1777	122
30-03-2016	1602	110
04-03-2016	1984	136
01-04-2016	2120	146
27-04-2016	2073	142
15-04-2016	1892	130

No Quadro 9, estão registados os dados relativos ao tipo de produto lavatório, e que correspondem às peças que foram inspecionadas e classificadas como conformes, assim como, foram posteriormente embaladas, obtendo-se o produto final.

Para o estudo e caracterização do percurso, numa primeira etapa foi realizada uma análise ao estado atual, ou seja, aos percursos atuais de cada tipo de produto – para lavatório, tanque e sanita.

Nesse sentido, observou-se cada percurso real, desde o ponto de origem até ao ponto de destino de cada tipo de produto, tendo como base uma palete com 24 peças. Posteriormente, com o auxílio do *layout* da secção (plano a duas dimensões) no *software* Autocad, traçou-se o percurso correspondente ao estado atual (Anexo 2).

Quadro 9 – Dados relativos às peças boas, correspondentes ao tipo de produto lavatórios.

<b>Data</b>	<b><math>\mu</math> (peças/dia)</b>	<b><math>\mu</math> (peças/hora)</b>
12-01-2016	1460	100
24-01-2016	1072	73
07-02-2016	1067	73
18-02-2016	1206	83
21-03-2016	1007	69
30-03-2016	1462	100
04-03-2016	1220	84
01-04-2016	1939	133
27-04-2016	1039	71
15-04-2016	1095	75

Salienta-se que o objetivo é criar uma solução de baixos custos que permita otimizar o processo logístico, tendo para o efeito sido desenvolvida uma matriz espacial com coordenadas associadas à localização de cada objeto no espaço da fábrica. Este processo de *indoor mapping* (descrito no Anexo 5) pretende criar uma referência espacial de fácil programação com o Matlab. A partir das coordenadas de localização à saída do elevador estudaram-se e registaram-se inicialmente todos os trajetos e posteriormente partiu-se para a criação do caminho ótimo, minimizando assim o trajeto dos produtos dentro da fábrica.

Após a observação e análise do percurso real e atual deste tipo de produto, definiram-se os nós e os arcos, mediante o percurso observado na secção.

Para o primeiro produto analisado, o tanque, foram definidos os nós, tal como se pode verificar no quadro seguinte:

Quadro 10 – Definição e descrição dos nós, para o produto tanque.

<b>Nó</b>	<b>Descrição do nó</b>
<b>1</b>	Elevador.
<b>2</b>	Zona de <i>stock</i> para peças a inspecionar.
<b>5</b>	<i>Robot</i> 1 (entrada na passadeira).
<b>6</b>	<i>Robot</i> 2 (saída da passadeira).
<b>7</b>	Entrada do forno retrátil.
<b>8</b>	Saída do forno retrátil.
<b>9</b>	Zona de <i>stock</i> para peças embaladas
<b>11</b>	Saída da secção

No Quadro seguinte, estão registados os nós e as distâncias correspondentes ao produto tanque.

Quadro 11 – Distância entre os diferentes nós, para o produto tanque.

<b>Nó inicial</b>	<b>Nó de chegada</b>	<b>Distância (m)</b>
<b>1</b>	<b>2</b>	35
<b>2</b>	<b>5</b>	11
<b>5</b>	<b>6</b>	25
<b>6</b>	<b>7</b>	19
<b>7</b>	<b>8</b>	15
<b>8</b>	<b>9</b>	5
<b>9</b>	<b>11</b>	10

Após a obtenção do percurso atual de cada tipo de produto, procedeu-se a uma análise crítica de possíveis percursos para cada tipo. Desta análise, advém propostas de diferentes cenários. Esta proposta é extrapolada segundo um estudo de observação de tarefas de rotina na secção.

Como tal, delineou-se uma matriz, em Microsoft Excel, com coordenadas definidas, como se pode verificar no Anexo 2. Esta matriz vai ser utilizada de forma análoga nos outros tipos de produtos. Esta matriz serve de suporte na definição da localização dos nós.

Face ao acima exposto e especificamente para o tipo de produto tanque, foram propostos dois cenários – o cenário 1 e o cenário 2. A proposta destes dois cenários é com base em observações do sistema real.

No quadro seguinte, verificam-se os dados referentes dados referentes ao Cenário Atual, ao Cenário 1 e ao Cenário 2 do percurso de uma palete, cujo tipo de produto é o tanque.

Quadro 12 – Dados referentes ao Cenário Atual, Cenário 1 e Cenário 2 do percurso do tanque.

<b>Cenário Atual</b>			<b>Cenário 1</b>			<b>Cenário 2</b>		
<b>Nó (i)</b>	<b>Nó (f)</b>	<b>Distância (m)</b>	<b>Nó (i)</b>	<b>Nó (f)</b>	<b>Distância (m)</b>	<b>Nó (i)</b>	<b>Nó (f)</b>	<b>Distância (m)</b>
<b>1</b>	<b>2</b>	35	<b>1</b>	<b>3</b>	25	<b>1</b>	<b>4</b>	31
<b>2</b>	<b>5</b>	11	<b>3</b>	<b>5</b>	14	<b>4</b>	<b>5</b>	16
<b>5</b>	<b>6</b>	25	<b>5</b>	<b>6</b>	25	<b>5</b>	<b>6</b>	25
<b>6</b>	<b>7</b>	19	<b>6</b>	<b>7</b>	19	<b>6</b>	<b>7</b>	19
<b>7</b>	<b>8</b>	15	<b>7</b>	<b>8</b>	15	<b>7</b>	<b>8</b>	25
<b>8</b>	<b>9</b>	5	<b>8</b>	<b>9</b>	5	<b>8</b>	<b>10</b>	2
<b>9</b>	<b>11</b>	10	<b>9</b>	<b>11</b>	10	<b>10</b>	<b>11</b>	4

Da observação e registo destes dados, obtém-se o trajeto total para cada cenário.

No Quadro 13 encontram-se registados as distâncias totais, para o produto Tanque.

Quadro 13 – Distância Total, para cada cenário, do produto Tanque.

	<b>Cenário Atual</b>	<b>Cenário 1</b>	<b>Cenário 2</b>
<b>Distância Total (m)</b>	120	113	112

No seguimento dos dados registados, construiu-se a rede, constituída por nós e arcos. Esta traduz os percursos que o produto pode seguir. No fluxograma da figura seguinte, está representado a rede, com origem, no nó um (elevador) e com destino ao nó onze (saída da seção).

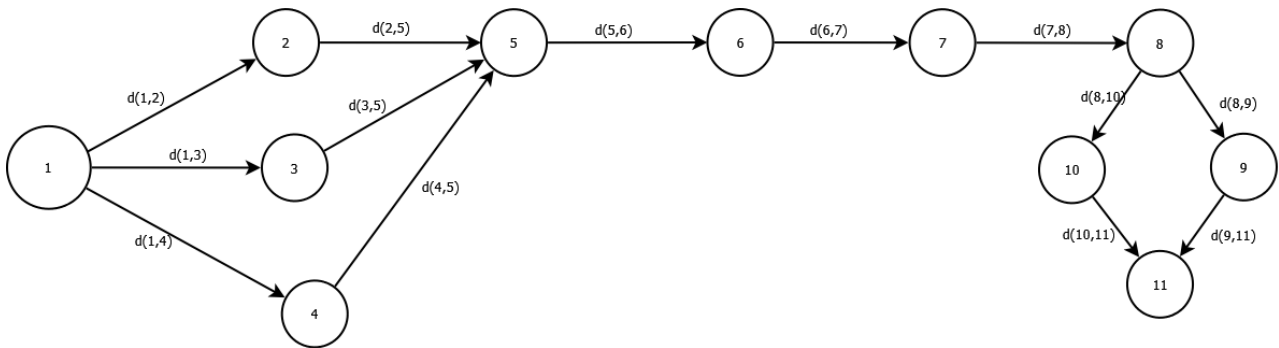


Figura 37 – Fluxograma do percurso do produto tanque.

Para o produto, a Sanita, foram definidos os nós, tal como se pode verificar no quadro seguinte.

Quadro 14 – Definição e descrição dos nós, para o produto Sanita.

Nó	Descrição do nó
1	Elevador.
2	Zona de <i>stock</i> para peças a inspecionar.
5	<i>Robot</i> 3 (entrada na passadeira).
6	<i>Robot</i> 4 (saída da passadeira).
7	Entrada do forno retrátil.
8	Saída do forno retrátil.
9	Zona de <i>stock</i> para peças embaladas
10	Saída da secção

No Quadro seguinte, estão registados os nós e as distâncias correspondentes ao produto sanita.

Quadro 15 – Distância entre os diferentes nós, para o produto sanita.

<b>Nó inicial</b>	<b>Nó de chegada</b>	<b>Distância (m)</b>
1	2	78
2	5	41
5	6	24
6	7	29
7	8	15
8	9	5
9	10	9

No quadro seguinte, apresenta-se o registo dos dados referentes ao Cenário Atual, ao Cenário 1 e ao Cenário 2 do percurso de uma palete, cujo tipo de produto é a Sanita.

Quadro 16 – Dados referentes ao Cenário Atual, Cenário 1 e Cenário 2 do percurso da Sanita.

<b>Cenário Atual</b>			<b>Cenário 1</b>			<b>Cenário 2</b>		
<b>Nó (i)</b>	<b>Nó (f)</b>	<b>Distância (m)</b>	<b>Nó (i)</b>	<b>Nó (f)</b>	<b>Distância (m)</b>	<b>Nó (i)</b>	<b>Nó (f)</b>	<b>Distância (m)</b>
<b>1</b>	<b>2</b>	78	<b>1</b>	<b>3</b>	69	<b>1</b>	<b>4</b>	42
<b>2</b>	<b>5</b>	41	<b>3</b>	<b>5</b>	35	<b>4</b>	<b>5</b>	6
<b>5</b>	<b>6</b>	24	<b>5</b>	<b>6</b>	24	<b>5</b>	<b>6</b>	24
<b>6</b>	<b>7</b>	29	<b>6</b>	<b>7</b>	29	<b>6</b>	<b>7</b>	29
<b>7</b>	<b>8</b>	15	<b>7</b>	<b>8</b>	15	<b>7</b>	<b>8</b>	15
<b>8</b>	<b>9</b>	5	<b>8</b>	<b>9</b>	5	<b>8</b>	<b>9</b>	5
<b>9</b>	<b>10</b>	9	<b>9</b>	<b>10</b>	9	<b>9</b>	<b>10</b>	9

No Quadro 17, encontram-se registados as distâncias totais, para o produto sanita.

Quadro 17 – Distância Total, para cada cenário, do produto Sanita.

	<b>Cenário Atual</b>	<b>Cenário 1</b>	<b>Cenário 2</b>
<b>Distância Total (m)</b>	201	186	130

No seguimento dos dados registados, construiu-se a rede, constituída por nós e arcos. Esta traduz os percursos que o produto pode seguir. No fluxograma da figura seguinte, está representado a rede, com origem, no nó um (elevador) e com destino ao nó dez (saída da seção).

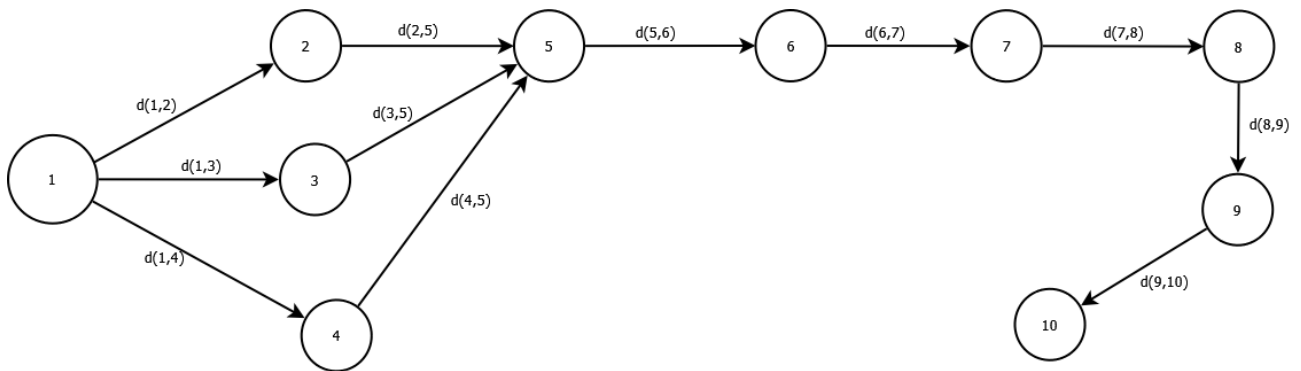


Figura 38 – Fluxograma do percurso do produto sanita.

Por fim, relativamente ao produto Lavatório, foram definidos os nós, como se pode verificar no quadro seguinte:

Quadro 18 – Definição e descrição dos nós, para o produto Lavatório.

<b>Nó</b>	<b>Descrição do nó</b>
<b>1</b>	Elevador.
<b>2</b>	Zona de <i>stock</i> para peças a inspecionar.
<b>5</b>	Plataforma 1 (antes de ir para a cinta).
<b>6</b>	Entrada na passadeira.
<b>7</b>	Saída da passadeira.
<b>8</b>	Plataforma 2 (durante a embalagem).
<b>9</b>	Entrada do forno retrátil.
<b>10</b>	Saída do forno retrátil.
<b>11</b>	Zona de <i>stock</i> para peças embaladas.
<b>13</b>	Saída da secção.

No Quadro seguinte, encontram-se registados os nós e as distâncias correspondentes ao produto lavatório.

Quadro 19 – Distância entre os diferentes nós, para o produto lavatório.

<b>Nó inicial</b>	<b>Nó de chegada</b>	<b>Distância (m)</b>
1	2	52
2	5	10
5	6	2
6	7	16
7	8	6
8	9	53
9	10	16
10	11	7
11	13	14

No Quadro 20, estão registados os dados referentes ao Cenário Atual, ao Cenário 1 e ao Cenário 2 do percurso de uma palete, cujo tipo de produto é o lavatório.

Quadro 20 – Dados referentes ao Cenário Atual, Cenário 1 e Cenário 2 do percurso do lavatório.

<b>Cenário Atual</b>			<b>Cenário 1</b>			<b>Cenário 2</b>		
<b>Nó (i)</b>	<b>Nó (f)</b>	<b>Distância (m)</b>	<b>Nó (i)</b>	<b>Nó (f)</b>	<b>Distância (m)</b>	<b>Nó (i)</b>	<b>Nó (f)</b>	<b>Distância (m)</b>
<b>1</b>	<b>2</b>	52	<b>1</b>	<b>3</b>	60	<b>1</b>	<b>4</b>	55
<b>2</b>	<b>5</b>	10	<b>3</b>	<b>5</b>	7	<b>4</b>	<b>6</b>	11
<b>5</b>	<b>6</b>	2	<b>5</b>	<b>6</b>	2	<b>6</b>	<b>7</b>	16
<b>6</b>	<b>7</b>	16	<b>6</b>	<b>7</b>	16	<b>7</b>	<b>8</b>	6
<b>7</b>	<b>8</b>	6	<b>7</b>	<b>8</b>	6	<b>8</b>	<b>9</b>	53
<b>8</b>	<b>9</b>	53	<b>8</b>	<b>9</b>	53	<b>9</b>	<b>10</b>	16
<b>9</b>	<b>10</b>	16	<b>9</b>	<b>10</b>	16	<b>10</b>	<b>12</b>	8
<b>10</b>	<b>11</b>	7	<b>10</b>	<b>12</b>	8	<b>12</b>	<b>13</b>	2
<b>11</b>	<b>13</b>	14	<b>12</b>	<b>13</b>	2			

No seguinte quadro encontram-se registados as distâncias totais, para o produto lavatório:

Quadro 21 – Distância Total, para cada cenário, do produto lavatório.

	<b>Cenário Atual</b>	<b>Cenário 1</b>	<b>Cenário 2</b>
<b>Distância Total (m)</b>	176	170	167

No seguimento dos dados registados, construiu-se a rede, constituída por nós e arcos. Esta traduz os percursos que o produto pode seguir. No fluxograma da figura seguinte, está representado a rede, com origem, no nó um (elevador) e com destino ao nó dez (saída da seção).

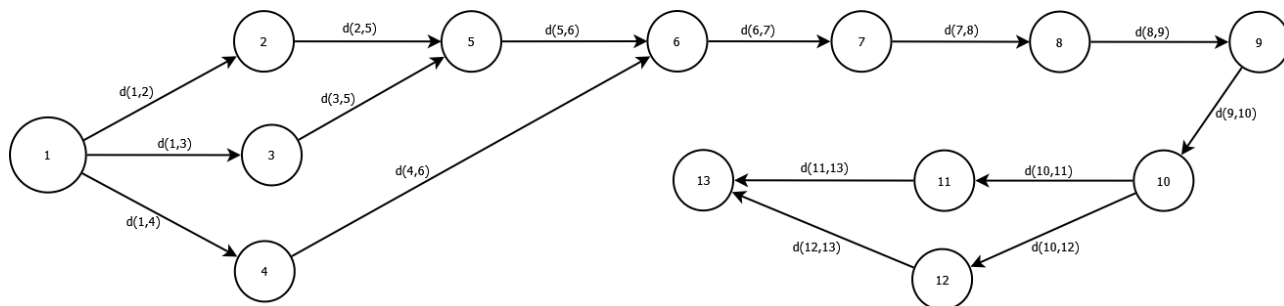


Figura 39 – Fluxograma do percurso do produto sanita.



## 4. Implementação

### 4.1 Tratamento de dados

Na implementação do modelo utilizou-se o *software* Matlab, como já foi referido anteriormente, para o desenvolvimento de uma interface gráfica. A construção, o desenvolvimento e a implementação da ferramenta em Matlab tem um papel coadjuvante no tratamento de dados, dada as suas significativas potencialidades. A elaboração desta interface só é possível com um processo de análise e observação, previamente realizado, com o intuito de enquadrar, executar e automatizar cálculos e informações, bem como agilizar de um modo simples os *inputs* e *outputs*.

Nesta fase do projeto, após os registos de dados, de observações e limitações do sistema, assim como o seu plano bidimensional é exequível iniciar o processo de *design* da interface gráfica. Assim, de um modo simples, o Matlab permitiu a procura de informações, a recolha de dados de diferentes perspetivas, no sentido de fornecer flexibilidade no tratamento e interpretação de dados. Esta flexibilidade está relacionada na facilidade de seleção de dados de uma lista ou de um menu (Marchand & Holland, 2003).

De forma resumida, a interface gráfica é constituída por elementos de controlo, com a finalidade de executar uma ação, ou configurar as opções para um evento futuro. Esta ação é executada quando é selecionado o elemento “*uicontrol*”. As figuras seguintes representam o *layout* da interface desenvolvida, e *layout* final da interface gráfica, quando esta é executada, respetivamente.

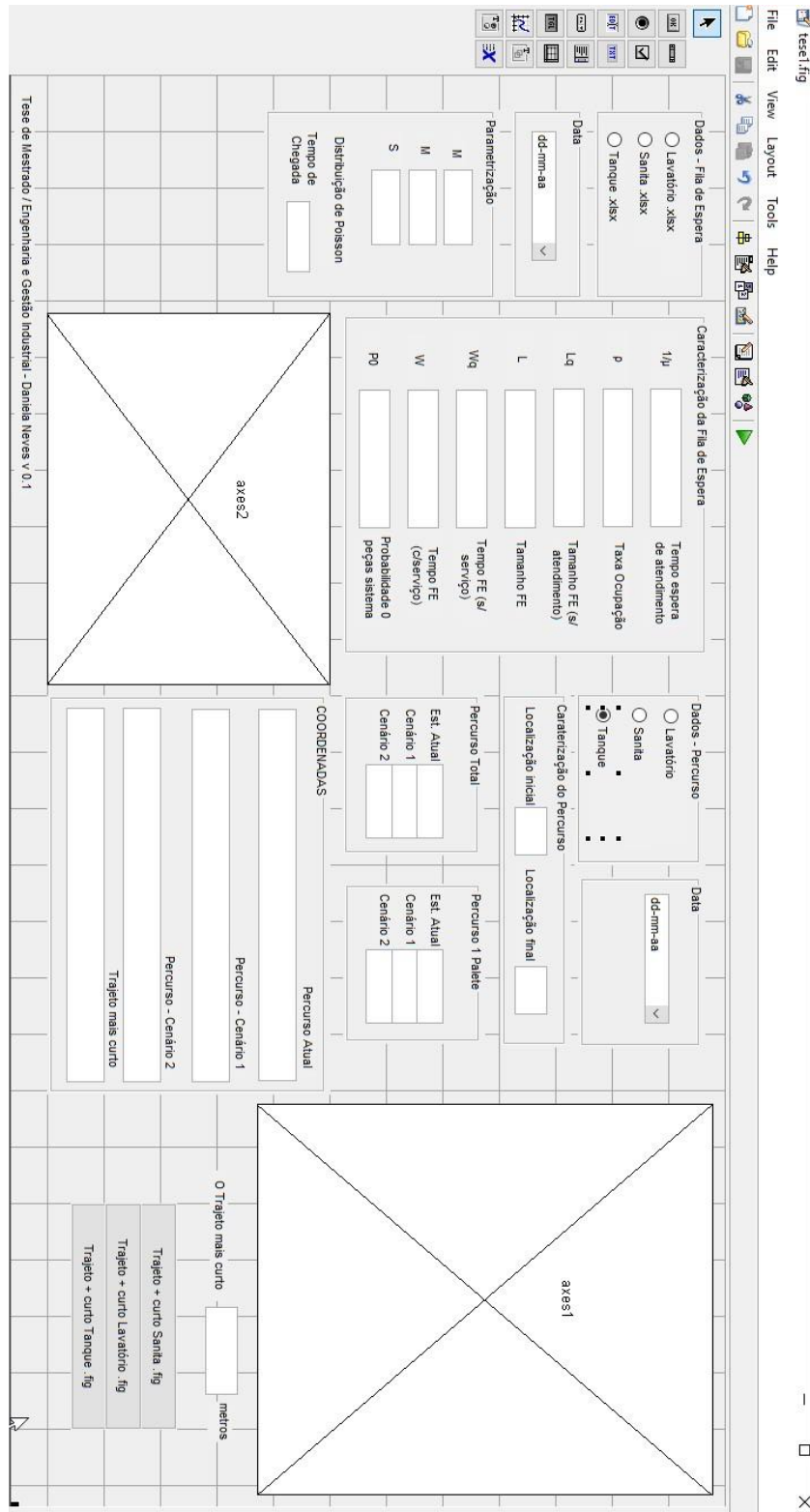


Figura 40 – Desenvolvimento do *layout* da interface gráfica que corresponde à ferramenta de simulação computacional.

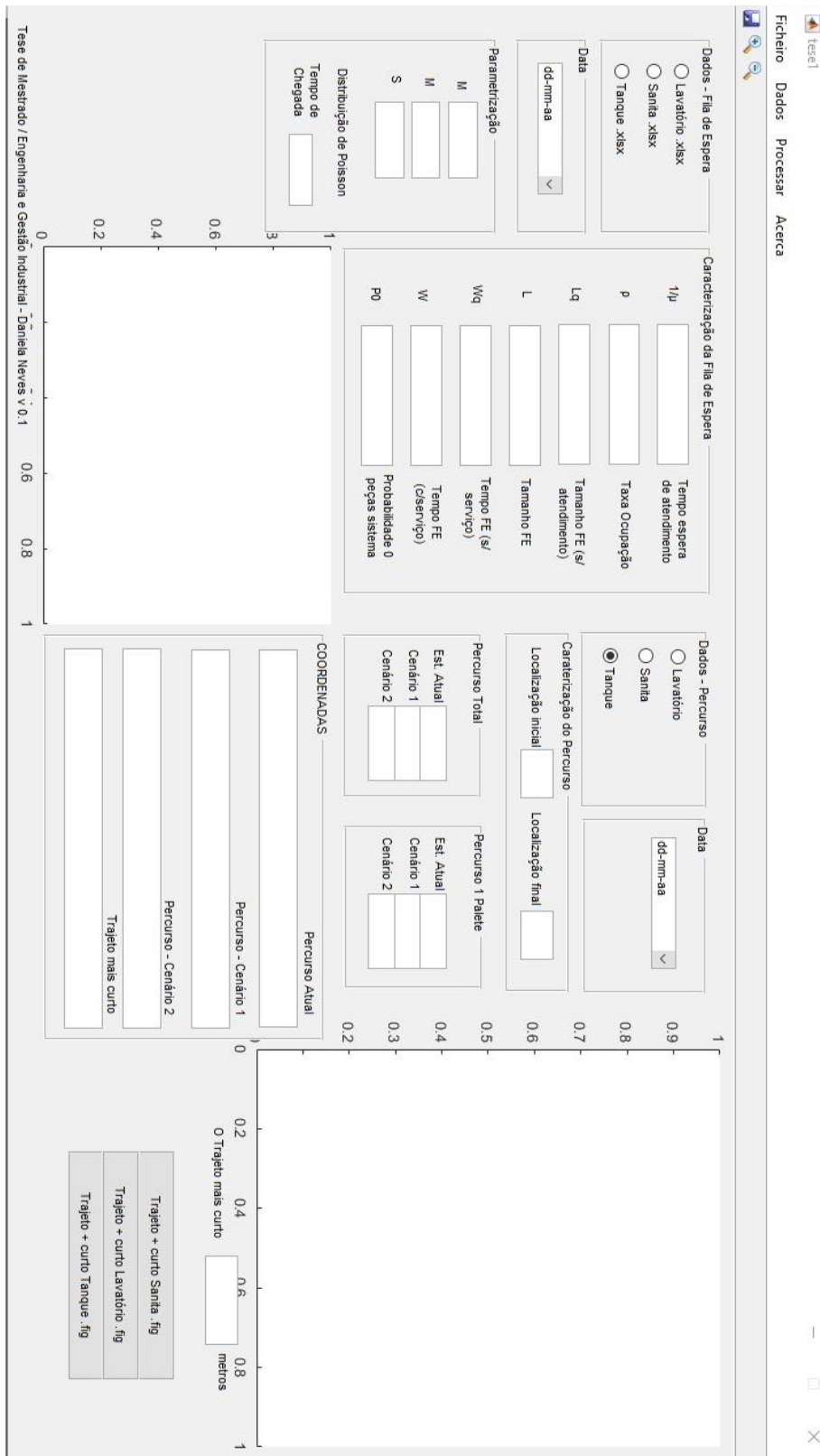


Figura 41 – Layout da interface gráfica que corresponde à ferramenta de simulação computacional.

Com isto, para o tratamento de dados, a utilização desta interface permitiu a seleção dos dados de cada tipo de produto, assim como a data correspondente aos dados selecionados. Como se observa na figura seguinte, a interface gráfica permite a importação de dados. A execução de cada botão permite, deste modo, a importação do ficheiro *Dados.xlsx*, em Microsoft excel. Esta ação tem em consideração a data. A data encontra-se inserida no menu. Os dados obtidos serão utilizados para a caracterização da fila de espera do sistema em causa.



Figura 42 – *Layout* da interface gráfica que corresponde à seleção de dados históricos, para a caracterização da fila de espera.



Figura 43 – *Layout* da interface gráfica que corresponde à seleção do tipo de produto, para a caracterização da fila de espera.

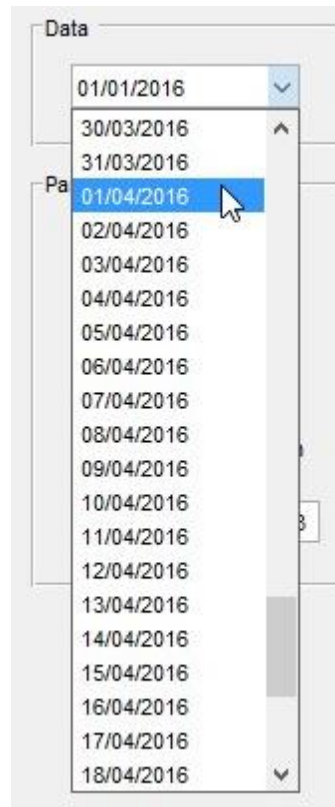


Figura 44 – *Layout* da interface gráfica que corresponde à seleção de dados de uma determinada data, para a caracterização da fila de espera.

Durante o desenvolvimento da interface gráfica definiram-se os parâmetros para a caracterização da fila de espera: a cadência de chegada ( $M$ ), o número de peças embaladas ( $M$ ) e o número de servidores ( $S$ ). Estes são obtidos através da importação de dados no ficheiro *Dados.xlsx*. O tempo de chegada, com uma distribuição de Poisson, é obtido através de cálculo numérico. Este parâmetro depende da cadência de chegada, isto é, do número médio de peças que chegam à Escolha Final, sendo inversamente proporcional. O código que traduz a obtenção do tempo de chegada encontra-se representado no quadro (22).

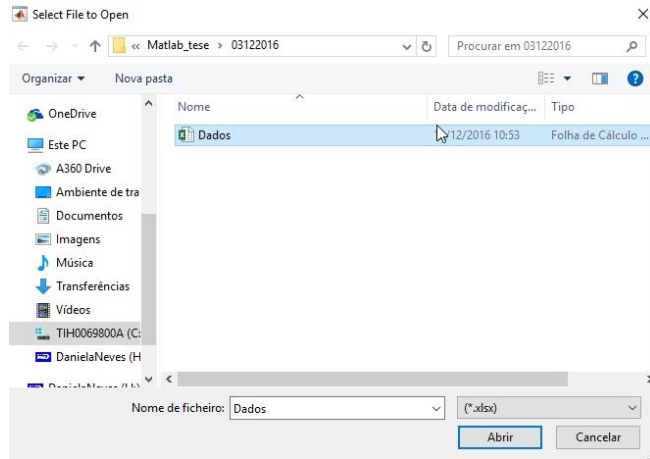


Figura 45 – Seleção do ficheiro em Microsoft Excel (*Dados.xlsx*).

Quadro 22 – Parte integral de código em *Matlab* para obter o tempo de chegada.

```
%Equação 1: Tempos entre chegadas  
T_cheg=1/lambda  
set(handles.edit13,'String',T_cheg)
```

A figura seguinte representa parte da interface gráfica, com a parametrização da fila de espera para os produtos em causa, e também o tempo de chegada das peças. Este caracteriza-se por seguir uma Distribuição de Poisson.



Figura 46 – *Layout* da interface gráfica que corresponde à parametrização da fila de espera.

Para o tratamento e análise do comportamento dos dados, esta interface permitiu a obtenção de importantes medidas de desempenho da fila de espera. No Quadro apresentado *a posteriori* encontram-se as

medidas de desempenho da caracterização da fila de espera, com parte integral do código em *Matlab* utilizado para obtenção destas medidas.

Quadro 23 – Parte integral de código em *Matlab* correspondentes à caracterização da fila de espera.

Símbolo	Descrição	Código em Matlab
$P_0$	Probabilidade de zero peças no sistema.	<code>P0=1-Rho set(handles.edit11,'String',P0)</code>
$1/\mu$	Tempo de serviço.	<code>T_serv=1/mu set(handles.edit5,'String',T_serv)</code>
$\rho$	Taxa de ocupação.	<code>Rho=lambda/(mu*S) set(handles.edit6,'String',Rho)</code>
$L_q$	Tamanho da fila de espera (sem atendimento)	<code>Lq= L-1*(1-P0) set(handles.edit7,'String',Lq)</code>
$L$	Tamanho da fila de espera (com atendimento)	<code>L= lambda/(mu-lambda) set(handles.edit8,'String',L)</code>
$W_q$	Tempo na fila de espera (sem serviço)	<code>Wq=Lq/lambda set(handles.edit9,'String',Wq)</code>
$W$	Tempo na fila de espera (com serviço)	<code>W= 1/(mu-lambda) set(handles.edit10,'String',W)</code>

A figura seguinte representa parte da ferramenta de simulação. Nesta figura, observam-se os resultados de cada medida de desempenho, da fila de espera do sistema, em função dos dados históricos.

The image shows a graphical user interface window titled "Caracterização da Fila de Espera". It contains seven rows, each with a label on the left, an empty text input field in the center, and a descriptive label on the right. The labels and their descriptions are:  $1/\mu$  (Tempo espera de atendimento),  $\rho$  (Taxa Ocupação),  $L_q$  (Tamanho FE (s/ atendimento)),  $L$  (Tamanho FE),  $W_q$  (Tempo FE (s/ serviço)),  $W$  (Tempo FE (c/serviço)), and  $P_0$  (Probabilidade 0 peças sistema).

Figura 47 – *Layout* da interface gráfica que corresponde à caracterização da fila de espera.

Figura 48 – *Layout* da interface gráfica que corresponde à caraterização do percurso.

Para além da caraterização da fila de espera do sistema a analisar, utilizou-se, também, esta interface para o tratamento e análise de dados do percurso de cada tipo de peça. Tal como na caraterização da fila de espera, para o tratamento de dados dos percursos, importaram-se os dados do ficheiro *Dados.xlsx*, ao executar cada botão. Tal como na análise da fila de espera, esta ação tem em consideração a data correspondente, que se encontra inserida no menu. Os dados obtidos serão utilizados para a caraterização do percurso, com vista a determinar o percurso total, em função do dia. Deste modo, o percurso total de um determinado tipo de peça obtido será não só o estado atual, como também os dois cenários propostos. Na figura seguinte, observa-se parte do layout da interface, com os elementos que permitirão a obtenção dos dados dos percursos de cada tipo de produto, tal como a obtenção do trajeto mais curto, e a sua respetiva visualização.

Figura 49 – *Layout* da interface gráfica que corresponde à obtenção do trajeto mais curto de cada peça.

Com isto, é possível a obtenção dos resultados e a demonstração e análise da caraterização da fila de espera do sistema, bem como da caraterização do percurso de cada tipo de produto.

## 5. Demonstração e discussão de resultados

### 5.1 A fila de espera

Relativamente à caracterização da fila de espera, após o tratamento de dados, obtiveram-se os seguintes resultados, para o produto sanita, tendo em consideração os dados referentes aos 10 dias previamente selecionados.

Sabe-se que a cadência de chegada deste produto é, em média, de 33 peças por hora (ou seja, 725 peças por dia), e que o número de servidores corresponde a um. Trata-se, portanto, de um modelo M/M/1. Verifica-se uma sequência de tarefas, em que se encontra um colaborador a transportar a peça ao posto de trabalho e eventualmente 3 ou 4 colaboradores que estão no posto de trabalho a ajudar o primeiro já referenciado. Pode existir o colaborador que retira a peça do posto de trabalho e por fim o colaborador que se encontra na zona do forno retrátil. No entanto, estes elementos apenas ajudam a execução deste processo. Portanto, pode-se concluir que existe apenas um servidor, logo  $S=1$ . Na literatura, encontram-se vários exemplos de  $S=N$ , porém, teria que existir processamento paralelo das peças do mesmo tipo, o que no caso em apreço não existe. Existem, no entanto, linhas em paralelo na secção, mas como processam outros produtos, conclui-se que estamos na presença de  $S=1$ .

A interface gráfica permitiu obter e registar os resultados. No quadro seguinte, estão registados os resultados das medidas de desempenho do produto sanita: a probabilidade de todos os servidores estarem desocupados ( $P_0$ ), o tempo entre chegadas ( $1/\lambda$ ), o tempo de serviço ( $1/\mu$ ) e a taxa de ocupação ( $\rho$ ).

Quadro 24 – Obtenção de cinco medidas de desempenho de fila de espera do produto sanita, para 10 dias.

Data	$\mu$ (peças/h)	$P_0$	$1/\lambda$ (h)	$1/\mu$ (h)	$\rho$
12/01/2016	39	0,152	0,0299	0,025	0,848
24/01/2016	46	0,278	0,0299	0,022	0,722
07/02/2016	46	0,289	0,0299	0,021	0,711
18/02/2016	35	0,060	0,0299	0,028	0,940
04/03/2016	48	0,305	0,0299	0,021	0,695
21/03/2016	35	0,066	0,0299	0,028	0,934

30/03/2016	41	0,200	0,0299	0,024	0,800
01/04/2016	55	0,399	0,0299	0,018	0,601
15/04/2016	45	0,268	0,0299	0,022	0,732
27/04/2016	38	0,131	0,0299	0,026	0,869

Em relação ao produto Sanita, no dia 1 de abril de 2016, o número de peças inspecionadas e embaladas foram de 55 peças por hora (corresponde a 0,92 peças por minuto). Verifica-se que o tempo de serviço é menor ( $1/\mu$ ), com um valor de 0,018 horas (1 minuto e 5 segundos). Para além disso, verifica-se que apresenta uma taxa de ocupação,  $\rho$ , com valor de 0,601, sendo o menor valor, dos dez dias analisados. No dia 21 de março de 2016, verifica-se o menor valor de sanitas embaladas, dentro dos 10 dias selecionados. A taxa de atendimento ( $\mu$ ) é diretamente proporcional à probabilidade de todos os servidores estarem desocupados, obtendo-se, assim, um tempo de serviço maior ( $1/\mu$ ) com o valor de 0,028 horas (1 minuto e 45 segundos).

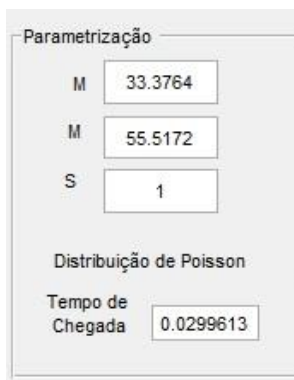


Figura 50 – Parametrização do dia 1 de abril de 2016, para o produto sanita.

Quadro 25 – Obtenção de seis medidas de desempenho de fila de espera do produto Sanita, para dez dias.

Data	Lq (peças)	L (peças)	Wq (h)	Wq (s)	W (h)	W (s)
12/01/2016	4,713	5,560	0,141	508,294	0,167	599,713
24/01/2016	1,879	2,601	0,056	202,672	0,078	280,582
07/02/2016	1,745	2,456	0,052	188,267	0,074	264,919
18/02/2016	14,651	15,591	0,439	1580,252	0,467	1681,611
04/03/2016	1,587	2,282	0,048	171,178	0,068	246,178

21/03/2016	13,282	14,217	0,398	1432,643	0,426	1533,415
30/03/2016	3,198	3,998	0,096	344,975	0,120	431,256
01/04/2016	0,906	1,507	0,027	97,751	0,045	162,596
15/04/2016	2,001	2,734	0,060	215,875	0,082	294,846
27/04/2016	5,757	6,626	0,172	620,944	0,199	714,661

A título de exemplo, na figura seguinte visualizam-se os resultados obtidos, para a caracterização a fila de espera, na interface gráfica, para o produto Sanita.



Figura 51 – Medidas de desempenho obtidas do dia 1 de abril de 2016, para o produto Sanita.

Para o dia 1 de abril de 2016, a probabilidade de se verificarem zero sanitas ( $P_0$ ) no sistema é de 39,9%, determinando-se, assim, uma taxa de ocupação ( $\rho$ ) de aproximadamente 60%. Consta-se que à medida que a taxa de ocupação é maior, o comprimento da fila de espera sem o atendimento ( $L_q$ ) é maior, bem como o tamanho da fila de espera que inclui o serviço às peças. Estas medidas de desempenho são diretamente proporcionais ao tempo de espera na fila, com e sem atendimento ( $W$  e  $W_q$  respetivamente). Neste dia verificou-se que para uma paleta de 24 sanitas, o tempo de espera sem atendimento ( $W_q$ ) é de 0,027 horas

(97,751 segundos). Este corresponde ao menor valor dos 10 dias a analisar. Verifica-se, também, que para uma sanita, o tempo de espera com atendimento ( $W$ ) é de 0,045 horas (162,596 segundos), sendo também o menor valor analisado.

Em relação ao produto tanque, tendo em consideração os dados referentes aos 10 dias previamente selecionados, obtiveram-se as respetivas medidas de desempenho.

Sabe-se que a cadência de chegada deste produto é, em média, 100 peças por hora (2185 peças por dia), e considera-se um único servidor, sabendo que se verifica um colaborador que transporta a peça ao posto de trabalho, um colaborador que está no posto de trabalho, um colaborador que retira a peça do posto de trabalho, e por fim um colaborador que se encontra na zona do forno retrátil, tal como já foi anteriormente referido.

De forma análoga ao produto sanita, a interface gráfica permitiu obter e registar os resultados, como se pode verificar no quadro seguinte. Neste quadro, estão registados os resultados das medidas de desempenho do produto tanque: a probabilidade de todos os servidores estarem desocupados ( $P_0$ ), o tempo entre chegadas ( $1/\lambda$ ), o tempo de serviço ( $1/\mu$ ) e a taxa de ocupação ( $\rho$ ).

Quadro 26 – Obtenção de cinco medidas de desempenho de fila de espera do produto tanque, para dez dias.

<b>Data</b>	<b><math>\mu</math> (peças/h)</b>	<b><math>P_0</math></b>	<b><math>1/\lambda</math>(h)</b>	<b><math>1/\mu</math> (h)</b>	<b><math>\rho</math></b>
12/01/2016	135	0,257	0,0099	0,0074	0,743
24/01/2016	123	0,184	0,0099	0,0081	0,816
07/02/2016	187	0,464	0,0099	0,0053	0,536
18/02/2016	140	0,285	0,0099	0,0071	0,715
04/03/2016	136	0,265	0,0099	0,0073	0,735
21/03/2016	122	0,180	0,0099	0,0081	0,820
30/03/2016	110	0,090	0,0099	0,009	0,910
01/04/2016	146	0,313	0,0099	0,0068	0,687
15/04/2016	130	0,230	0,0099	0,0077	0,770
27/04/2016	142	0,297	0,0099	0,0069	0,703

Relativamente ao produto tanque, no dia 7 de fevereiro de 2016, o número de peças inspecionadas e embaladas foram 187 peças por hora (corresponde a 3 peças por minuto). Verifica-se que o tempo de serviço é menor ( $1/\mu$ ), com um valor de 0,0053 horas (19 segundos). Para além disso, verifica-se que apresenta uma taxa de ocupação,  $\rho$ , com valor de 0,536, sendo o menor valor, dos 10 dias analisados. Ao invés, no dia 30 de março de 2016 verifica-se o menor valor de tanques embalados, dentro dos 10 dias selecionados, pelo que é a taxa de atendimento ( $\mu$ ) é diretamente proporcional à probabilidade de todos os servidores estarem desocupados. Posto isto, obtém-se um tempo de serviço maior ( $1/\mu$ ) com o valor de 0,009 horas (32 segundos).

Quadro 27 – Obtenção de seis medidas de desempenho de fila de espera do produto Tanque, para 10 dias.

<b>Data</b>	<b>Lq (peças)</b>	<b>L (peças)</b>	<b>Wq (h)</b>	<b>Wq (s)</b>	<b>W (h)</b>	<b>W (s)</b>
12/01/2016	2,144	2,887	0,021	76,812	0,029	103,418
24/01/2016	3,617	4,433	0,036	129,556	0,044	158,783
07/02/2016	0,620	1,156	0,006	22,199	0,012	41,404
18/02/2016	1,790	2,505	0,018	64,128	0,025	89,729
04/03/2016	2,032	2,766	0,020	72,788	0,028	99,098
21/03/2016	3,737	4,557	0,037	133,877	0,045	163,253
30/03/2016	9,158	10,067	0,091	328,037	0,100	360,622
01/04/2016	1,511	2,199	0,015	54,140	0,022	78,763
15/04/2016	2,582	3,352	0,026	92,479	0,033	120,069
27/04/2016	1,664	2,367	0,017	59,594	0,024	84,775

Para o dia 7 de fevereiro de 2016, a probabilidade de se verificarem zero tanques ( $P_0$ ) no sistema é de 46,4%, determinando-se, assim, uma taxa de ocupação ( $\rho$ ) de aproximadamente 53,6%. Verificou-se, neste dia, que para um tanque o tempo de espera sem atendimento ( $W_q$ ) é de 0,006 horas (22,199 segundos), o que corresponde ao menor valor dos 10 dias a analisar. Verifica-se, também, que o tempo de espera com atendimento ( $W$ ) é de 0,012 horas (41,404 segundos).

Para o produto lavatório, consideraram-se os dados de 10 dias previamente selecionados e obtiveram-se as respetivas medidas de desempenho.

A cadência de chegada deste produto é, em média, 67 peças por hora (1459 peças por dia), e considera-se um único servidor, sabendo que se encontra um colaborador que transporta a peça ao posto de trabalho, um colaborador que está no posto de trabalho, um colaborador que retira a peça do posto de trabalho, e por fim o forno retrátil.

No quadro seguinte estão registados os resultados das medidas de desempenho do produto lavatório: a probabilidade de todos os servidores estarem desocupados ( $P_0$ ), o tempo entre chegadas ( $1/\lambda$ ), o tempo de serviço ( $1/\mu$ ) e a taxa de ocupação ( $\rho$ ).

Quadro 28 – Obtenção de cinco medidas de desempenho de fila de espera do produto Lavatório, para 10 dias.

Data	$\mu$ (peças/h)	$P_0$	$1/\lambda$ (h)	$1/\mu$ (h)	$\rho$
12/01/2016	100	0,333	0,0149	0,0099	0,667
24/01/2016	73	0,092	0,0149	0,0135	0,908
07/02/2016	73	0,088	0,0149	0,0135	0,912
18/02/2016	83	0,193	0,0149	0,0120	0,807
04/03/2016	84	0,202	0,0149	0,0119	0,798
21/03/2016	69	0,034	0,0149	0,0143	0,966
30/03/2016	100	0,334	0,0149	0,0099	0,666
01/04/2016	133	0,498	0,0149	0,0074	0,502
15/04/2016	75	0,111	0,0149	0,0132	0,889
27/04/2016	71	0,100	0,0149	0,0139	0,900

Em relação ao produto lavatório, no dia 1 de abril de 2016, o número de peças inspeccionadas e embaladas foram de 133 peças por hora (corresponde a 2 peças por minuto). Verifica-se que o tempo de serviço é menor ( $1/\mu$ ), com um valor de 0,0074 horas (26 segundos). Para além disso, verifica-se que apresenta uma taxa de ocupação,  $\rho$ , com valor de 0,502 sendo o menor valor. Por outro lado, no dia 27 de abril verifica-se o menor valor de lavatórios embalados. Obteve-se um tempo de serviço maior ( $1/\mu$ ) com o valor de 0,0139 horas (50 segundos).

Quadro 29 – Obtenção de seis medidas de desempenho de fila de espera do produto Lavatório, para 10 dias.

Data	Lq (peças)	L (peças)	Wq (h)	Wq (s)	W (h)	W (s)
12/01/2016	1,332	1,999	0,020	71,461	0,030	107,214
24/01/2016	8,934	9,842	0,133	479,245	0,147	527,939
07/02/2016	9,454	10,366	0,141	507,136	0,154	556,059
18/02/2016	3,372	4,179	0,050	180,871	0,062	224,155
04/03/2016	3,144	3,942	0,047	168,656	0,059	211,443
21/03/2016	27,761	28,727	0,414	1489,122	0,428	1540,959
30/03/2016	1,325	1,991	0,020	71,071	0,030	106,776
01/04/2016	0,506	1,008	0,008	27,123	0,015	54,044
15/04/2016	7,096	7,985	0,106	380,636	0,119	428,308
27/04/2016	8,121	9,021	0,121	435,605	0,134	483,893

Verificou-se, no dia 1 de abril de 2016, que o tempo de espera sem atendimento ( $W_q$ ) é de 0,008 horas (27,123 segundos). Verificou-se, também, que o tempo de espera com atendimento ( $W$ ) é de 0,015 horas (54,044 segundos).

De um modo geral, constatou-se que quando a taxa de ocupação diminui,  $\rho$ ,  $P_0$  aumenta e no sentido inverso,  $W_q$  diminui, assim como o número médio de clientes na fila de espera ( $L_q$ ) e no sistema ( $L$ ).

Teria sido possível apresentar nesta monografia mais resultados obtidos através de um intervalo mais alargado, entre as datas do estágio, mas para efeitos de apresentação apenas se consideraram 10 dias, sendo possível através da aplicação desenvolvida, obter manifestamente um período mais alargado que corresponderá a necessariamente mais dados.

## 5.2 O trajeto mais curto

Relativamente à caracterização do percurso, no Quadro 30 encontram-se registados as distâncias totais do produto sanita que corresponde ao percurso de uma paleta de 24 peças.

Quadro 30 – Distância Total, para cada cenário, do produto sanita.

	<b>Cenário Atual</b>	<b>Cenário 1</b>	<b>Cenário 2</b>	<b>Trajeto mais curto</b>
<b>Distância Total (m)</b>	201	186	130	130

No Quadro 31 constata-se os dados da distância total para cada cenário, mas em função do número de peças embaladas.

Quadro 31 – Distância Total, para cada cenário, do produto Sanita, em função do número de peças embaladas.

<b>Data</b>	<b>Cenário Atual</b>	<b>Cenário 1</b>	<b>Cenário 2</b>	<b>Trajeto mais curto</b>
12-01-2016	4623	4278	2990	2990
24-01-2016	5427	5022	3510	3510
07-02-2016	5628	5208	3640	3640
18-02-2016	4221	3906	2730	2730
21-03-2016	4221	3906	2730	2730
30-03-2016	5025	4650	3250	3250
04-03-2016	5829	5394	3770	3770
01-04-2016	6633	6138	4290	4290
27-04-2016	4623	4278	2990	2990

Do tratamento de dados constatou-se que, no dia 1 de abril de 2016, foi embalado o maior número de sanitas, pelo que se traduziu num maior número de percursos realizados. No total percorreram-se 6633 metros, que correspondem à movimentação em média de 33 paletes por dia. Contudo, ao proceder-se à simulação, obteve-se o trajeto mais curto de 133 metros, que equivale ao cenário dois, anteriormente proposto. Com isto, são percorridos 4290 metros, o que equivale a uma diferença de 2343 metros. No quadro seguinte encontram-se as diferenças obtidas entre o estado atual do percurso do produto Sanita e o trajeto mais curto, obtido da simulação.

Quadro 32 – Diferença obtida entre o Estado Atual e o Trajeto mais curto, do produto sanita.

Data	12/01	24/01	07/02	18/02	21/03	30/03	04/03	01/04	27/04
$\Delta$ Percurso (m)	1633	1917	1988	1491	1491	1775	2059	2343	1633

Através da simulação, obteve-se o trajeto mais curto, como se pode visualizar na figura seguinte. Na figura encontra-se representado com contorno a vermelho o trajeto mais curto, os nós e as distâncias entre estes.

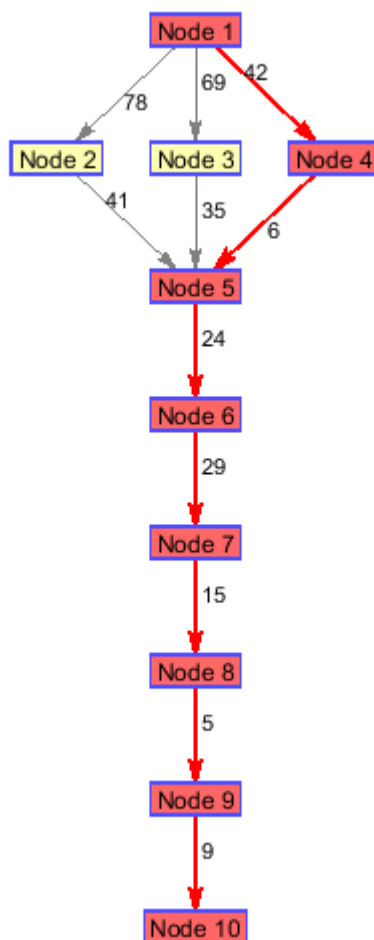


Figura 52 – Trajeto mais curto, do produto Sanita.

Em relação ao produto tanque encontram-se registados as distâncias totais que correspondem ao percurso realizado por uma paleta de 24 peças.

Quadro 33 – Distância Total, para cada cenário, do produto tanque.

	<b>Cenário Atual (m)</b>	<b>Cenário 1 (m)</b>	<b>Cenário 2 (m)</b>	<b>Trajeto mais curto (m)</b>
<b>Distância Total (m)</b>	120	113	112	104

No seguinte Quadro estão registados os dados da distância total de cada cenário, mas em função do número de peças embaladas.

Quadro 34 – Distância Total, para cada cenário, do produto tanque, em função do número de peças embaladas.

<b>Data</b>	<b>Cenário Atual (m)</b>	<b>Cenário 1 (m)</b>	<b>Cenário 2 (m)</b>	<b>Trajeto mais curto (m)</b>
12-01-2016	9720	9153	9072	8424
24-01-2016	8880	8362	8288	7696
07-02-2016	13560	12769	12656	11752
18-02-2016	10080	9492	9408	8736
21-03-2016	8880	8362	8288	7696
30-03-2016	7920	7458	7392	6864
04-03-2016	9840	9266	9184	8528
01-04-2016	10560	9944	9856	9152
27-04-2016	10320	9718	9632	8944

Dos dados registados, pode-se constatar que no dia 7 de fevereiro de 2016 foi embalado o maior número de tanques, traduzindo-se no maior número de percursos realizados, com um percurso total de 13560 metros, que correspondem à movimentação em média de 113 paletes por dia. Contudo, ao proceder-se à simulação, obtém-se, para uma paleta, o trajeto mais curto de 104 metros, que contraria os cenários um e dois como potenciais trajetos mais curtos. Com 104 metros percorridos por paleta, no dia 7 de fevereiro de 2016 são percorridos 11752 metros, o que equivale a uma diferença de 1808 metros. No quadro seguinte encontram-se as diferenças obtidas entre o estado atual do percurso do produto Tanque, e o trajeto mais curto, obtido da simulação.

Quadro 35 – Diferença obtida entre o Estado Atual e o Trajeto mais curto, do produto tanque.

Data	12/01	24/01	07/02	18/02	21/03	30/03	04/03	01/04	27/04
$\Delta$ Percurso (m)	1296	1184	1808	1344	1184	1056	1312	1408	1376

Através da simulação, obteve-se o trajeto mais curto, como se pode visualizar na figura seguinte, salientando com contorno a vermelho o trajeto mais curto, os nós e as distâncias entre estes.

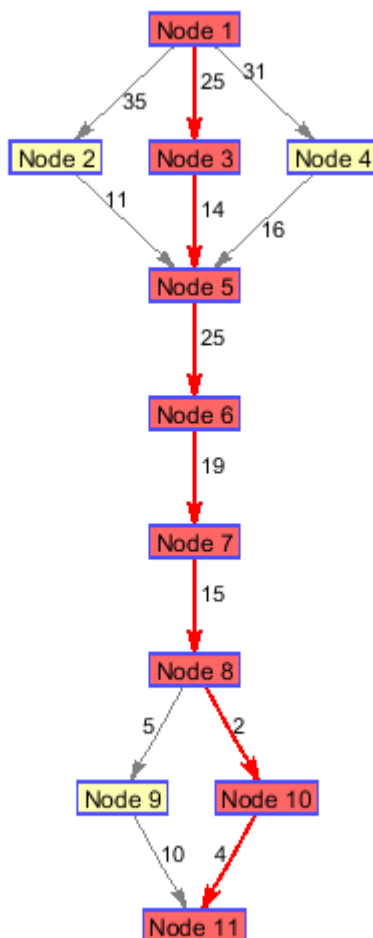


Figura 53 – Trajeto mais curto, do produto tanque.

Por fim, relativamente ao lavatório encontram-se registados as distâncias totais para este, e que correspondem ao percurso realizado por uma paleta de 24 peças.

No Quadro 36, encontram-se estas distâncias totais, para o produto lavatório:

Quadro 36 – Distância Total, para cada cenário, do produto lavatório.

	<b>Cenário Atual</b>	<b>Cenário 1</b>	<b>Cenário 2</b>	<b>Trajeto mais curto</b>
<b>Distância Total (m)</b>	176	170	167	165

No Quadro 37, encontram-se registados os dados da distância total cada cenário, mas em função do número de peças embaladas.

Quadro 37 – Distância Total, para cada cenário, do produto lavatório, em função do número de peças embaladas.

<b>Data</b>	<b>Cenário Atual</b>	<b>Cenário 1</b>	<b>Cenário 2</b>	<b>Trajeto mais curto</b>
12-01-2016	10560	10200	10020	9900
24-01-2016	7744	7480	7348	7260
07-02-2016	7744	7480	7348	7260
18-02-2016	8800	8500	8350	8250
21-03-2016	7216	6970	6847	6765
30-03-2016	10560	10200	10020	9900
04-03-2016	8800	8500	8350	8250
01-04-2016	14080	13600	13360	13200
27-04-2016	7568	7310	7181	7095

Do tratamento de dados constatou-se que, no dia 1 de abril de 2016, foi embalado o maior número de lavatórios, com percursos no total de 14080 metros, que correspondem à movimentação em média de 80 paletes por dia. Ao executar-se a simulação obteve-se, para uma paleta, o trajeto mais curto de 165 metros, que contraria os cenários um e dois como potenciais trajetos mais curtos. Com este novo trajeto, no dia 1 de abril de 2016 foram percorridos 13200 metros, o que equivale a uma diferença de 880 metros. No quadro seguinte encontram-se as diferenças obtidas entre o estado atual do percurso do produto lavatório, e o trajeto mais curto, obtido da simulação.

Quadro 38 – Diferença obtida entre o Estado Atual e o Trajeto mais curto, do produto lavatório.

Data	12/01	24/01	07/02	18/02	21/03	30/03	04/03	01/04	27/04
$\Delta$ Percurso (m)	660	484	484	550	451	660	550	880	473

Através da simulação, obteve-se o trajeto mais curto, como se pode visualizar na figura seguinte, salientando com contorno a vermelho o trajeto mais curto, os nós e as distâncias entre estes.

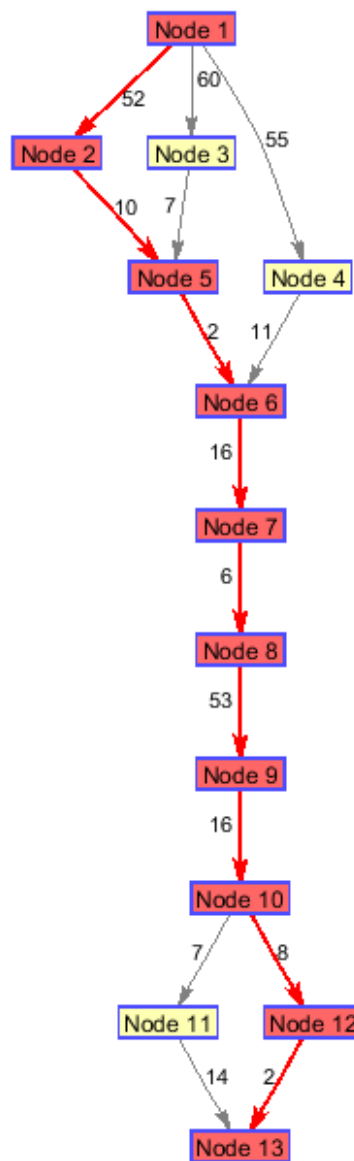


Figura 54 – Trajeto mais curto, do produto tanque.



## 6. Conclusão

Atualmente, a simulação computacional em contextos industriais conduz a estudos e análises mais rigorosas e à obtenção de resultados cada vez mais eficazes. Este tipo de análise dinâmica visa o dimensionamento e a otimização de processos, sendo de extrema importância a sua utilização face às alterações da procura no mercado. A constante mudança do mercado tem consequências no modo de funcionamento de uma unidade fabril e no planeamento e controlo da sua capacidade. Face a estas exigências de alteração rápida e ágil da configuração do processo produtivo, a simulação computacional tem um papel de destaque indelével na gestão da produção e subseqüentemente na otimização da mesma.

A realização desta monografia conduziu à oportunidade de analisar a simulação computacional, no meio industrial, com o intuito de avaliar e melhorar o funcionamento das atividades e tarefas. Através do desenvolvimento de um *software* em Matlab, foi possível realizar um estudo no que diz respeito a duas importantes vertentes: filas de espera e trajetos mais curtos.

De forma a melhor fundamentar a análise, foi efetuado um estudo prévio de modo a caracterizar a fila de espera. No tocante aos trajetos mais curtos para os três tipos de peças analisados: o tanque, o lavatório e a sanita foram analisados vários cenários e cujos resultados já foram apresentados no capítulo anterior. Em ambos os casos, o estudo simultâneo, das duas vertentes, foi realizado na secção de inspeção e embalagem final, designada Escolha Final, da fábrica de Leiria 1.

No estudo e caracterização de filas de espera, a dimensão da população é o número total de peças que têm cadência de chegada na secção com o intuito de serem embaladas. A dimensão caracteriza-se por ser infinita, uma vez que o número de peças na secção é uma percentagem reduzida relativamente às restantes peças da cadeia produtiva. Por outro lado, após observações, concluiu-se que a dimensão da fila de espera é infinita e a disciplina desta respeita o critério aleatório.

De acordo com a Teoria de Filas de Espera, foi aplicado o modelo marcoviano M/M/S, com um servidor em que as distribuições de tempos entre chegadas e de tempos de serviços seguem uma distribuição de Poisson.

Ao efetuar-se o estudo de filas de espera, determinou-se a cadência de chegada para os três produtos, onde o lavatório apresentou um valor médio de cadência de chegada de 1459 peças por dia, o tanque apresentou um valor médio de 2185 peças por dia e a sanita apresentou um valor médio de 725 peças por dia. Constatou-se que o tanque é o produto com um maior índice de cadência de chegada. O número de peças que são embaladas é diferente entre os vários tipos de produtos, sendo que o tanque é o produto com um maior número de peças embaladas. Deste modo, este produto apresenta fluxos logísticos com maior complexidade,

traduzindo-se em valores elevados no percurso percorrido ao fim de um dia, ao que não será estranho o facto de o tanque requerer a embalagem de peças adicionais (mecanismos, parafusos de fixação e acessórios). O facto de o tanque ser o produto com maior cadência de chegada está relacionado com uma linha que dispõe de maior capacidade de mão-de-obra.

Em relação ao estudo de filas de espera para a sanita, no dia 1 de abril de 2016, a probabilidade de se verificarem zero sanitas ( $P_0$ ) no sistema foi de 39,9%, determinando-se, assim, uma taxa de ocupação ( $\rho$ ) de aproximadamente 60%. Constatou-se que à medida que a taxa de ocupação é maior, o comprimento da fila de espera sem atendimento ( $L_q$ ) é maior, bem como o tamanho da fila de espera que inclui o serviço às peças de cerâmica. Estas medidas de desempenho são diretamente proporcionais ao tempo de espera na fila com e sem atendimento ( $W$  e  $W_q$  respetivamente). O tempo de espera sem atendimento ( $W_q$ ) foi de 0,027 horas (97,751 segundos) que corresponde ao menor valor num conjunto de dados obtidos num período de dez dias. Verificou-se que para uma sanita o tempo de espera com atendimento ( $W$ ) é de 0,045 horas (162,596 segundos).

Da análise realizada para o tanque, no dia 7 de fevereiro de 2016, a probabilidade de se verificarem zero tanques ( $P_0$ ) no sistema foi de 46,4%, com uma taxa de ocupação ( $\rho$ ) de 53,6%. Verificou-se, neste dia, que para um tanque o tempo de espera sem atendimento ( $W_q$ ) foi de 0,006 horas (22,199 segundos), o que corresponde ao menor valor dos dez dias a analisar. Verificou-se, também, que o tempo de espera com atendimento ( $W$ ) é de 0,012 horas (41,404 segundos).

Relativamente ao lavatório, no dia 1 de abril de 2016, o tempo de espera sem atendimento ( $W_q$ ) foi de 0,008 horas (27,123 segundos). Verificou-se que o tempo de espera com atendimento ( $W$ ) foi de 0,015 horas (54,044 segundos). De um modo geral, constatou-se que quando a taxa de ocupação diminui,  $\rho$ ,  $P_0$  aumenta, e  $W_q$  diminui, assim como o número médio de clientes na fila de espera ( $L_q$ ) e no sistema ( $L$ ).

Na análise dos percursos de cada produto, através da observação do estado atual e de dois cenários propostos, procedeu-se à determinação do trajeto mais curto. Verificou-se que a sanita revela um trajeto mais curto de 130 metros, face ao estado atual. Este trajeto obtido traduz o cenário dois. O trajeto mais curto, para um tanque, teve o valor de 104 metros obtendo-se assim um novo trajeto diferente dos cenários. Por outro lado, para um lavatório o valor do trajeto mais curto foi de 165 metros.

Deste modo, para a sanita, no dia 1 de abril 2016 verificou-se um maior número de peças embaladas (805 peças no total), pelo que a soma total de percursos para o trajeto mais curto foi de 4290 metros. Constatou-se uma diferença de 2343 metros face ao cenário atual. Assim, para o tanque no dia 7 de fevereiro de 2016 obteve-se a soma total do trajeto mais curto de 11752 metros para 2718 peças, com uma diferença de 1808 metros, e para o lavatório, no dia 1 de abril de 2016, verificou-se um trajeto ótimo de 13200 metros para 1939 peças, com uma diferença de 880 metros, em comparação ao cenário atual.

A utilização da aplicação desenvolvida para a presente monografia foi uma mais valia, no sentido de facilitar a simulação do processo logístico para cada tipo de produto. A aplicação integrou, assim, duas partes, a caracterização da fila de espera e a caracterização do percurso.



## Bibliografia

- Anu Maria. (1997). *Introduction to modeling and simulation*. State University of New York at Binghamton, Department of Systems Science and Industrial Engineering, Binghamton, NY 13902-6000, U.S.A.
- Banks, J. (Ed.). (1998). *Handbook of Simulation - Principles, Methodology, Advances, Applications, and Practice*. Atlanta, Georgia: Georgis Institute of Technology.
- Barrico, C. (2016). Programação III - Capítulo 5 - Grafos e redes. Covilhã: Universidade da Beira Interior. Retrieved from ubi.
- Barrico, C. (2016). Programação III - Capítulo 5 - Grafos e redes. Covilhã: Universidade da Beira Interior.
- Beaverstock, M., Greenwood, A., Lavery, E., & Nordgren, W. (n.d.). *Applied Simulation - Modeling and Analysis using Flexsim visualizing and optimizing dynamic systems*.
- Chaturvedi, D. K. (2010). *Modeling and Simulation of Systems using Matlab and Simulink*. CRC Press.
- Dantzig, G. B., & Thapa, M. N. (1997). *Linear Programming - I: Introduction*. Springer.
- Goldberg, M. C., & Luna, H. P. (2000). *Otimização Combinatória e Programação Linear - Modelos e Algoritmos*. Campus Ltda.
- Hillier, F. S., & Lieberman, G. J. (1990). *"Introduction to Operations Research"* (5th ed. ed.). (E. M. Munson, & S. Amerman, Eds.) MCGRAW-HILL INTERNATIONAL EDITIONS.
- Law, A. (1986). *"Introduction to simulation: a powerful tool for analyzing complex manufacturing systems"*.
- Law, A., & Kelton, W. (2000). *"Simulation Modeling and Analysis"* (2nd ed. ed.). New York: McGraw-Hill, Inc.
- Marchand, P., & Holland, O. (2003). *Graphics and Guis with Matlab - Third Edition*. Chapman & Hall/CRC.
- Martins, P., Pascoal, M., Santos, J., & Joaquim J. (2006). *Otimização em redes*. Universidade de Coimbra, Departamento de Matemática.
- Miglani, V., Khera, S., & Aggarawal, A. (2016). *Intrusion Protection System in Dijkstra's Algorithm*. YMCA University of Science Technology, Department of Electronics Engineering, Faridabad, Haryana, India.
- Mohd Sabri, N., Hasan Basari, A., Husin, B., & Abu Samah, K. (2014). *The Utilisation of Dijkstra's Algorithm to assist evacuation route in higher and close building*. Center of Advanced Computing Technology, Faculty of Information and Communication Technology. Malaysia: Original Research Paper.

Preston White, J. K., & Ingalls, R. (2015). *"Introduction to Simulation"*. University of Virginia; Oklahoma State University, Department of Systems and Information Engineering; School of Industrial Engineering and Management, Charlottesville, VA 22904-4747, U.S.A; Stillwater, OK 74078, U.S.A.

*Promodel*. (2016, Janeiro 8). Retrieved Janeiro 8, 2016, from [www.promodel.com](http://www.promodel.com)

Reeb, J., & Leavengood, S. (2003). *"Simulating a Manufacturing System: An Introduction"*.

Seleima, A., Azaba, A., & Algeddwy, T. (2012). *Simulation Methods for Changeable Manufacturing*.

Tavares, L. V., Oliveira, R. C., Themido, I. H., & Correia, F. N. (1996). *Investigação Operacional*. Mc-Graw-Hill de Portugal, Lda.

Williams, E., & Narayanaswamy, R. (1997). *"Application of Simulation to Scheduling, Sequencing, and Material Handling"*. (K. H. Andradóttir, D. Withers, & B. Nelson, Eds.) Piscataway, New Jersey: IEEE, Inc.

(2016, Janeiro 10). Retrieved from Arena Simulation: [www.arenasimulation.com/industry-solutions/manufacturing-simulation-software](http://www.arenasimulation.com/industry-solutions/manufacturing-simulation-software)

(2016, Fevereiro 10). Retrieved from Roca: <http://www.pt.roca.com/>

## **ANEXOS**



## **Anexo 1**

**Dados registados de peças embaladas em janeiro, fevereiro,  
março e abril de 2016**

Data	Quantidade peças embaladas	% peças embaladas	Data	Quantidade peças embaladas	% peças embaladas
01/01	0	0%	17/01	6017	89%
02/01	0	0%	18/01	6724	90%
03/01	0	0%	19/01	6578	91%
04/01	1137	83%	20/01	6185	90%
05/01	521	84%	21/01	6545	90%
06/01	3895	83%	22/01	7020	90%
07/01	6123	89%	23/01	6310	92%
08/01	5510	89%	24/01	6091	91%
09/01	5760	88%	25/01	6499	88%
10/01	6448	89%	26/01	6293	89%
11/01	5765	88%	27/01	6190	89%
12/01	6567	89%	28/01	7001	90%
13/01	6601	88%	29/01	5666	87%
14/01	6595	90%	30/01	7159	90%
15/01	6638	90%	31/01	5756	91%
16/01	6775	90%		6017	89%

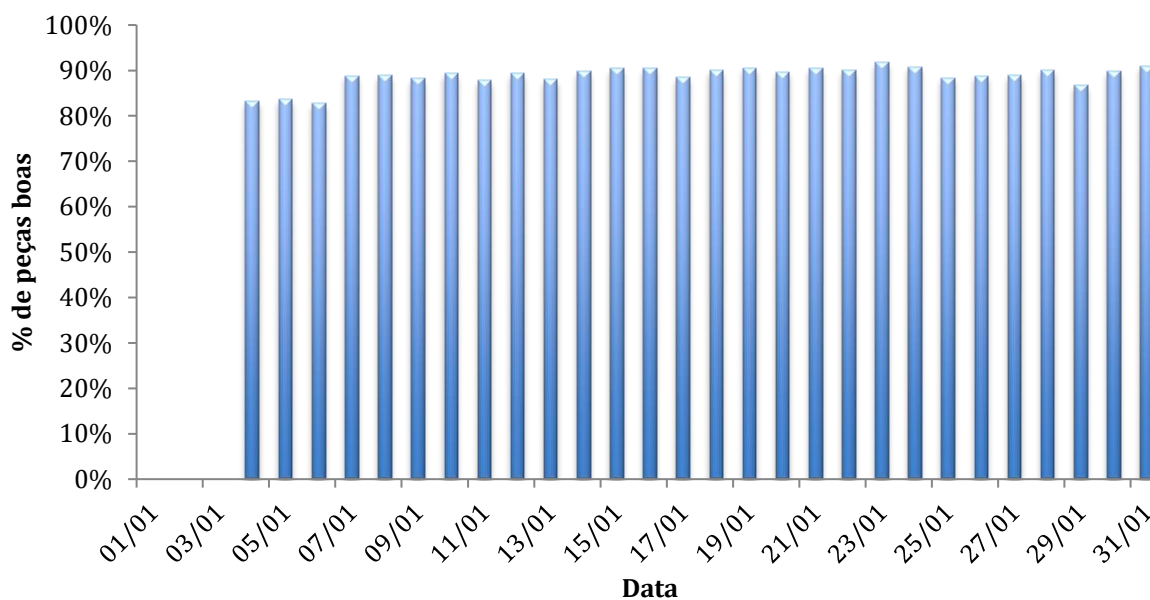


Figura 55 – Percentagem de peças boas embaladas durante o mês de janeiro.

Data	Quantidade peças embaladas	% peças embaladas	Data	Quantidade peças embaladas	% peças embaladas
01/02	7527	90%	17/02	6153	90%
02/02	6660	89%	18/02	6395	89%
03/02	5974	88%	19/02	6807	89%
04/02	7011	90%	20/02	6573	91%
05/02	7176	90%	21/02	6704	91%
06/02	5843	88%	22/02	7466	90%
07/02	7692	90%	23/02	6407	89%
08/02	6485	90%	24/02	6496	88%
09/02	6509	91%	25/02	6847	90%
10/02	6973	90%	26/02	7181	89%
11/02	7140	88%	27/02	6760	88%
12/02	6460	90%	28/02	6511	89%
13/02	5990	89%	29/02	6258	90%
14/02	6048	88%			
15/02	6621	88%			
16/02	6879	88%			

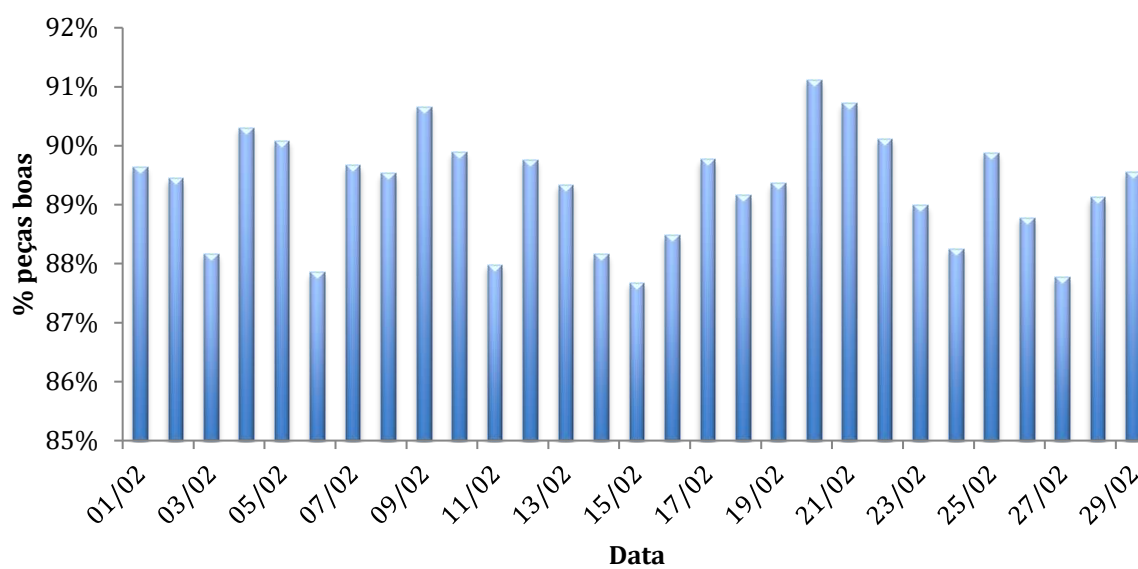


Figura 56 – Percentagem de peças boas embaladas durante o mês de fevereiro.

Data	Quantidade peças embaladas	% peças embaladas	Data	Quantidade peças embaladas	% peças embaladas
01/03	7761	88%	17/03	6547	90%
02/03	7450	90%	18/03	6714	90%
03/03	6884	89%	19/03	6583	90%
04/03	6778	90%	20/03	6681	91%
05/03	6278	87%	21/03	6928	89%
06/03	6493	88%	22/03	6879	88%
07/03	8330	89%	23/03	6179	89%
08/03	6602	89%	24/03	6771	91%
09/03	6846	87%	25/03	6797	89%
10/03	6795	87%	26/03	6931	90%
11/03	6558	89%	27/03	5529	89%
12/03	6528	91%	28/03	6801	89%
13/03	6615	88%	29/03	7248	88%
14/03	6891	89%	30/03	6299	86%
15/03	7788	89%	31/03	5994	90%
16/03	7251	89%			

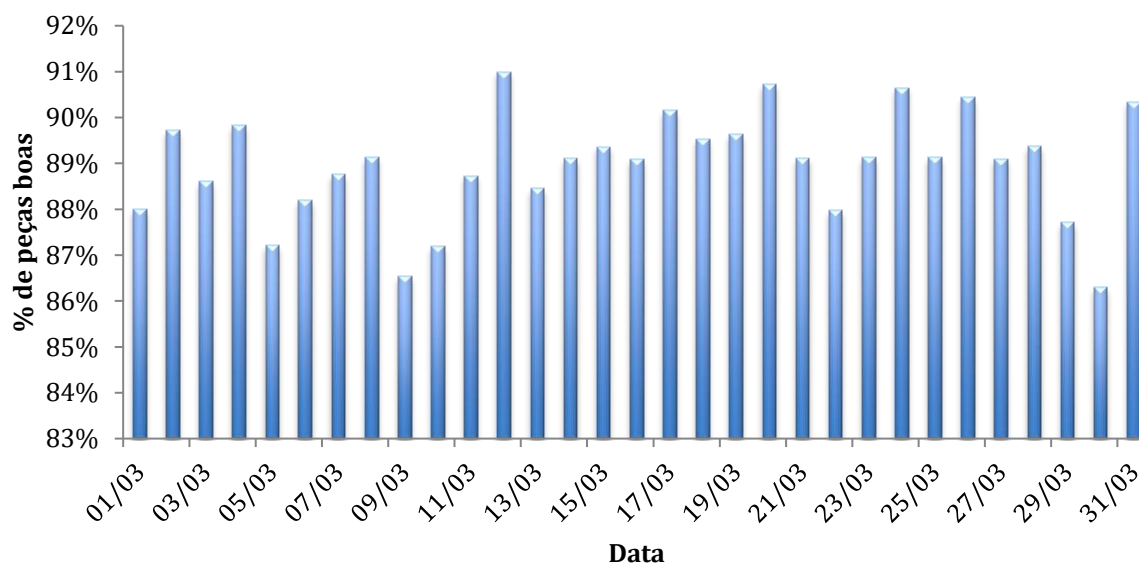


Figura 57 – Percentagem de peças boas embaladas durante o mês de março.

Data	Quantidade peças embaladas	% peças embaladas	Data	Quantidade peças embaladas	% peças embaladas
01/04	8141	90%	17/04	7048	88%
02/04	6702	87%	18/04	7119	89%
03/04	7005	89%	19/04	6884	89%
04/04	6992	89%	20/04	7096	89%
05/04	6523	90%	21/04	7550	90%
06/04	6471	88%	22/04	6875	89%
07/04	6957	88%	23/04	7089	88%
08/04	6259	88%	24/04	6983	91%
09/04	6819	90%	25/04	6880	89%
10/04	6714	89%	26/04	6948	88%
11/04	7259	89%	27/04	6998	87%
12/04	7522	89%	28/04	6458	87%
13/04	6466	89%	29/04	6382	89%
14/04	6870	90%	30/04	396	81%
15/04	6880	88%			
16/04	6550	89%			

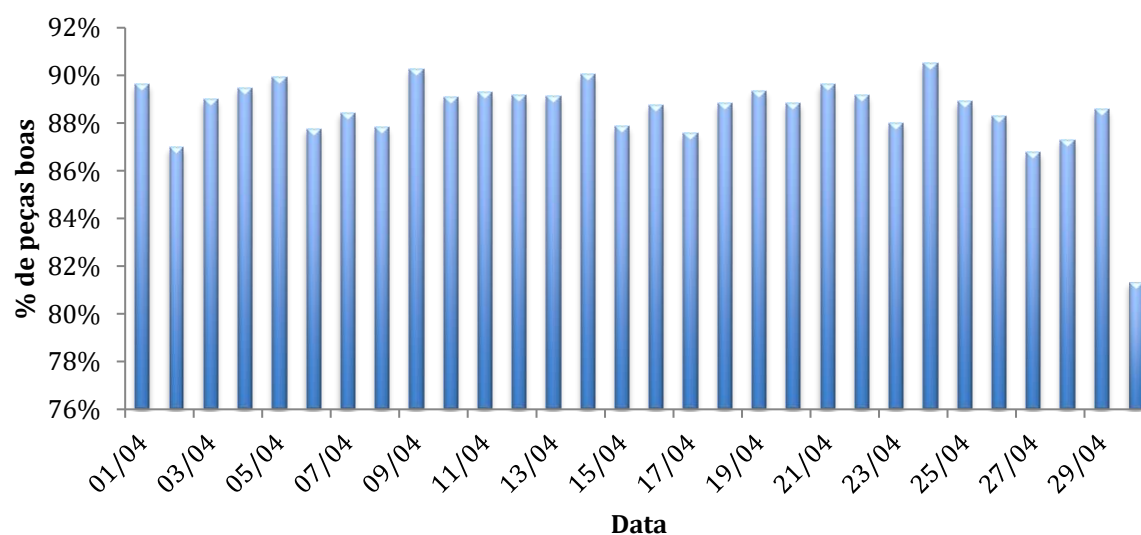


Figura 58 – Percentagem de peças boas embaladas durante o mês de abril.

Tipo de Produto: sanita.

Data	Quantidade peças embaladas	% peças embaladas	Data	Quantidade peças embaladas	% peças embaladas
01/01	0	0%	17/01	798	12%
02/01	0	0%	18/01	670	9%
03/01	0	0%	19/01	682	9%
04/01	143	10%	20/01	837	12%
05/01	296	48%	21/01	683	9%
06/01	448	10%	22/01	510	7%
07/01	605	9%	23/01	633	9%
08/01	646	10%	24/01	670	10%
09/01	566	9%	25/01	583	8%
10/01	455	6%	26/01	689	10%
11/01	805	12%	27/01	504	7%
12/01	571	8%	28/01	687	9%
13/01	573	8%	29/01	831	13%
14/01	659	9%	30/01	508	6%
15/01	655	9%	31/01	456	7%
16/01	443	6%			

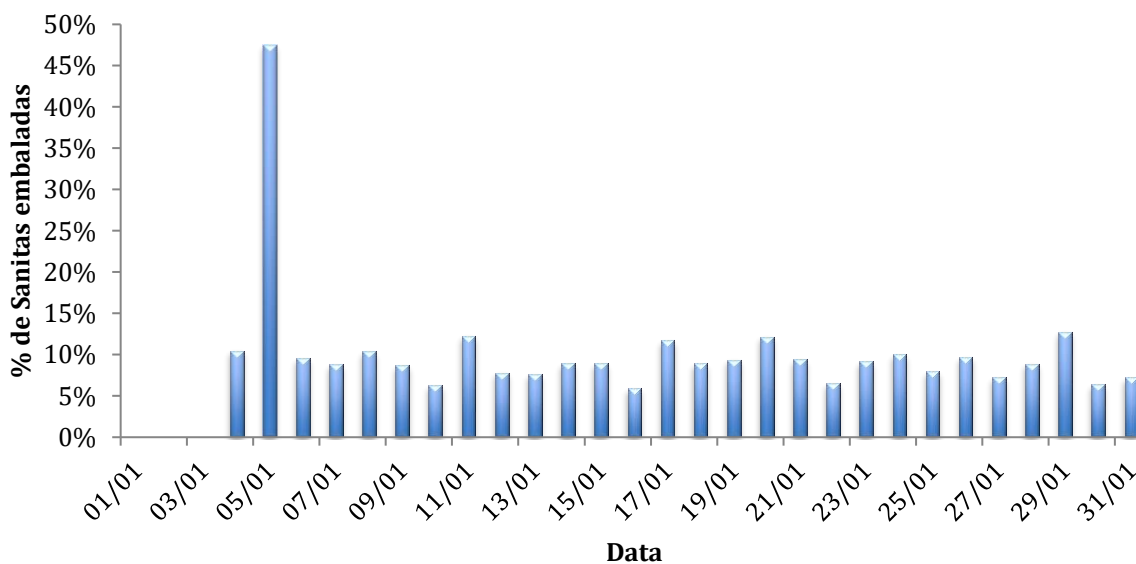


Figura 59 – Percentagem do tipo de produto sanita, embalado durante o mês de janeiro.

Data	Quantidade peças embaladas	% peças embaladas	Data	Quantidade peças embaladas	% peças embaladas
01/02	682	8%	17/02	675	10%
02/02	619	8%	18/02	515	7%
03/02	646	10%	19/02	690	9%
04/02	570	7%	20/02	611	8%
05/02	508	6%	21/02	719	10%
06/02	748	11%	22/02	525	6%
07/02	681	8%	23/02	671	9%
08/02	515	7%	24/02	582	8%
09/02	585	8%	25/02	542	7%
10/02	672	9%	26/02	670	8%
11/02	509	6%	27/02	749	10%
12/02	642	9%	28/02	628	9%
13/02	562	8%	29/02	554	8%
14/02	723	11%			
15/02	445	6%			
16/02	497	6%			

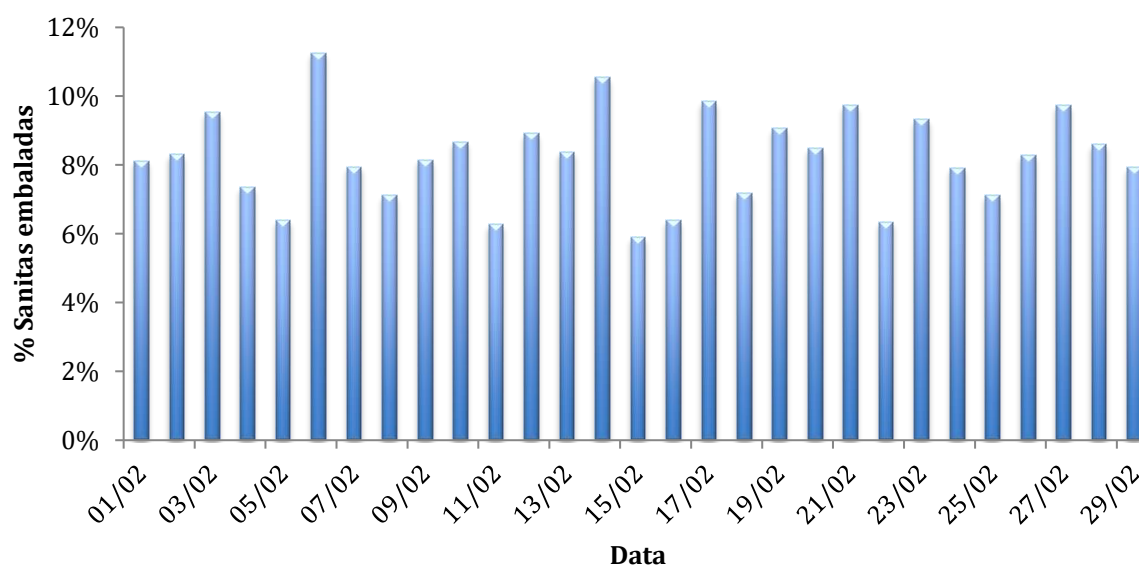


Figura 60 – Percentagem do tipo de produto sanita, embalado durante o mês de fevereiro.

Data	Quantidade peças embaladas	% peças embaladas	Data	Quantidade peças embaladas	% peças embaladas
01/03	763	9%	17/03	529	7%
02/03	427	5%	18/03	792	11%
03/03	786	10%	19/03	584	8%
04/03	696	9%	20/03	657	9%
05/03	618	9%	21/03	518	7%
06/03	568	8%	22/03	657	8%
07/03	487	5%	23/03	607	9%
08/03	538	7%	24/03	500	7%
09/03	981	12%	25/03	593	8%
10/03	631	8%	26/03	447	6%
11/03	706	10%	27/03	574	9%
12/03	842	12%	28/03	733	10%
13/03	643	9%	29/03	536	6%
14/03	924	12%	30/03	605	8%
15/03	824	9%	31/03	549	8%
16/03	500	6%			

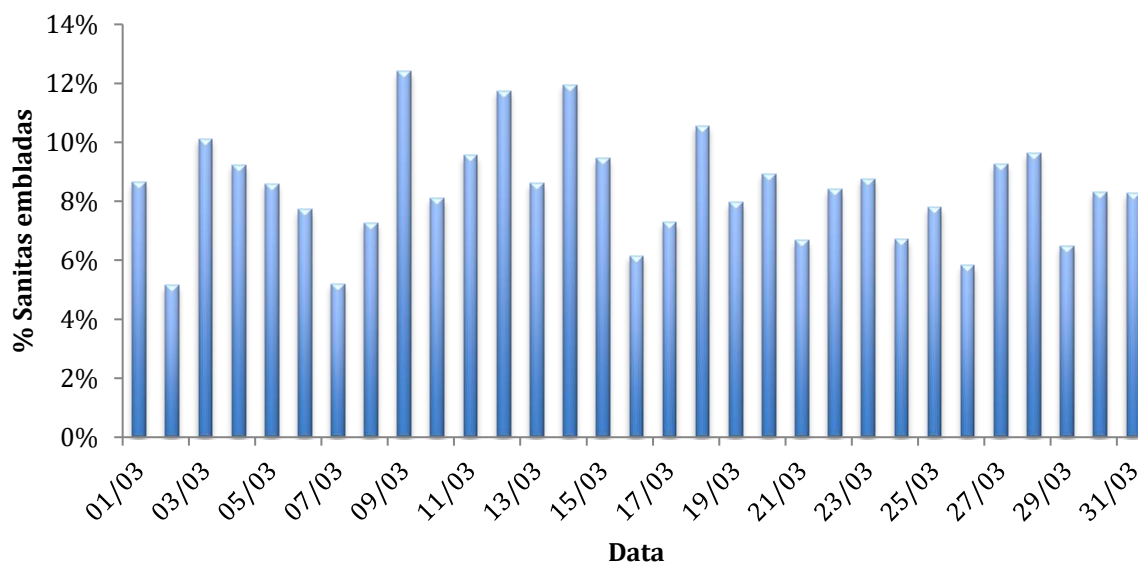


Figura 61 – Percentagem do tipo de produto sanita, embalado durante o mês de março.

Data	Quantidade peças embaladas	% peças embaladas	Data	Quantidade peças embaladas	% peças embaladas
01/04	805	9%	17/04	618	8%
02/04	564	7%	18/04	586	7%
03/04	437	6%	19/04	641	8%
04/04	739	9%	20/04	690	9%
05/04	609	8%	21/04	687	8%
06/04	508	7%	22/04	617	8%
07/04	549	7%	23/04	778	10%
08/04	670	9%	24/04	540	7%
09/04	705	9%	25/04	800	10%
10/04	488	6%	26/04	522	7%
11/04	654	8%	27/04	557	7%
12/04	438	5%	28/04	562	8%
13/04	760	10%	29/04	703	10%
14/04	571	7%	30/04	81	17%
15/04	661	8%			
16/04	816	11%			

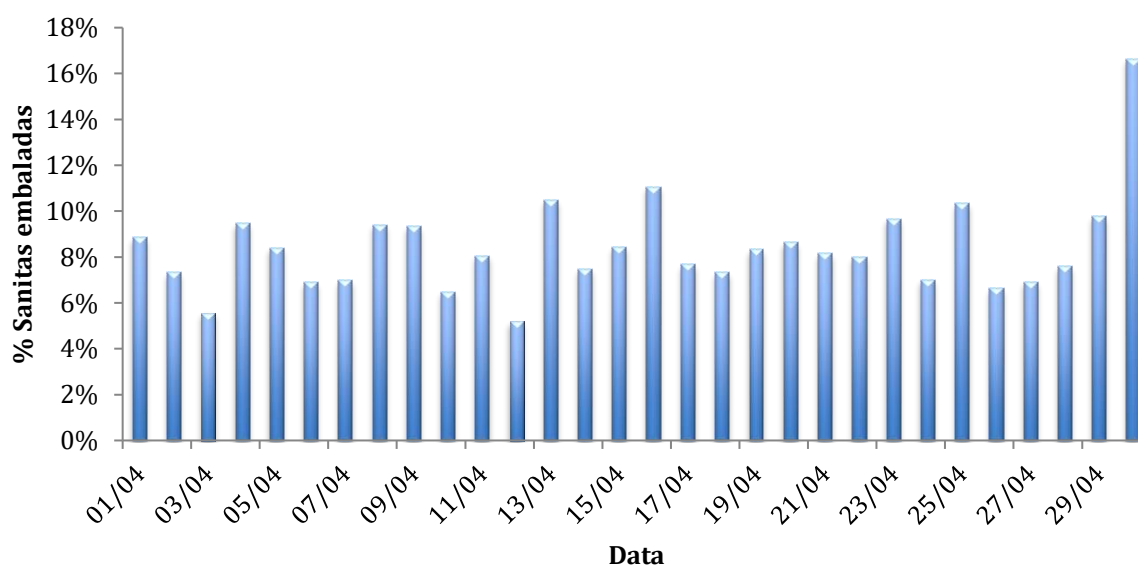


Figura 62 – Percentagem do tipo de produto sanita, embalado durante o mês de abril.

Tipo de Produto: lavatório.

Data	Quantidade peças embaladas	% peças embaladas	Data	Quantidade peças embaladas	% peças embaladas
01/01	0	0%	17/01	966	14%
02/01	0	0%	18/01	1660	22%
03/01	0	0%	19/01	1057	15%
04/01	464	34%	20/01	986	14%
05/01	228	37%	21/01	1491	21%
06/01	782	17%	22/01	977	13%
07/01	742	11%	23/01	961	14%
08/01	653	11%	24/01	1072	16%
09/01	885	14%	25/01	1412	19%
10/01	1370	19%	26/01	1125	16%
11/01	969	15%	27/01	1424	20%
12/01	1460	20%	28/01	1181	15%
13/01	1231	16%	29/01	1037	16%
14/01	990	13%	30/01	1190	15%
15/01	939	13%	31/01	924	15%
16/01	1353	18%			

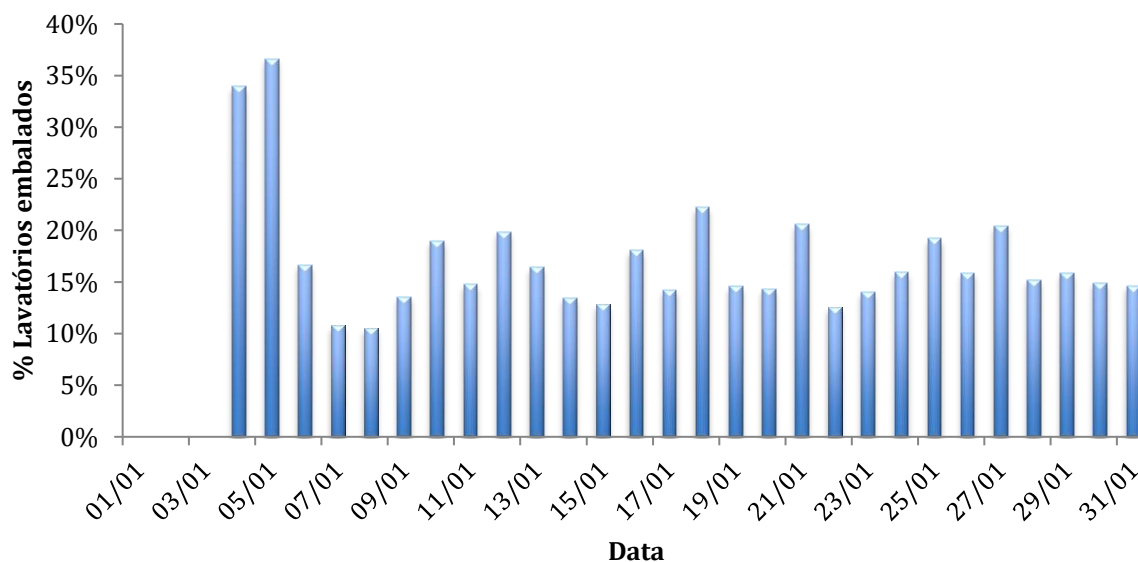


Figura 63 – Percentagem do tipo de produto lavatório, embalado durante o mês de janeiro.

Data	Quantidade peças embaladas	% peças embaladas	Data	Quantidade peças embaladas	% peças embaladas
01/02	1708	20%	17/02	1016	15%
02/02	1166	16%	18/02	1206	17%
03/02	1577	23%	19/02	919	12%
04/02	1185	15%	20/02	1402	19%
05/02	1289	16%	21/02	1308	18%
06/02	1440	22%	22/02	1249	15%
07/02	1067	12%	23/02	1453	20%
08/02	1362	19%	24/02	1086	15%
09/02	1290	18%	25/02	1186	16%
10/02	1178	15%	26/02	1363	17%
11/02	1434	18%	27/02	815	11%
12/02	1281	18%	28/02	1664	23%
13/02	1392	21%	29/02	879	13%
14/02	1148	17%			
15/02	1166	15%			
16/02	997	13%			

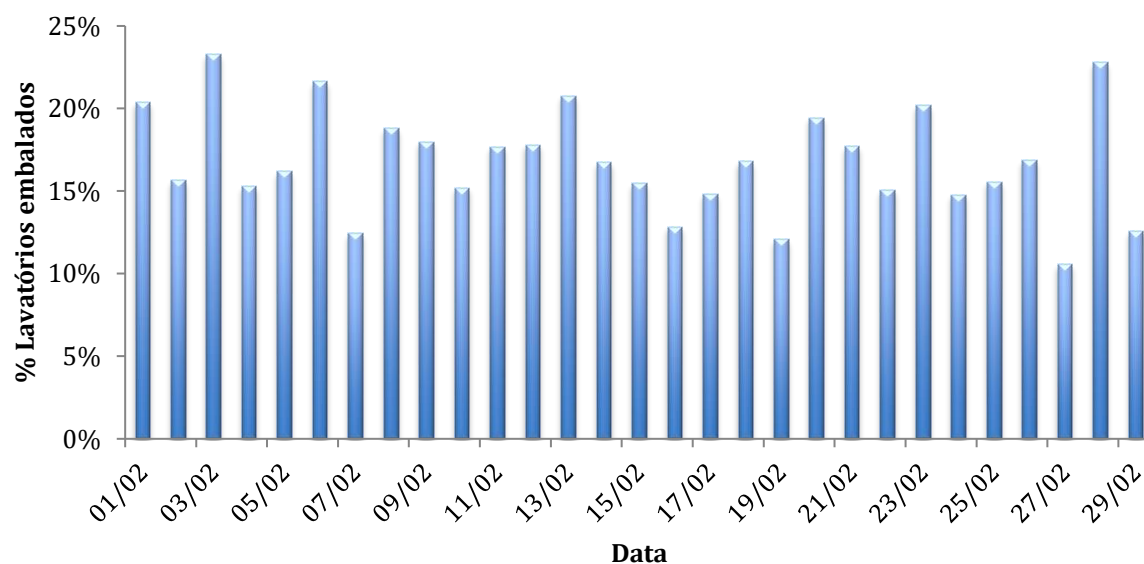


Figura 64 – Percentagem do tipo de produto lavatório, embalado durante o mês de fevereiro.

Data	Quantidade peças embaladas	% peças embaladas	Data	Quantidade peças embaladas	% peças embaladas
01/03	1441	16%	17/03	1200	17%
02/03	1125	14%	18/03	907	12%
03/03	1217	16%	19/03	989	13%
04/03	1220	16%	20/03	1307	18%
05/03	1007	14%	21/03	1007	13%
06/03	1343	18%	22/03	1182	15%
07/03	1112	12%	23/03	1112	16%
08/03	1221	16%	24/03	1108	15%
09/03	1200	15%	25/03	1181	15%
10/03	992	13%	26/03	1131	15%
11/03	1054	14%	27/03	747	12%
12/03	1226	17%	28/03	1394	18%
13/03	1017	14%	29/03	1412	17%
14/03	1130	15%	30/03	1462	20%
15/03	1153	13%	31/03	1011	15%
16/03	1027	13%			

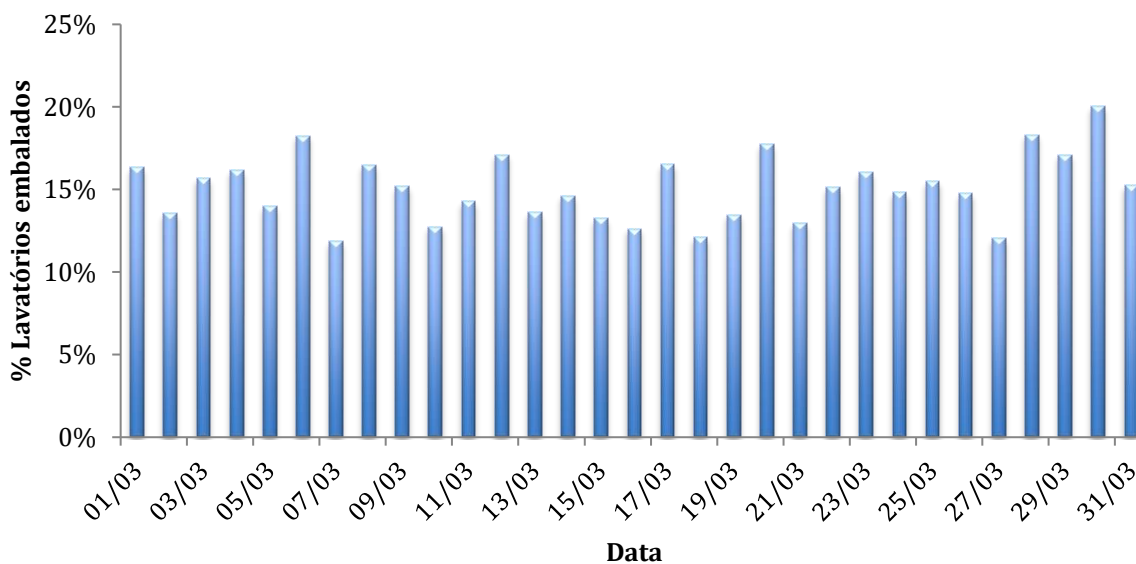


Figura 65 – Percentagem do tipo de produto lavatório, embalado durante o mês de março.

Data	Quantidade peças embaladas	% peças embaladas	Data	Quantidade peças embaladas	% peças embaladas
01/04	1939	21%	17/04	1298	16%
02/04	1152	15%	18/04	1151	14%
03/04	1153	15%	19/04	1216	16%
04/04	1095	14%	20/04	999	13%
05/04	1167	16%	21/04	1355	16%
06/04	1045	14%	22/04	1111	14%
07/04	1145	15%	23/04	1105	14%
08/04	1081	15%	24/04	1081	14%
09/04	1760	23%	25/04	1203	16%
10/04	1111	15%	26/04	1148	15%
11/04	1098	14%	27/04	1039	13%
12/04	1118	13%	28/04	1160	16%
13/04	1099	15%	29/04	992	14%
14/04	963	13%	30/04	62	13%
15/04	1095	14%			
16/04	1086	15%			

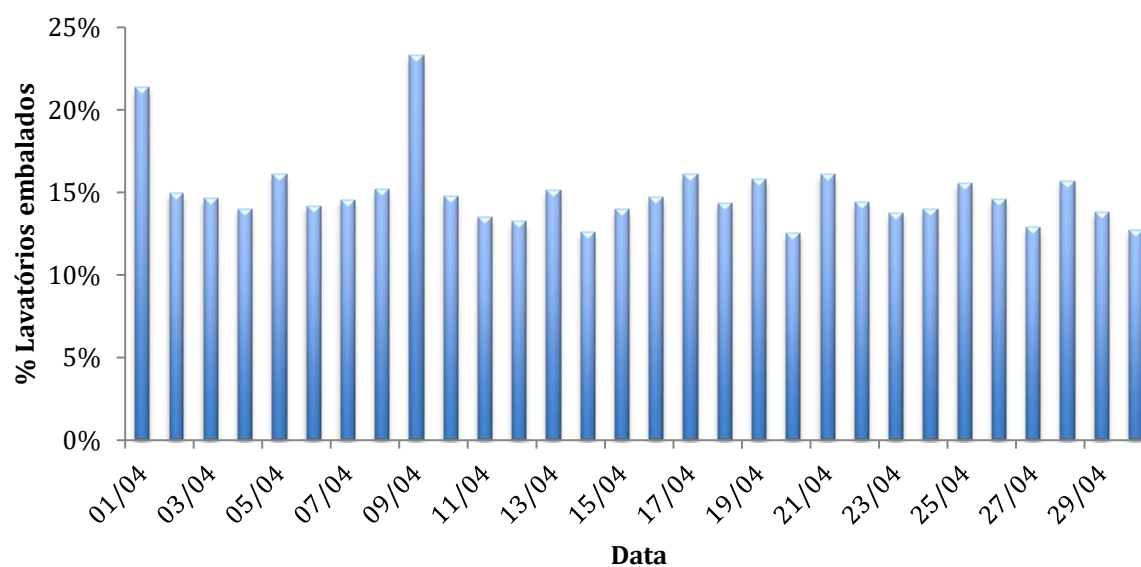


Figura 66 – Percentagem do tipo de produto lavatório, embalado durante o mês de abril.

Tipo de Produto: tanque.

Data	Quantidade peças embaladas	% peças embaladas	Data	Quantidade peças embaladas	% peças embaladas
01/01	0	0%	17/01	1719	25%
02/01	0	0%	18/01	1532	21%
03/01	0	0%	19/01	2032	28%
04/01	0	0%	20/01	1892	27%
05/01	14	2%	21/01	1931	27%
06/01	1220	26%	22/01	2468	32%
07/01	2170	31%	23/01	2036	30%
08/01	1908	31%	24/01	1786	27%
09/01	1985	30%	25/01	1941	26%
10/01	2160	30%	26/01	1722	24%
11/01	1826	28%	27/01	1800	26%
12/01	1962	27%	28/01	2107	27%
13/01	2059	27%	29/01	1740	27%
14/01	2210	30%	30/01	2229	28%
15/01	2168	30%	31/01	1276	20%
16/01	1907	25%			

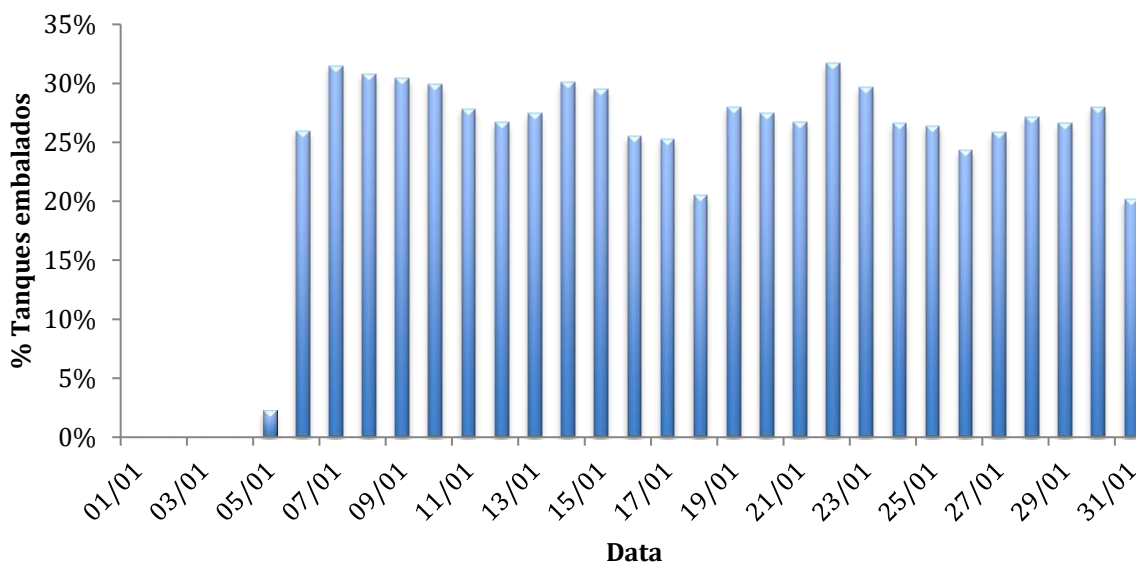


Figura 67 – Percentagem do tipo de produto tanque, embalado durante o mês de janeiro.

Data	Quantidade peças embaladas	% peças embaladas	Data	Quantidade peças embaladas	% peças embaladas
01/02	2159	26%	17/02	1903	28%
02/02	2080	28%	18/02	2039	28%
03/02	1369	20%	19/02	2452	32%
04/02	2331	30%	20/02	2142	30%
05/02	2216	28%	21/02	2025	27%
06/02	1319	20%	22/02	2426	29%
07/02	2718	32%	23/02	1921	27%
08/02	1871	26%	24/02	2083	28%
09/02	1903	27%	25/02	2181	29%
10/02	2121	27%	26/02	2309	29%
11/02	2428	30%	27/02	2222	29%
12/02	1830	25%	28/02	2027	28%
13/02	1641	24%	29/02	2051	29%
14/02	1772	26%			
15/02	2129	28%			
16/02	2401	31%			

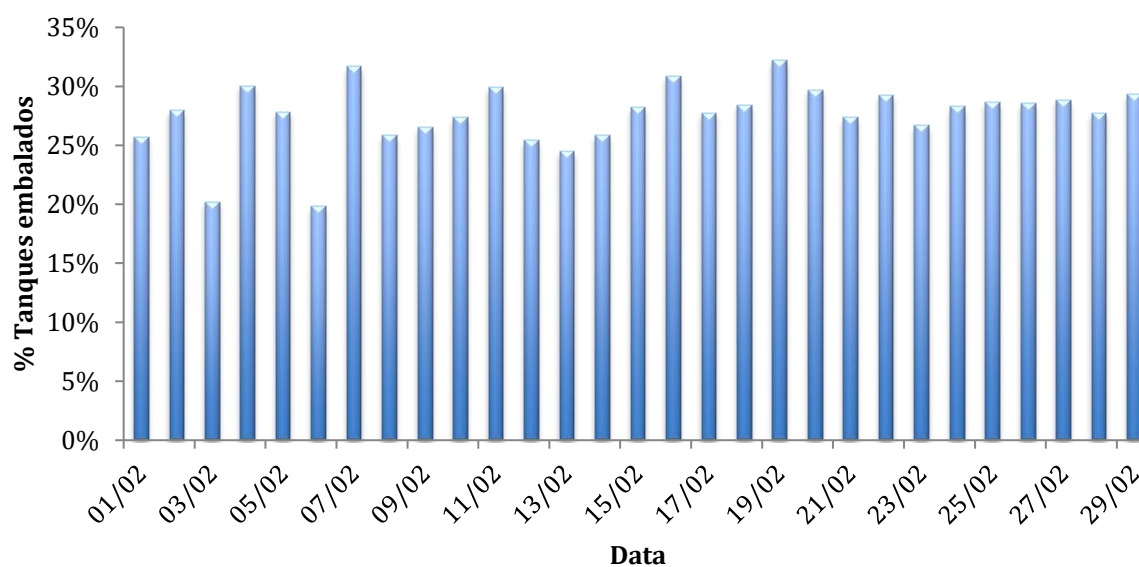


Figura 68 – Percentagem do tipo de produto tanque, embalado durante o mês de fevereiro.

Data	Quantidade peças embaladas	% peças embaladas	Data	Quantidade peças embaladas	% peças embaladas
01/03	2396	27%	17/03	2033	28%
02/03	2668	32%	18/03	2103	28%
03/03	2203	28%	19/03	2234	30%
04/03	1984	26%	20/03	1766	24%
05/03	1817	25%	21/03	1777	23%
06/03	1963	27%	22/03	1787	23%
07/03	2251	24%	23/03	1112	16%
08/03	1903	26%	24/03	2138	29%
09/03	2035	26%	25/03	1520	20%
10/03	2349	30%	26/03	2123	28%
11/03	1932	26%	27/03	1770	29%
12/03	1896	26%	28/03	1816	24%
13/03	2075	28%	29/03	1625	20%
14/03	2065	27%	30/03	1602	22%
15/03	2697	31%	31/03	1527	23%
16/03	2463	30%			

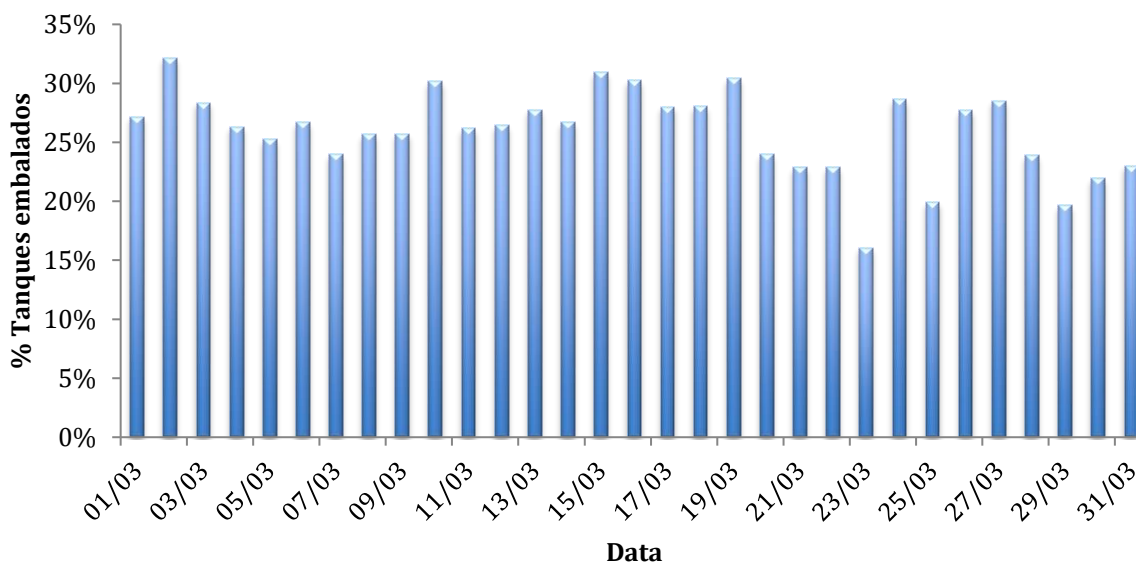


Figura 69 – Percentagem do tipo de produto tanque, embalado durante o mês de março.

Data	Quantidade peças embaladas	% peças embaladas	Data	Quantidade peças embaladas	% peças embaladas
01/04	2120	23%	17/04	1950	24%
02/04	1818	24%	18/04	1497	19%
03/04	1979	25%	19/04	1511	20%
04/04	1430	18%	20/04	2178	27%
05/04	1740	24%	21/04	1713	20%
06/04	1498	20%	22/04	2097	27%
07/04	1887	24%	23/04	1695	21%
08/04	1375	19%	24/04	1681	22%
09/04	1575	21%	25/04	1588	21%
10/04	1928	26%	26/04	2143	27%
11/04	1475	18%	27/04	2073	26%
12/04	2210	26%	28/04	2019	27%
13/04	1600	22%	29/04	1785	25%
14/04	2209	29%	30/04	147	30%
15/04	1892	24%			
16/04	1776	24%			

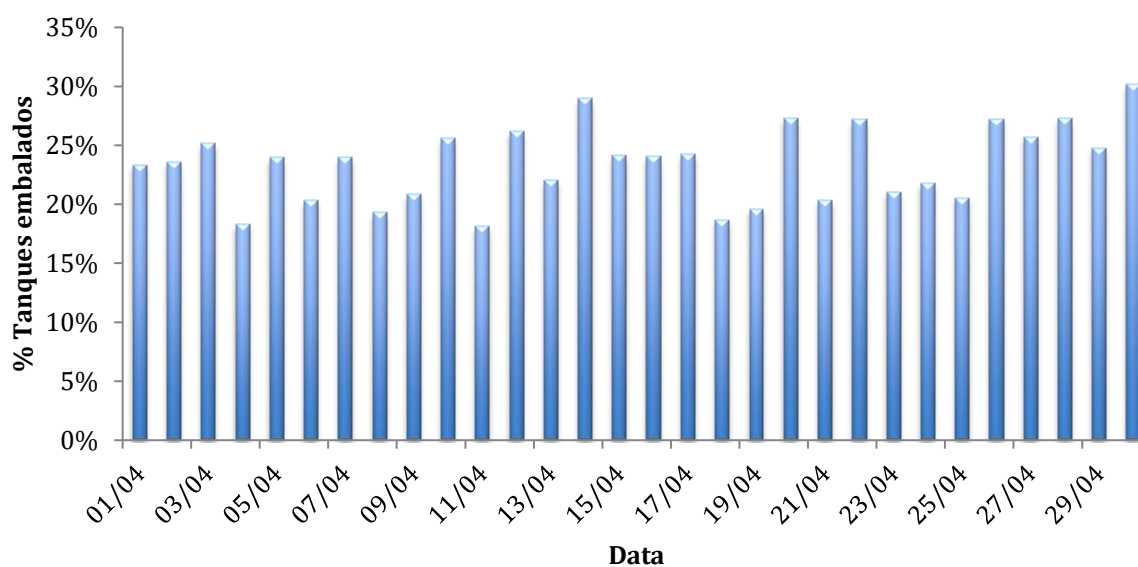


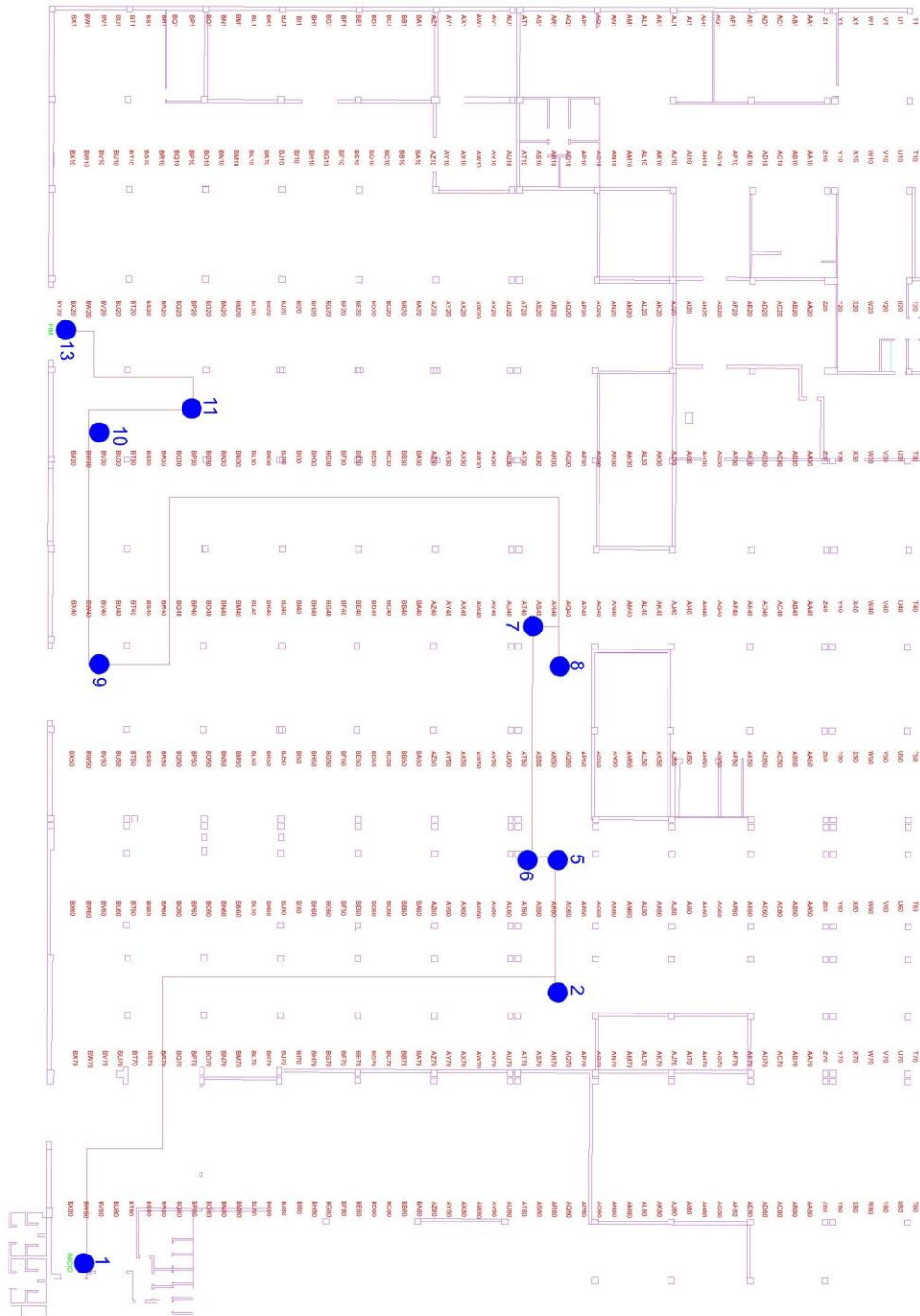
Figura 70 – Percentagem do tipo de produto tanque, embalado durante o mês de abril.



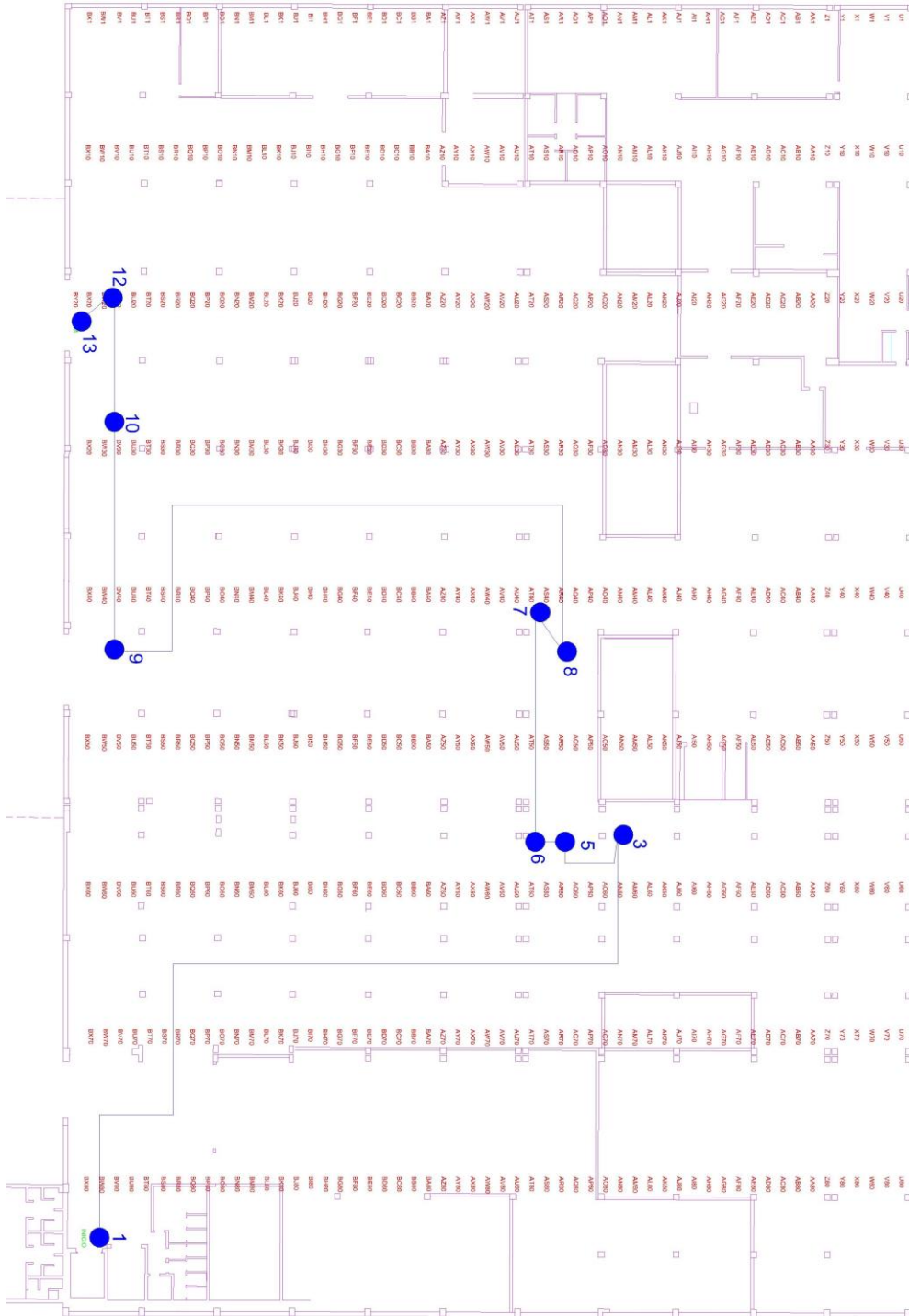
## **Anexo 2**

### **Percursos registados por tipo de produto: o estado atual, o cenário 1 e o cenário 2**

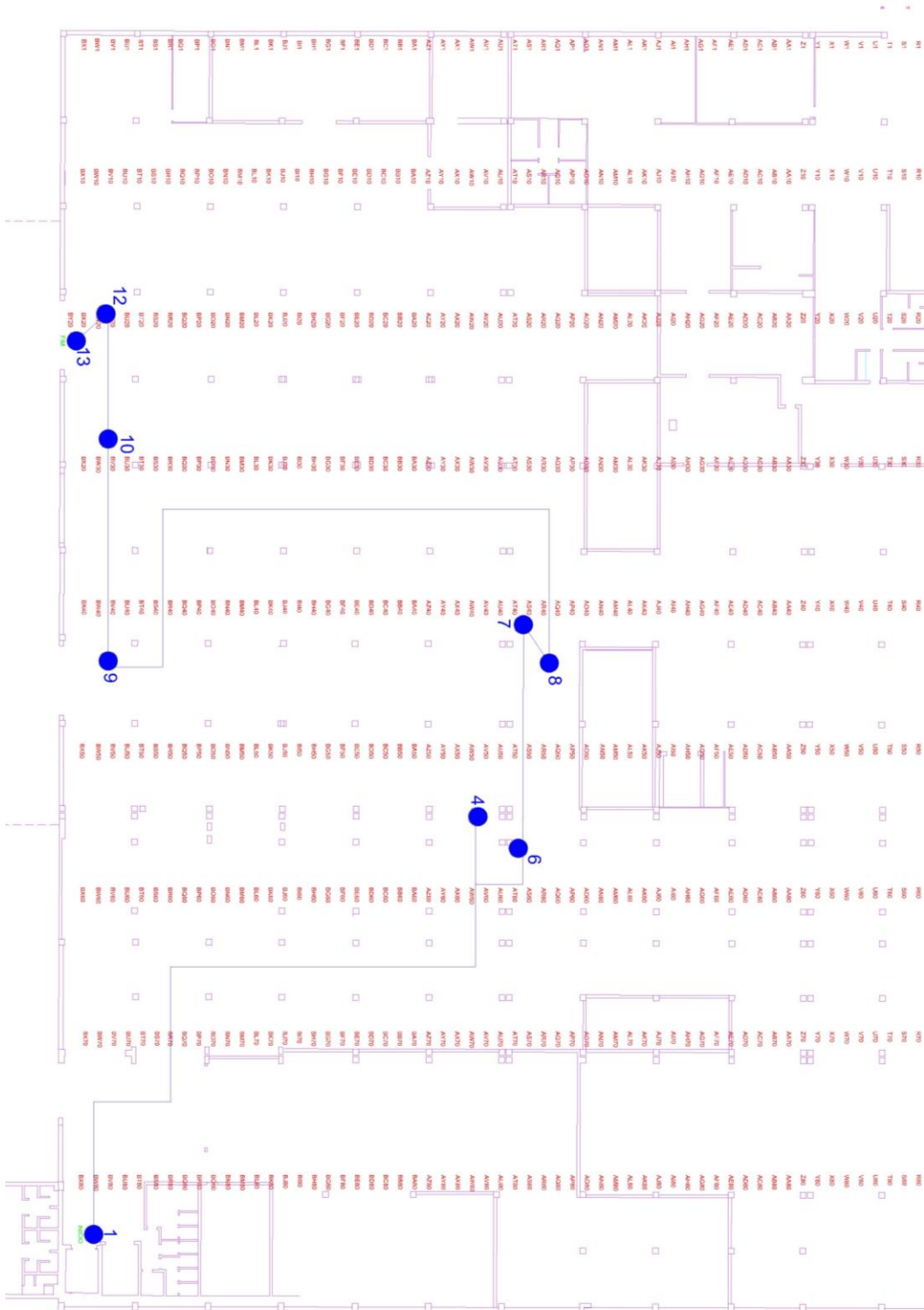
Percurso do produto lavatório: Estado Atual.



Percurso do produto lavatório: Cenário 1.

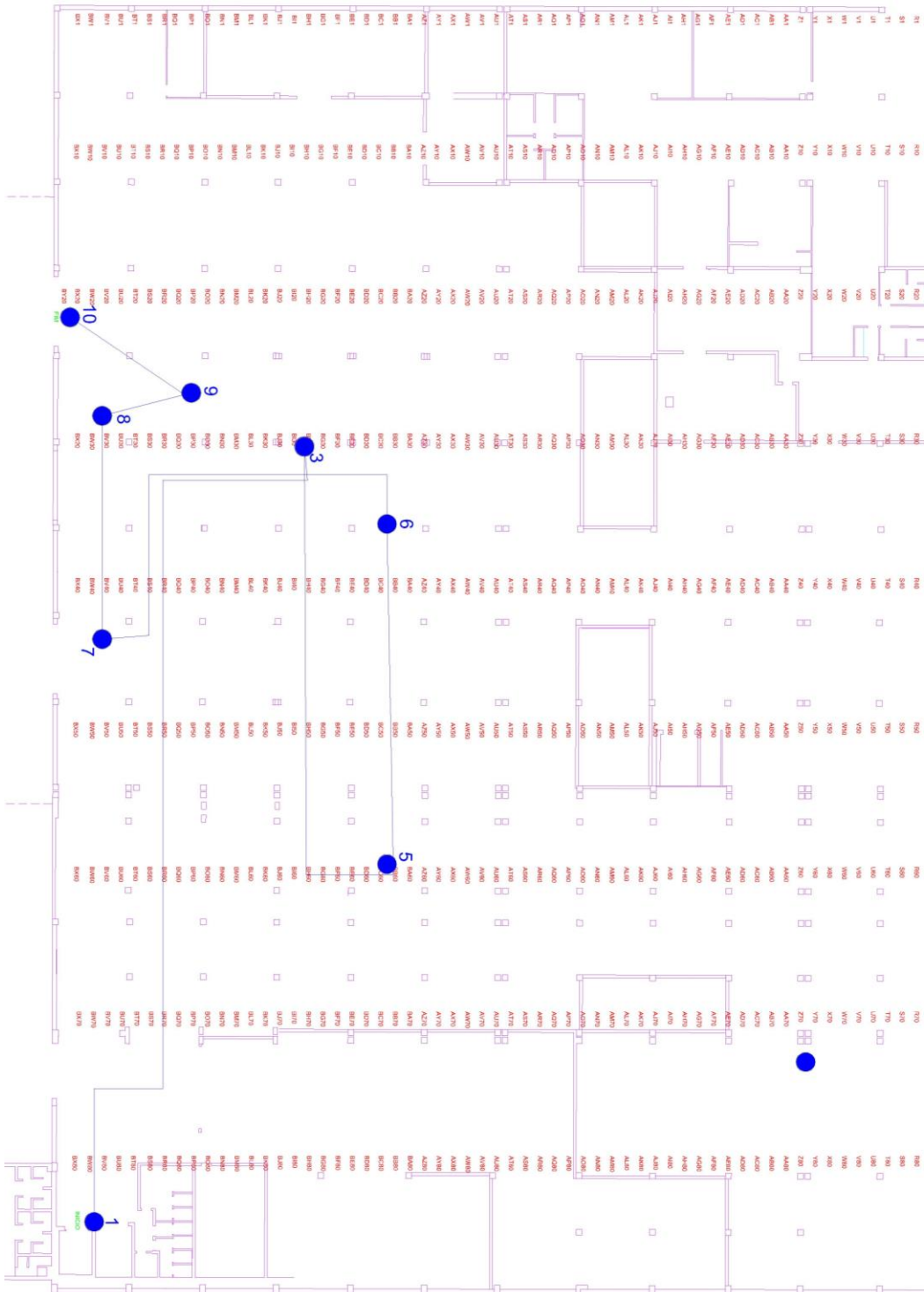


Percurso do produto lavatório: Cenário 2.

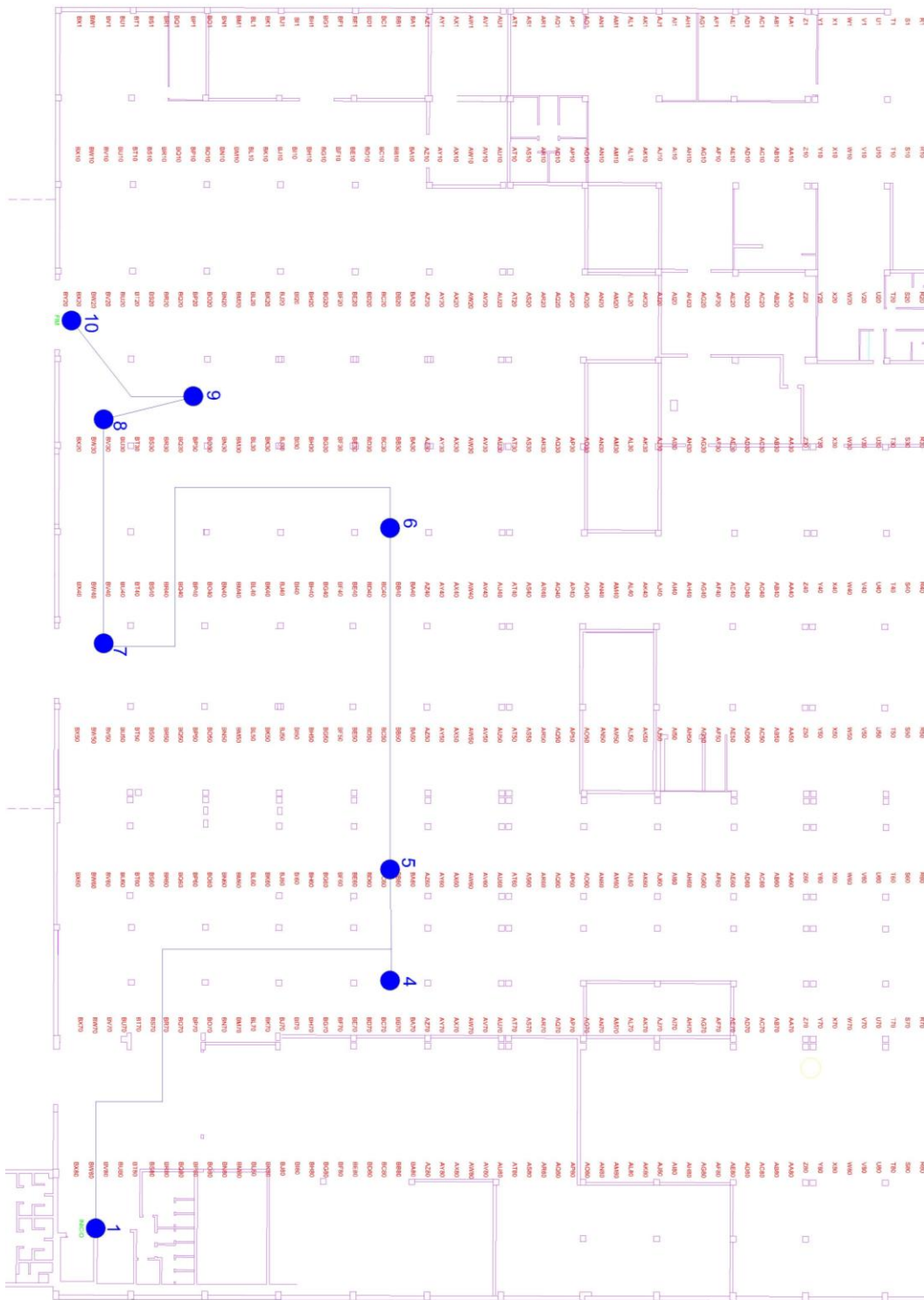




Percurso do produto sanita: Cenário 1.

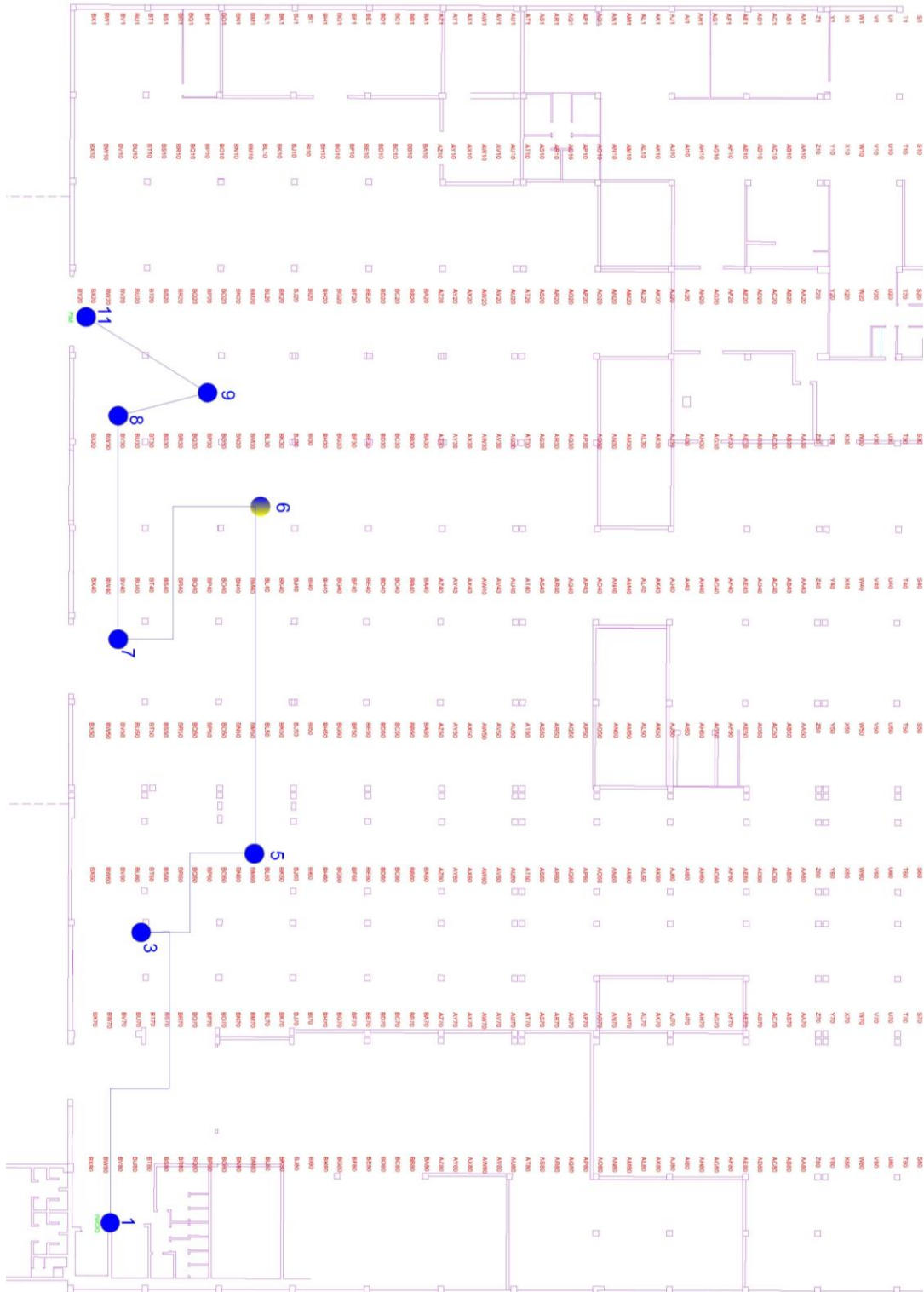


Percurso do produto sanitário: Cenário 2.





Percurso do produto tanque: Cenário 1.

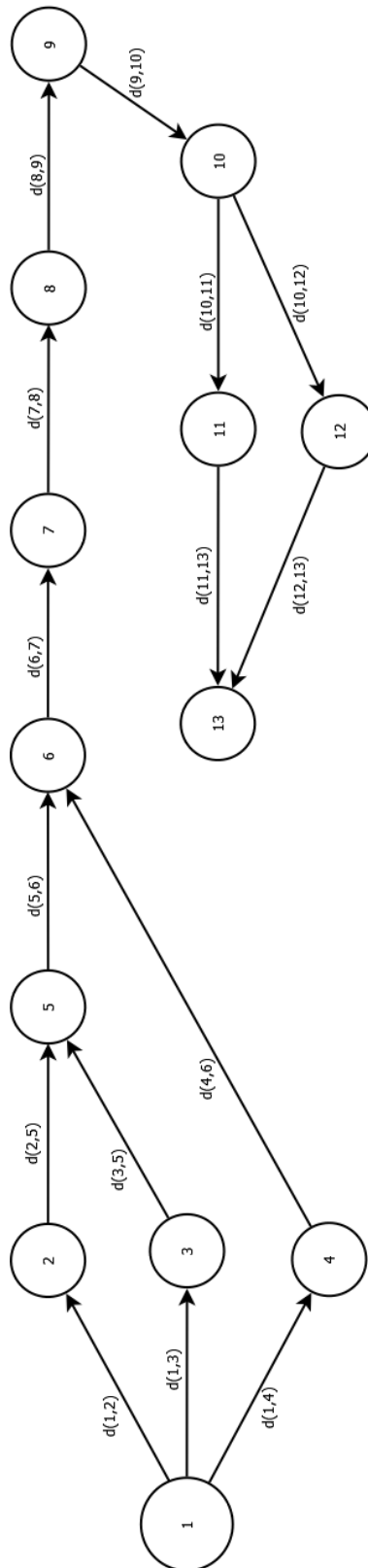




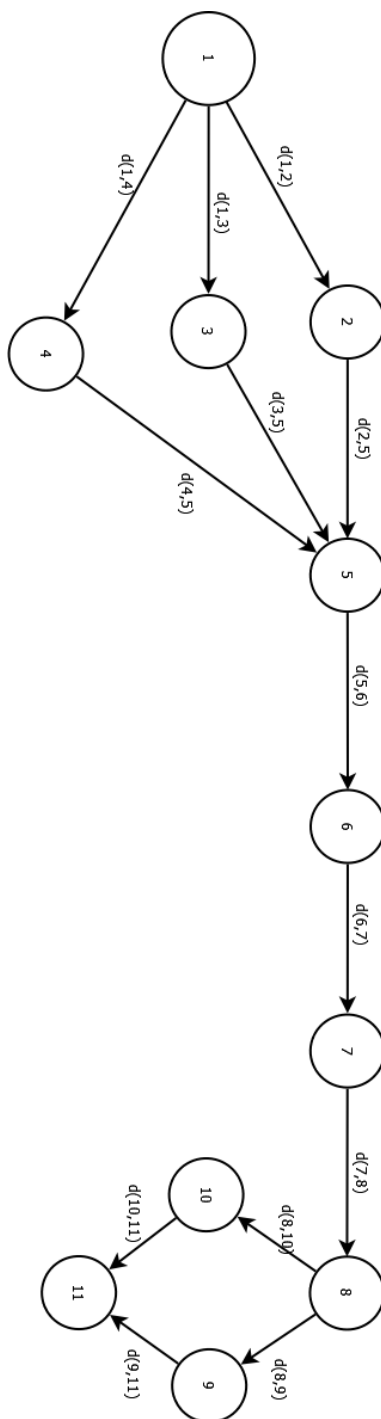
## **Anexo 3**

**Rede construída por tipo de produto:  
lavatório, tanque e sanita**

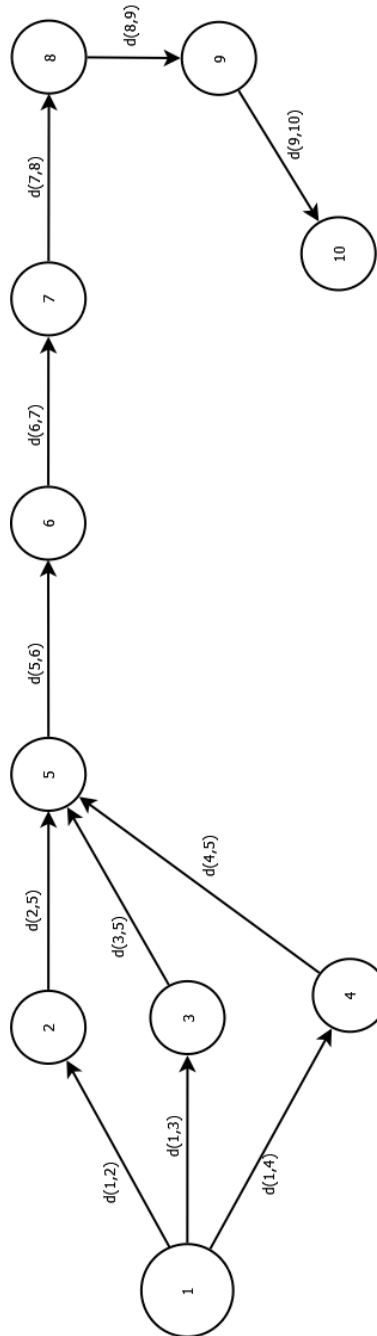
Rede do produto Lavatório.



Rede do produto Tanque.



Rede do Produto Sanita:



## **Anexo 4**

### **Registo dos percursos de uma palete com 24 peças**

Produto tanque: Base 1 palete com 24 peças.

DATA	Localização Inicial	Localização Final	Percorso total (m)	Percorso total (Cen 1)	Percorso total (Cen 2)	nº peças boas (F50)	nº peças boas (F55)	Total peças boas	Nº médio de paletes (peças boas)	Percorso total (Peças BOAS) (m)	Percorso total (Cen 1)	Percorso total (Cen 2)
12/01/2016	BW85	BY22	120	113	112	1507	10	1517	63	7560	7119	7056
24/01/2016	BW85	BY22	120	113	112	1340	152	1492	62	7440	7006	6944
07/02/2016	BW85	BY22	120	113	112	1498	466	1964	81	9720	9153	9072
18/02/2016	BW85	BY22	120	113	112	1325	198	1523	63	7560	7119	7056
21/03/2016	BW85	BY22	120	113	112	1615	162	1777	74	8880	8362	8288
30/03/2016	BW85	BY22	120	113	112	1264	30	1294	53	6360	5989	5936
04/03/2016	BW85	BY22	120	113	112	1236	12	1248	52	6240	5876	5824
01/04/2016	BW85	BY22	120	113	112	1904	47	1951	81	9720	9153	9072
27/04/2016	BW85	BY22	120	113	112	1814	94	1908	79	9480	8927	8848
15/04/2016	BW85	BY22	120	113	112	1550	0	1550	64	7680	7232	7168

Produto Sanita: Base 1 paleta com 24 peças.

DATA	Localização Inicial	Localização Final	Percurso total (m)	Percurso total (Cen 1)	Percurso total (Cen 2)	nº peças boas (F50)	Total peças boas	Nº médio de paletes (peças boas)	Percurso total (Peças BOAS) (m)	Percurso total (Cen 1)	Percurso total (Cen 2)
12/01/2016	BW85	BY22	201	186	130	520	571	23	4623	4278	2990
24/01/2016	BW85	BY22	201	186	130	635	670	27	5427	5022	3510
07/02/2016	BW85	BY22	201	186	130	607	681	28	5628	5208	3640
18/02/2016	BW85	BY22	201	186	130	469	515	21	4221	3906	2730
21/03/2016	BW85	BY22	201	186	130	404	518	21	4221	3906	2730
30/03/2016	BW85	BY22	201	186	130	438	605	25	5025	4650	3250
04/03/2016	BW85	BY22	201	186	130	531	696	29	5829	5394	3770
01/04/2016	BW85	BY22	201	186	130	581	805	33	6633	6138	4290
27/04/2016	BW85	BY22	201	186	130	461	557	23	4623	4278	2990
15/04/2016	BW85	BY22	201	186	130	608	661	27	5427	5022	3510

Produto Lavatório: Base 1 palete com 24 peças.

DATA	Localiz ação Inicial	Localizaçã o Final	Percorso total (m)	Percorso total (Cen 1)	Percorso nº peças boas (F50)	nº peças boas (F55)	Total peças boas	Nº médio de paletes (peças boas)	Percorso total (Peças BOAS) (m)	Percorso total (Cen 1)	Percorso total (Cen 2)
12/01/2016	BW85	BY22	176	170	1282	178	1460	60	10560	10200	10020
24/01/2016	BW85	BY22	176	170	981	91	1072	44	7744	7480	7348
07/02/2016	BW85	BY22	176	170	1042	25	1067	44	7744	7480	7348
18/02/2016	BW85	BY22	176	170	1073	133	1206	50	8800	8500	8350
21/03/2016	BW85	BY22	176	170	864	143	1007	41	7216	6970	6847
30/03/2016	BW85	BY22	176	170	692	770	1462	60	10560	10200	10020
04/03/2016	BW85	BY22	176	170	979	241	1220	50	8800	8500	8350
01/04/2016	BW85	BY22	176	170	1371	99	1470	61	10736	10370	10187
27/04/2016	BW85	BY22	176	170	937	102	1039	43	7568	7310	7181
15/04/2016	BW85	BY22	176	170	1039	56	1095	45	7920	7650	7515

## **Anexo 5**

### **Matriz de coordenadas**

Matriz elaborada em Microsoft Excel para determinação de coordenadas (excerto).

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15
B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15
C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15
D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15
E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	E13	E14	E15
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15
H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15
I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	I11	I12	I13	I14	I15
J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12	J13	J14	J15
K1	K2	K3	K4	K5	K6	K7	K8	K9	K10	K11	K12	K13	K14	K15
L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	L13	L14	L15
M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15
N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11	N12	N13	N14	N15
O1	O2	O3	O4	O5	O6	O7	O8	O9	O10	O11	O12	O13	O14	O15
P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15
Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15
R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15
S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15
T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15
U1	U2	U3	U4	U5	U6	U7	U8	U9	U10	U11	U12	U13	U14	U15
V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15
W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15
X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15
Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10	Y11	Y12	Y13	Y14	Y15
Z1	Z2	Z3	Z4	Z5	Z6	Z7	Z8	Z9	Z10	Z11	Z12	Z13	Z14	Z15

## **Anexo 6**

### **Código desenvolvido em Matlab**

O seguinte código corresponde à implementação da aplicação em Matlab:

```
function varargout = tesel(varargin)
% TESEL MATLAB code for tesel.fig
%   TESEL, by itself, creates a new TESEL or raises the existing
%   singleton*.
%
%   H = TESEL returns the handle to a new TESEL or the handle to
%   the existing singleton*.
%
%   TESEL('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in TESEL.M with the given input arguments.
%
%   TESEL('Property','Value',...) creates a new TESEL or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before tesel_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to tesel_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help tesel

% Last Modified by GUIDE v2.5 14-Dec-2016 15:01:55

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @tesel_OpeningFcn, ...
                  'gui_OutputFcn',  @tesel_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before tesel is made visible.
function tesel_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
```

---

```
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% varargin command line arguments to tesel (see VARARGIN)

% Choose default command line output for tesel
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes tesel wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = tesel_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% -----
function Untitled_1_Callback(hObject, eventdata, handles)
% hObject handle to Untitled_1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% -----
function Untitled_6_Callback(hObject, eventdata, handles)
% hObject handle to Untitled_6 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% -----
function Untitled_7_Callback(hObject, eventdata, handles)
% hObject handle to Untitled_7 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% -----
function Untitled_8_Callback(hObject, eventdata, handles)
% hObject handle to Untitled_8 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

```

% -----
function Untitled_15_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_15 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function Untitled_16_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_16 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
h=msgbox('Daniela Neves Mestrado/Engenharia e Gestão Industrial 2016','Acerca')

% -----
function Untitled_12_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function Untitled_13_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function Untitled_14_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_14 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function Untitled_9_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function Untitled_10_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function Untitled_11_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```

---

```

% handles      structure with handles and user data (see GUIDATA)

% -----
function Untitled_2_Callback(hObject, eventdata, handles)
% hObject      handle to Untitled_2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% -----
function Untitled_3_Callback(hObject, eventdata, handles)
% hObject      handle to Untitled_3 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% -----
function Untitled_4_Callback(hObject, eventdata, handles)
% hObject      handle to Untitled_4 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% -----
function Untitled_5_Callback(hObject, eventdata, handles)
% hObject      handle to Untitled_5 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

exit;

function edit1_Callback(hObject, eventdata, handles)
% hObject      handle to edit1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%         str2double(get(hObject,'String')) returns contents of edit1 as a double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```
function edit2_Callback(hObject, eventdata, handles)
% hObject      handle to edit2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%         str2double(get(hObject,'String')) returns contents of edit2 as a double

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)
% hObject      handle to edit3 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%         str2double(get(hObject,'String')) returns contents of edit3 as a double

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit3 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in listbox2.
function listbox2_Callback(hObject, eventdata, handles)
% hObject      handle to listbox2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
```

---

```

% Hints: contents = cellstr(get(hObject,'String')) returns listBox2 contents as
cell array
%         contents{get(hObject,'Value')} returns selected item from listBox2

items = get(hObject,'String');
index_selected = get(hObject,'Value');
item_selected = items{index_selected};
display(item_selected);

subsetA=xlsread(filename, sheet, xlRange1);
set(handles.edit4,'String',subsetA)

% --- Executes during object creation, after setting all properties.
function listBox2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to listBox2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: listBox controls usually have a white background on Windows.
%         See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

set(hObject,'String',{'01/01/2016';'02/01/2016';'03/01/2016';'04/01/2016';'05/01/
/2016';'06/01/2016';'07/01/2016';'08/01/2016';'09/01/2016';'10/01/2016';'11/01/2
016';'12/01/2016';'13/01/2016';'14/01/2016';'15/01/2016';'16/01/2016';'17/01/201
6';'18/01/2016';'19/01/2016';'20/01/2016';'21/01/2016';'22/01/2016';'23/01/2016'
;'24/01/2016';'25/01/2016';'26/01/2016';'27/01/2016';'28/01/2016';'29/01/2016';'
30/01/2016';'31/01/2016';'01/02/2016';'02/02/2016';'03/02/2016';'04/02/2016';'05
/02/2016';'06/02/2016';'07/02/2016';'08/02/2016';'09/02/2016';'10/02/2016';'11/0
2/2016';'12/02/2016';'13/02/2016';'14/02/2016';'15/02/2016';'16/02/2016';'17/02/
2016';'18/02/2016';'19/02/2016';'20/02/2016';'21/02/2016';'22/02/2016';'23/02/20
16';'24/02/2016';'25/02/2016';'26/02/2016';'27/02/2016';'28/02/2016';'29/02/2016
';'01/03/2016';'02/03/2016';'03/03/2016';'04/03/2016';'05/03/2016';'06/03/2016';
'07/03/2016';'08/03/2016';'09/03/2016';'10/03/2016';'11/03/2016';'12/03/2016';'1
3/03/2016';'14/03/2016';'15/03/2016';'16/03/2016';'17/03/2016';'18/03/2016';'19/
03/2016';'20/03/2016';'21/03/2016';'22/03/2016';'23/03/2016';'24/03/2016';'25/03
/2016';'26/03/2016';'27/03/2016';'28/03/2016';'29/03/2016';'30/03/2016';'31/03/2
016';'01/04/2016';'02/04/2016';'03/04/2016';'04/04/2016';'05/04/2016';'06/04/201
6';'07/04/2016';'08/04/2016';'09/04/2016';'10/04/2016';'11/04/2016';'12/04/2016'
;'13/04/2016';'14/04/2016';'15/04/2016';'16/04/2016';'17/04/2016';'18/04/2016';'
19/04/2016';'20/04/2016';'21/04/2016';'22/04/2016';'23/04/2016';'24/04/2016';'25
/04/2016';'26/04/2016';'27/04/2016';'28/04/2016';'29/04/2016';'30/04/2016'});

set(handles.listBox2,'Value',2)

```

```

function edit4_Callback(hObject, eventdata, handles)

% hObject      handle to edit4 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%         str2double(get(hObject,'String')) returns contents of edit4 as a double

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit4 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

handles
% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit5_Callback(hObject, eventdata, handles)
% hObject      handle to edit5 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit5 as text
%         str2double(get(hObject,'String')) returns contents of edit5 as a double

% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit5 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

---

```
function edit6_Callback(hObject, eventdata, handles)
% hObject      handle to edit6 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
%         str2double(get(hObject,'String')) returns contents of edit6 as a double

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit6 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)
% hObject      handle to edit7 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit7 as text
%         str2double(get(hObject,'String')) returns contents of edit7 as a double

% --- Executes during object creation, after setting all properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit7 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit8_Callback(hObject, eventdata, handles)
% hObject      handle to edit8 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
```

```
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit8 as text
%         str2double(get(hObject,'String')) returns contents of edit8 as a double

% --- Executes during object creation, after setting all properties.
function edit8_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit8 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit9_Callback(hObject, eventdata, handles)
% hObject      handle to edit9 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit9 as text
%         str2double(get(hObject,'String')) returns contents of edit9 as a double

% --- Executes during object creation, after setting all properties.
function edit9_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit9 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit10_Callback(hObject, eventdata, handles)
% hObject      handle to edit10 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
```

---

```
% Hints: get(hObject,'String') returns contents of edit10 as text
%         str2double(get(hObject,'String')) returns contents of edit10 as a
double
```

```
% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit11_Callback(hObject, eventdata, handles)
% hObject    handle to edit11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of edit11 as text
%         str2double(get(hObject,'String')) returns contents of edit11 as a
double
```

```
% --- Executes during object creation, after setting all properties.
function edit11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes on button press in radiobutton1.
function radiobutton1_Callback(hObject, eventdata, handles)
handles.fileName=uigetfile('Dados.xlsx')
guidata(hObject,handles)
```

```
setPopupMenuString(handles.popupmenu5,eventdata,handles)
```

```
function setPopupMenuString(hObject,eventdata,handles, numbers)
fileName=handles.fileName;
[numbers,colNames]=xlsread('Dados.xlsx','mulav')
```

```

set(hObject,'string',colNames);
selected_item=get(hObject,'value');
set(handles.edit14,'string',numbers(selected_item));
guidata(hObject, handles);

fileName='Dados.xlsx'
%Dados.xlsx
lambda=xlsread('Dados.xlsx','lambda','B2');
set(handles.edit1,'String',lambda);

servidores=xlsread('Dados.xlsx','lambda','B3');
set(handles.edit15,'String',servidores);

setPopupMenuString19(handles.popupmenu5,eventdata,handles);

function setPopupMenuString19(hObject,eventdata,handles, numbers);
fileName=handles.fileName;
[numbers,colNames]=xlsread('Dados.xlsx','P0_lav');
set(hObject,'string',colNames);
selected_item=get(hObject,'value');
set(handles.edit11,'string',numbers(selected_item));
guidata(hObject, handles);
%set(handles.edit11,'String',P0)

P0=str2double(get(handles.edit11,'string'));
lambda=str2double(get(handles.edit1,'string'));
mu=str2double(get(handles.edit14,'string'));
S=str2double(get(handles.edit15,'string'));

%Equação 1: Tempos entre chegadas
T_cheg=1/lambda;
set(handles.edit13,'String',T_cheg);

%Equação 2:
Rho=lambda/(mu*S);
set(handles.edit6,'String',Rho);

%Equação 3: Calculos intermédios para P0
%CI_1=(1/factorial(0))*(lambda/mu)^0
%CI_2=((1/factorial(S))*(lambda/mu)^S)*((1/(1-(lambda/mu*S))))
%P0=1/(CI_1+CI_2)
%P0=1-Rho

%Equação 3:
T_serv=1/mu
set(handles.edit5,'String',T_serv)

%Equação 5:
L=(lambda/(mu-lambda))
set(handles.edit8,'String',L)

%Equação 6:
Lq=L-1*(1-P0)

```

---

```

set(handles.edit7,'String',Lq)

%Equação 7:
Wq=Lq/lambda
set(handles.edit9,'String',Wq)

%Equação 8:
W=(1/(mu-lambda))
set(handles.edit10,'String',W)
guidata(hObject, handles);

% hObject    handle to radiobutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of radiobutton1
%%nota : set(handles.edit17,'String','Coluna');

% --- Executes on button press in radiobutton2.
function radiobutton2_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of radiobutton2

handles.fileName=uigetfile('Dados.xlsx')
guidata(hObject,handles)
setPopupMenuString2(handles.popupmenu5,eventdata,handles)

function setPopupMenuString2(hObject,eventdata,handles, numbers)

fileName=handles.fileName;
[numbers,colNames]=xlsread('Dados.xlsx','musan')
set(hObject,'string',colNames);
selected_item=get(hObject,'value');
set(handles.edit14,'string',numbers(selected_item));
guidata(hObject, handles);

setPopupMenuString03(handles.popupmenu5,eventdata,handles);

function setPopupMenuString03(hObject,eventdata,handles, numbers);
fileName=handles.fileName;
[numbers,colNames]=xlsread('Dados.xlsx','P0_san');
set(hObject,'string',colNames);
selected_item=get(hObject,'value');
set(handles.edit11,'string',numbers(selected_item));
guidata(hObject, handles);
%set(handles.edit11,'String',P0)

fileName='Dados.xlsx'

lambda=xlsread('Dados.xlsx','lambda','C2')

```

```

set(handles.edit1,'String',lambda)

servidores=xlsread('Dados.xlsx','lambda','C3')
set(handles.edit15,'String',servidores)

P0=str2double(get(handles.edit11,'string'));
lambda=str2double(get(handles.edit1,'string'));
mu=str2double(get(handles.edit14,'string'));
S=str2double(get(handles.edit15,'string'));

%Equação 1: Tempos entre chegadas
T_cheg=1/lambda
set(handles.edit13,'String',T_cheg)

%Equação 2: Calculos intermédios para P0
%CI_1=(1/factorial(0))*(lambda/mu)^0
%CI_2=((1/factorial(S))*(lambda/mu)^S)*((1/(1-(lambda/mu*S))))
%P0=1/(CI_1+CI_2)
%set(handles.edit11,'String',P0)

%Equação 3:
T_serv=1/mu
set(handles.edit5,'String',T_serv)

%Equação 4:
Rho=lambda/(mu*S)
set(handles.edit6,'String',Rho)

%Equação 5:
L=(lambda/(mu-lambda))
set(handles.edit8,'String',L)

%Equação 6:
Lq=L-1*(1-P0)
set(handles.edit7,'String',Lq)

%Equação 7:
Wq=Lq/lambda
set(handles.edit9,'String',Wq)

%Equação 8:
W=(1/(mu-lambda))
set(handles.edit10,'String',W)

%set(handles.edit23,'String','BX85')
%set(handles.edit22,'String','BY22')

% --- Executes on button press in radiobutton3.
function radiobutton3_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.fileName=uigetfile('Dados.xlsx')
guidata(hObject,handles)

```

```
setPopupMenuString3(handles.popupmenu5,eventdata,handles)
```

```
function setPopupMenuString3(hObject,eventdata,handles, numbers)
```

```
fileName=handles.fileName;
[numbers,colNames]=xlsread('Dados.xlsx','mutq')
set(hObject,'string',colNames);
selected_item=get(hObject,'value');
set(handles.edit14,'string',numbers(selected_item));
guidata(hObject, handles);
```

```
setPopupMenuString01(handles.popupmenu5,eventdata,handles);
```

```
function setPopupMenuString01(hObject,eventdata,handles, numbers);
```

```
fileName=handles.fileName;
[numbers,colNames]=xlsread('Dados.xlsx','P0_tq');
set(hObject,'string',colNames);
selected_item=get(hObject,'value');
set(handles.edit11,'string',numbers(selected_item));
guidata(hObject, handles);
```

```
fileName='Dados.xlsx'
```

```
lambda=xlsread('Dados.xlsx','lambda','D2')
```

```
set(handles.edit1,'String',lambda)
lambda=str2double(get(handles.edit1,'string'));
```

```
servidores=xlsread('Dados.xlsx','lambda','D3')
set(handles.edit15,'String',servidores)
```

```
mu=str2double(get(handles.edit14,'string'));
P0=str2double(get(handles.edit11,'string'));
S=str2double(get(handles.edit15,'string'));
```

```
%Equação 1: Tempos entre chegadas
T_cheg=1/lambda
set(handles.edit13,'String',T_cheg)
```

```
%Equação 2: Calculos intermédios para P0
%CI_1=(1/factorial(0))*(lambda/mu)^0
%CI_2=((1/factorial(S))*(lambda/mu)^S)*((1/(1-(lambda/mu*S))))
%P0=1/(CI_1+CI_2)
%set(handles.edit11,'String',P0)
```

```
%Equação 3:
T_serv=1/mu
set(handles.edit5,'String',T_serv)
```

```
%Equação 4:
Rho=lambda/(mu*S)
set(handles.edit6,'String',Rho)
```

```

%Equação 5:
L=lambda/(mu-lambda)
set(handles.edit8,'String',L)

%Equação 6:
Lq=L-1*(1-P0)
set(handles.edit7,'String',Lq)

%Equação 7:
Wq=Lq/lambda
set(handles.edit9,'String',Wq)

%Equação 8:
W=Wq+(1/mu)

set(handles.edit10,'String',W)

%set(handles.edit23,'String','BX85')
%set(handles.edit22,'String','BY22')
% Hint: get(hObject,'Value') returns toggle state of radiobutton3

% --- Executes on button press in radiobutton4.
function radiobutton4_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of radiobutton4

function edit13_Callback(hObject, eventdata, handles)
% hObject    handle to edit13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit13 as text
%        str2double(get(hObject,'String')) returns contents of edit13 as a
double

% --- Executes during object creation, after setting all properties.
function edit13_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

---

```

% -----
function Untitled_17_Callback(hObject, eventdata, handles)
% hObject      handle to Untitled_17 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

function edit14_Callback(hObject, eventdata, handles, numbers)
% hObject      handle to edit14 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit14 as text
%        str2double(get(hObject,'String')) returns contents of edit14 as a
double

% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles,numbers)
% hObject      handle to edit14 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit15_Callback(hObject, eventdata, handles)
% hObject      handle to edit15 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit15 as text
%        str2double(get(hObject,'String')) returns contents of edit15 as a
double

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit15 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

```

```
% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit18_Callback(hObject, eventdata, handles)
% hObject handle to edit18 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit18 as text
% str2double(get(hObject,'String')) returns contents of edit18 as a
double

% --- Executes during object creation, after setting all properties.
function edit18_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit18 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit19_Callback(hObject, eventdata, handles)
% hObject handle to edit19 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit19 as text
% str2double(get(hObject,'String')) returns contents of edit19 as a
double

% --- Executes during object creation, after setting all properties.
function edit19_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit19 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
```

---

end

```
function edit20_Callback(hObject, eventdata, handles)
% hObject      handle to edit20 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit20 as text
%         str2double(get(hObject,'String')) returns contents of edit20 as a
double
```

```
% --- Executes during object creation, after setting all properties.
function edit20_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit20 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit21_Callback(hObject, eventdata, handles)
% hObject      handle to edit21 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit21 as text
%         str2double(get(hObject,'String')) returns contents of edit21 as a
double
```

```
% --- Executes during object creation, after setting all properties.
function edit21_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit21 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

---

```
function edit22_Callback(hObject, eventdata, handles)
% hObject      handle to edit22 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit22 as text
%         str2double(get(hObject,'String')) returns contents of edit22 as a
double

% --- Executes during object creation, after setting all properties.
function edit22_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit22 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit23_Callback(hObject, eventdata, handles)
% hObject      handle to edit23 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit23 as text
%         str2double(get(hObject,'String')) returns contents of edit23 as a
double

% --- Executes during object creation, after setting all properties.
function edit23_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit23 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

---

```

function listBox3_Callback(hObject, eventdata, handles)
% hObject      handle to listBox3 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns listBox3 contents as
cell array
%           contents{get(hObject,'Value')} returns selected item from listBox3
set(handles.edit17,'String',{'Coluna';'Lavatorio';'Sanita';'Tanque';'Bidé';'Tamp
as';'Urinol';'Todas'});

% --- Executes during object creation, after setting all properties.
function listBox3_CreateFcn(hObject, eventdata, handles)
% hObject      handle to listBox3 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: listbox controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

%set(hObject,'String',{'Coluna';'Lavatório';'Sanita';'Tanque';'Bidé';'Tampas';'U
rinol';'Todas'});
%set(handles.listBox3,'Value',2)

% --- Executes during object deletion, before destroying properties.
function radiobutton1_DeleteFcn(hObject, eventdata, handles)
% hObject      handle to radiobutton1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% --- Executes during object creation, after setting all properties.
function radiobutton1_CreateFcn(hObject, eventdata, handles)
% hObject      handle to radiobutton1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB

```

```
% handles      structure with handles and user data (see GUIDATA)

% --- Executes on button press in radiobutton10.
function radiobutton10_Callback(hObject, eventdata, handles)
% hObject      handle to radiobutton10 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

fileName='Dados.xlsx'

lambda=xlsread('Dados.xlsx','lambda','B2')

set(handles.edit1,'String',lambda)

% Hint: get(hObject,'Value') returns toggle state of radiobutton10

% --- Executes on button press in radiobutton11.
function radiobutton11_Callback(hObject, eventdata, handles)
% hObject      handle to radiobutton11 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of radiobutton11

fileName='Dados.xlsx'

a=xlsread('Dados.xlsx','lambda','C2')

set(handles.edit1,'String',a)

function edit27_Callback(hObject, eventdata, handles)
% hObject      handle to edit27 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit27 as text
%        str2double(get(hObject,'String')) returns contents of edit27 as a
double

% --- Executes during object creation, after setting all properties.
function edit27_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit27 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function radiobutton10_CreateFcn(hObject, eventdata, handles)
% hObject    handle to radiobutton10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

function edit28_Callback(hObject, eventdata, handles)
% hObject    handle to edit28 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit28 as text
%        str2double(get(hObject,'String')) returns contents of edit28 as a
double

% --- Executes during object creation, after setting all properties.
function edit28_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit28 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function txtInfo_Callback(hObject, eventdata, handles)
% hObject    handle to edit30 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit30 as text
%        str2double(get(hObject,'String')) returns contents of edit30 as a
double

% --- Executes during object creation, after setting all properties.
```

---

```

function txtInfo_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit30 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.

txtInfo = sprintf('Qual o tipo?');
    set(handles.txtInfo, 'string', txtInfo);

    fprintf(1, 'Coluna');
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

% --- Executes on button press in checkbox10.
function checkbox10_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
checkboxState = get(hObject, 'Value');
    if checkboxState
        set(handles.txtInfo, 'string', 'Coluna');
    else
        set(handles.txtInfo, 'string', 'Coluna');
    end

% Hint: get(hObject, 'Value') returns toggle state of checkbox10

function edit31_Callback(hObject, eventdata, handles)
% hObject    handle to edit31 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of edit31 as text
%         str2double(get(hObject, 'String')) returns contents of edit31 as a
double

% --- Executes during object creation, after setting all properties.
function edit31_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit31 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))

```

```

    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function uibuttongroup1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to uibuttongroup1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% --- Executes when selected object is changed in uibuttongroup1.
function uibuttongroup1_SelectionChangedFcn(hObject, eventdata, handles)
% hObject    handle to the selected object in uibuttongroup1
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
switch get(eventdata.NewValue, 'Tag') % Get Tag of selected object.
    case 'radiobutton13'
        % Code for when radiobutton1 is selected.
        %txtInfo = sprintf('Lavatório');

        % Continue with more cases as necessary.
    otherwise
        % Code for when there is no match.
end
%set(handles.txtInfo, 'String', txtInfo);
return; % from grpRadioButtonGroup_SelectionChangeFcn

% --- Executes on button press in radiobutton13.
function radiobutton13_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

handles.fileName=uigetfile('Dados.xlsx')
guidata(hObject,handles)
setPopupMenuString6(handles.popupmenu6,eventdata,handles)

function setPopupMenuString6(hObject,eventdata,handles, numbers)

fileName=handles.fileName;
[numbers,colNames]=xlsread('Dados.xlsx','PercursoLav1')
set(hObject,'string',colNames);
selected_item=get(hObject,'value');
set(handles.edit52,'string',numbers(selected_item));
guidata(hObject, handles);

setPopupMenuString7(handles.popupmenu6,eventdata,handles)
function setPopupMenuString7(hObject,eventdata,handles, numbers)

fileName=handles.fileName;
[numbers,colNames]=xlsread('Dados.xlsx','PercursoLav2')
set(hObject,'string',colNames);

```

```

selected_item=get(hObject,'value');
set(handles.edit43,'string',numbers(selected_item));
guidata(hObject, handles);

setPopupMenuString8(handles.popupmenu6,eventdata,handles)

function setPopupMenuString8(hObject,eventdata,handles, numbers)

fileName=handles.fileName;
[numbers,colNames]=xlsread('Dados.xlsx','PercursoLav3')
set(hObject,'string',colNames);
selected_item=get(hObject,'value');
set(handles.edit45,'string',numbers(selected_item));

fileName='Dados.xlsx'
%Dados.xlsx
atual=xlsread('Dados.xlsx','lpal','B2');
set(handles.edit61,'String',atual);

cen1=xlsread('Dados.xlsx','lpal','B3');
set(handles.edit59,'String',cen1);

cen2=xlsread('Dados.xlsx','lpal','B4');
set(handles.edit60,'String',cen2);

net=sparse([1 1 1 2 3 4 5 6 7 8 9 10 10 11 12],[2 3 4 5 5 6 6 7 8 9 10 11 12 13
13],[52 60 55 10 7 11 2 16 6 53 16 7 8 14 2],13,13);
[dist,path,pred]=graphshortestpath(net,1,13);
%h = view(biograph(net,[],'ShowWeights','on'))
%set(h.Nodes(path),'Color',[1 0.4 0.4])
%edges = getedgesbynodeid(h,get(h.Nodes(path),'ID'));
%set(edges,'LineColor',[1 0 0])
%set(edges,'LineWidth',1.5)
set(handles.edit34,'String',dist);

%fileName='Dados.xlsx'
%atuall=xlsread('Dados.xlsx','Lav','A1');
%set(handles.edit62,'String',atuall);
set(handles.edit62,'string','BW84-AR66-AR57-AS57-AS41-AR44-BV44-BV28-BP27-
BX21')
set(handles.edit63,'string','BW84-AN56-AR57-AS57-AS41-AR44-BV44-BV28-BV20-
BX21')
set(handles.edit64,'string','BW84-AV54-AR57-AS57-AS41-AR44-BV44-BV28-BV20-
BX21')
set(handles.edit65,'string','BW84-AR66-AR57-AS57-AS41-AR44-BV44-BV28-BV20-
BX21')
set(handles.edit23,'string','BW84')
set(handles.edit55,'string','BX21')

axes(handles.axes2);
imshow('Lav_estado_atual.jpg');

axes(handles.axes1);

```

---

```

imshow('lav22.jpg');

guidata(hObject, handles);

%net=sparse([1 1 1 2 3 4 5 6 7 8 9],[2 3 4 5 5 5 6 7 8 9 10],[78 69 42 41 35 6
24 29 15 5 9],10,10)
%[dist,path,pred]=graphshortestpath(net,1,10)
%h = view(biograph(net,[],'ShowWeights','on'))
%set(h.Nodes(path),'Color',[1 0.4 0.4])
%edges = getedgesbynodeid(h,get(h.Nodes(path),'ID'));
%set(edges,'LineColor',[1 0 0])
%set(edges,'LineWidth',1.5)
%set(handles.edit34,'String',dist)

%axes(handles.axes1);
imshow('Sanita network.jpg');

% Hint: get(hObject,'Value') returns toggle state of radiobutton13

% --- Executes on key press with focus on edit30 and none of its controls.
function txtInfo_KeyPressFcn(hObject, eventdata, handles)
% hObject    handle to edit30 (see GCBO)
% eventdata  structure with the following fields (see
MATLAB.UI.CONTROL.UICONTROL)
%   Key: name of the key that was pressed, in lower case
%   Character: character interpretation of the key(s) that was pressed
%   Modifier: name(s) of the modifier key(s) (i.e., control, shift) pressed
% handles    structure with handles and user data (see GUIDATA)

% --- If Enable == 'on', executes on mouse press in 5 pixel border.
% --- Otherwise, executes on mouse press in 5 pixel border or over edit30.
function txtInfo_ButtonDownFcn(hObject, eventdata, handles)
% hObject    handle to edit30 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes during object creation, after setting all properties.
function radiobutton2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to radiobutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% --- Executes on button press in radiobutton16.
function radiobutton16_Callback(hObject, eventdata, handles)

```

```
% hObject    handle to radiobutton16 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of radiobutton16

fileName='Dados.xlsx'
a=xlsread('Dados.xlsx','lambda','D2');
set(handles.edit1,'String',a);

function edit34_Callback(hObject, eventdata, handles)
% hObject    handle to edit34 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit34 as text
%        str2double(get(hObject,'String')) returns contents of edit34 as a
double

% --- Executes during object creation, after setting all properties.
function edit34_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit34 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit37_Callback(hObject, eventdata, handles)
% hObject    handle to edit37 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit37 as text
%        str2double(get(hObject,'String')) returns contents of edit37 as a
double

% --- Executes during object creation, after setting all properties.
function edit37_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit37 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
```

---

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function axes1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to axes1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: place code in OpeningFcn to populate axes1

% --- Executes on selection change in popupmenu3.
function popupmenu3_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu3 contents
as cell array
%         contents{get(hObject,'Value')} returns selected item from popupmenu3

% --- Executes during object creation, after setting all properties.
function popupmenu3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in popupmenu4.
function popupmenu4_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu4 contents
as cell array
%         contents{get(hObject,'Value')} returns selected item from popupmenu4

% --- Executes during object creation, after setting all properties.
function popupmenu4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu4 (see GCBO)

```

```
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% --- Executes on selection change in popupmenu5.
function popupmenu5_Callback(hObject, eventdata, handles)
% hObject handle to popupmenu5 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

%set(handles.edit1,'String',lambda)

% function pushbuttonApply_Callback(hObject, eventdata, handles)
% % hObject handle to pushbuttonApply (see GCBO)
% % eventdata reserved - to be defined in a future version of MATLAB
% % handles structure with handles and user data (see GUIDATA)
% data = handles.filename;
% set(handles.uitable1,'Value',xColNum)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu5 contents
as cell array
% contents{get(hObject,'Value')} returns selected item from popupmenu5

% --- Executes during object creation, after setting all properties.
function popupmenu5_CreateFcn(hObject, eventdata, handles)
% hObject handle to popupmenu5 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

---

```

% --- Executes on button press in pushbutton3.

% hObject      handle to pushbutton3 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles, numbers)
% hObject      handle to pushbutton4 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

%Para obter os dados históricos xlsx
%handles.fileName=uigetfile('Dados.xlsx')
%guidata(hObject,handles)
%setPopupmenuString(handles.popupmenu5,eventdata,handles)

%function setPopupmenuString(hObject,eventdata,handles, numbers)

%fileName=handles.fileName;
%[numbers,colNames]=xlsread('Dados.xlsx','mulav')
%set(hObject,'string',colNames);
%selected_item=get(hObject,'value');
%set(handles.edit14,'string',numbers(selected_item));
%guidata(hObject, handles);

function edit38_Callback(hObject, eventdata, handles)
% hObject      handle to edit38 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit38 as text
%        str2double(get(hObject,'String')) returns contents of edit38 as a
double

% --- Executes during object creation, after setting all properties.
function edit38_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit38 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```
function edit39_Callback(hObject, eventdata, handles)
% hObject      handle to edit39 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit39 as text
%         str2double(get(hObject,'String')) returns contents of edit39 as a
double

% --- Executes during object creation, after setting all properties.
function edit39_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit39 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function pushbutton4_CreateFcn(hObject, eventdata, handles)
% hObject      handle to pushbutton4 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% --- Executes during object creation, after setting all properties.
function uipanel9_CreateFcn(hObject, eventdata, handles)
% hObject      handle to uipanel9 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% --- Executes during object creation, after setting all properties.
function uipanel2_CreateFcn(hObject, eventdata, handles)
% hObject      handle to uipanel2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% --- Executes on button press in radiobutton14.
function radiobutton14_Callback(hObject, eventdata, handles)
% hObject      handle to radiobutton14 (see GCBO)
```

---

```

% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
handles.fileName=uigetfile('Dados.xlsx');
guidata(hObject,handles);
setPopupMenuString9(handles.popupmenu6,eventdata,handles);

function setPopupMenuString9(hObject,eventdata,handles, numbers);

fileName=handles.fileName;
[numbers,colNames]=xlsread('Dados.xlsx','PercursoSan');
set(hObject,'string',colNames);
selected_item=get(hObject,'value');
set(handles.edit52,'string',numbers(selected_item));
guidata(hObject, handles);

setPopupMenuString10(handles.popupmenu6,eventdata,handles);

function setPopupMenuString10(hObject,eventdata,handles, numbers);

fileName=handles.fileName;
[numbers,colNames]=xlsread('Dados.xlsx','PercursoSan2');
set(hObject,'string',colNames);
selected_item=get(hObject,'value');
set(handles.edit43,'string',numbers(selected_item));
guidata(hObject, handles);

setPopupMenuString11(handles.popupmenu6,eventdata,handles);

function setPopupMenuString11(hObject,eventdata,handles, numbers);

fileName=handles.fileName;
[numbers,colNames]=xlsread('Dados.xlsx','PercursoSan3');
set(hObject,'string',colNames);
selected_item=get(hObject,'value');
set(handles.edit45,'string',numbers(selected_item));

fileName='Dados.xlsx'
%Dados.xlsx
atuall=xlsread('Dados.xlsx','1pal','D2');
set(handles.edit61,'String',atuall);

cen11=xlsread('Dados.xlsx','1pal','D3');
set(handles.edit59,'String',cen11);

cen22=xlsread('Dados.xlsx','1pal','D4');
set(handles.edit60,'String',cen22);

net=sparse([1 1 1 2 3 4 5 6 7 8 9],[2 3 4 5 5 5 6 7 8 9 10],[78 69 42 41 35 6 24
29 15 5 9],10,10);
[dist,path,pred]=graphshortestpath(net,1,10);
%h = view(biograph(net,[],'ShowWeights','on'))
%set(h.Nodes(path),'Color',[1 0.4 0.4])
%edges = getedgesbynodeid(h,get(h.Nodes(path),'ID'));
%set(edges,'LineColor',[1 0 0])

```

---

```

%set(edges,'LineWidth',1.5)
set(handles.edit34,'String',dist);
guidata(hObject, handles);

set(handles.edit62, 'string', 'BW84-Z73-BB60-BC36-BV44-BV28-BP27-BX21')
set(handles.edit63, 'string', 'BW84-BH31-BB60-BC36-BV44-BV28-BP27-BX21')
set(handles.edit64, 'string', 'BW84-BB67-BB60-BC36-BV44-BV28-BP27-BX21')
set(handles.edit65, 'string', 'BW84-BB67-BB60-BC36-BV44-BV28-BP27-BX21')
set(handles.edit23, 'string', 'BW84')
set(handles.edit55, 'string', 'BX21')

axes(handles.axes2);
imshow('San_estado_atual.jpg');

axes(handles.axes1);
imshow('san.bmp');

% Hint: get(hObject,'Value') returns toggle state of radiobutton14

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

function edit43_Callback(hObject, eventdata, handles)
% hObject    handle to edit43 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit43 as text
%        str2double(get(hObject,'String')) returns contents of edit43 as a
double

% --- Executes during object creation, after setting all properties.
function edit43_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit43 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

---

```
function edit44_Callback(hObject, eventdata, handles)
% hObject      handle to edit44 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit44 as text
%         str2double(get(hObject,'String')) returns contents of edit44 as a
double

% --- Executes during object creation, after setting all properties.
function edit44_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit44 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit50_Callback(hObject, eventdata, handles)
% hObject      handle to edit50 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit50 as text
%         str2double(get(hObject,'String')) returns contents of edit50 as a
double

% --- Executes during object creation, after setting all properties.
function edit50_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit50 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit51_Callback(hObject, eventdata, handles)
% hObject      handle to edit51 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
```

```
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit51 as text
%         str2double(get(hObject,'String')) returns contents of edit51 as a
double

% --- Executes during object creation, after setting all properties.
function edit51_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit51 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit53_Callback(hObject, eventdata, handles)
% hObject      handle to edit53 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit53 as text
%         str2double(get(hObject,'String')) returns contents of edit53 as a
double

% --- Executes during object creation, after setting all properties.
function edit53_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit53 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit45_Callback(hObject, eventdata, handles)
% hObject      handle to edit45 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit45 as text
```

```
%          str2double(get(hObject,'String')) returns contents of edit45 as a
double

% --- Executes during object creation, after setting all properties.
function edit45_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit45 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit52_Callback(hObject, eventdata, handles)
% hObject    handle to edit52 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit52 as text
%         str2double(get(hObject,'String')) returns contents of edit52 as a
double

% --- Executes during object creation, after setting all properties.
function edit52_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit52 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in popupmenu6.
function popupmenu6_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu6 contents
as cell array
%         contents{get(hObject,'Value')} returns selected item from popupmenu6
```

```

% --- Executes during object creation, after setting all properties.
function popupmenu6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit54_Callback(hObject, eventdata, handles)
% hObject    handle to edit54 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit54 as text
%         str2double(get(hObject,'String')) returns contents of edit54 as a
double

% --- Executes during object creation, after setting all properties.
function edit54_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit54 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit55_Callback(hObject, eventdata, handles)
% hObject    handle to edit55 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit55 as text
%         str2double(get(hObject,'String')) returns contents of edit55 as a
double

% --- Executes during object creation, after setting all properties.
function edit55_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit55 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```

```
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit59_Callback(hObject, eventdata, handles)
% hObject      handle to edit59 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit59 as text
%       str2double(get(hObject,'String')) returns contents of edit59 as a
double

% --- Executes during object creation, after setting all properties.
function edit59_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit59 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit60_Callback(hObject, eventdata, handles)
% hObject      handle to edit60 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit60 as text
%       str2double(get(hObject,'String')) returns contents of edit60 as a
double

% --- Executes during object creation, after setting all properties.
function edit60_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit60 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit61_Callback(hObject, eventdata, handles)
% hObject      handle to edit61 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit61 as text
%         str2double(get(hObject,'String')) returns contents of edit61 as a
double

% --- Executes during object creation, after setting all properties.
function edit61_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit61 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in radiobutton15.
function radiobutton15_Callback(hObject, eventdata, handles)
% hObject      handle to radiobutton15 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of radiobutton15
handles.fileName=uigetfile('Dados.xlsx');
guidata(hObject,handles);
setPopupMenuString12(handles.popupmenu6,eventdata,handles);

function setPopupMenuString12(hObject,eventdata,handles, numbers);

fileName=handles.fileName;
[numbers,colNames]=xlsread('Dados.xlsx','PercursoTq');
set(hObject,'string',colNames);
selected_item=get(hObject,'value');
set(handles.edit52,'string',numbers(selected_item));
guidata(hObject, handles);

setPopupMenuString13(handles.popupmenu6,eventdata,handles);

function setPopupMenuString13(hObject,eventdata,handles, numbers);

```

---

```

fileName=handles.fileName;
[numbers,colNames]=xlsread('Dados.xlsx','PercursoTq2');
set(hObject,'string',colNames);
selected_item=get(hObject,'value');
set(handles.edit43,'string',numbers(selected_item));
guidata(hObject, handles);

setPopupMenuString14(handles.popupmenu6,eventdata,handles);

function setPopupMenuString14(hObject,eventdata,handles, numbers);

fileName=handles.fileName;
[numbers,colNames]=xlsread('Dados.xlsx','PercursoTq3');
set(hObject,'string',colNames);
selected_item=get(hObject,'value');
set(handles.edit45,'string',numbers(selected_item));

fileName='Dados.xlsx'
%Dados.xlsx
atuall1=xlsread('Dados.xlsx','1pal','C2');
set(handles.edit61,'String',atuall1);

cen111=xlsread('Dados.xlsx','1pal','C3');
set(handles.edit59,'String',cen111);

cen222=xlsread('Dados.xlsx','1pal','C4');
set(handles.edit60,'String',cen222);

net=sparse([1 1 1 2 3 4 5 6 7 8 8 9 10],[2 3 4 5 5 5 6 7 8 9 10 11 11],[35 25 31
11 14 16 25 19 15 5 2 10 4],11,11);
[dist,path,pred]=graphshortestpath(net,1,11);
%h = view(biograph(net,[],'ShowWeights','on'))
%set(h.Nodes(path),'Color',[1 0.4 0.4])
%edges = getedgesbynodeid(h,get(h.Nodes(path),'ID'));
%set(edges,'LineColor',[1 0 0])
%set(edges,'LineWidth',1.5)
set(handles.edit34,'String',dist);
guidata(hObject, handles);

axes(handles.axes2);
imshow('Estado_Atual_tq.jpg');

axes(handles.axes1);
imshow('tq.bmp');

% --- Executes on button press in pushbutton9.
function pushbutton9_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton9 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

```

```

net=sparse([1 1 1 2 3 4 5 6 7 8 9],[2 3 4 5 5 5 6 7 8 9 10],[78 69 42 41 35 6 24
29 15 5 9],10,10);
[dist,path,pred]=graphshortestpath(net,1,10);
h = view(biograph(net,[],'ShowWeights','on'));
set(h.Nodes(path),'Color',[1 0.4 0.4]);
edges = getedgesbynodeid(h,get(h.Nodes(path),'ID'));
set(edges,'LineColor',[1 0 0]);
set(edges,'LineWidth',1.5);
set(handles.edit34,'String',dist);

set(handles.edit62, 'string', 'BW84-BI68-BK59-BL35-BV44-BV28-BP27-BX21')
set(handles.edit63, 'string', 'BW84-BT64-BK59-BL35-BV44-BV28-BP27-BX21')
set(handles.edit64, 'string', 'BW84-BN69-BK59-BL35-BV44-BV28-BX27-BX21')
set(handles.edit65, 'string', 'BW84-BT64-BK59-BL35-BV44-BV28-BX27-BX21')
set(handles.edit23, 'string', 'BW84')
set(handles.edit55, 'string', 'BX21')

% --- Executes during object creation, after setting all properties.
function radiobutton13_CreateFcn(hObject, eventdata, handles)
% hObject    handle to radiobutton13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% --- Executes on button press in pushbutton12.
function pushbutton12_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
net=sparse([1 1 1 2 3 4 5 6 7 8 8 9 10],[2 3 4 5 5 5 6 7 8 9 10 11 11],[35 25 31
11 14 16 25 19 15 5 2 10 4],11,11);
[dist,path,pred]=graphshortestpath(net,1,11);
h = view(biograph(net,[],'ShowWeights','on'));
set(h.Nodes(path),'Color',[1 0.4 0.4]);
edges = getedgesbynodeid(h,get(h.Nodes(path),'ID'));
set(edges,'LineColor',[1 0 0]);
set(edges,'LineWidth',1.5);
set(handles.edit34,'String',dist);

% --- Executes on button press in pushbutton13.
function pushbutton13_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton13 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
net=sparse([1 1 1 2 3 4 5 6 7 8 9 10 10 11 12],[2 3 4 5 5 6 6 7 8 9 10 11 12 13
13],[52 60 55 10 7 11 2 16 6 53 16 7 8 14 2],13,13);
[dist,path,pred]=graphshortestpath(net,1,13);
h = view(biograph(net,[],'ShowWeights','on'));
set(h.Nodes(path),'Color',[1 0.4 0.4]);
edges = getedgesbynodeid(h,get(h.Nodes(path),'ID'));
set(edges,'LineColor',[1 0 0]);
set(edges,'LineWidth',1.5);
set(handles.edit34,'String',dist);

```

---

```
function edit62_Callback(hObject, eventdata, handles)
% hObject      handle to edit62 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit62 as text
%        str2double(get(hObject,'String')) returns contents of edit62 as a
double

% --- Executes during object creation, after setting all properties.
function edit62_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit62 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit63_Callback(hObject, eventdata, handles)
% hObject      handle to edit63 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit63 as text
%        str2double(get(hObject,'String')) returns contents of edit63 as a
double

% --- Executes during object creation, after setting all properties.
function edit63_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit63 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

---

```

function edit64_Callback(hObject, eventdata, handles)
% hObject      handle to edit64 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit64 as text
%         str2double(get(hObject,'String')) returns contents of edit64 as a
double

% --- Executes during object creation, after setting all properties.
function edit64_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit64 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit65_Callback(hObject, eventdata, handles)
% hObject      handle to edit65 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit65 as text
%         str2double(get(hObject,'String')) returns contents of edit65 as a
double

% --- Executes during object creation, after setting all properties.
function edit65_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit65 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton14.
function pushbutton14_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton14 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

```

```
% --- Executes on button press in radiobutton30.
function radiobutton30_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton30 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of radiobutton30

% --- Executes on button press in radiobutton31.
function radiobutton31_Callback(hObject, eventdata, handles)
% hObject    handle to radiobutton31 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of radiobutton31
```