



# DESENVOLVIMENTO DE UMA FERRAMENTA DE SIMULAÇÃO PARA APOIO À TOMADA DE DECISÃO EM OFICINA AUTOMÓVEL

**DANIEL JORGE FERREIRA DE AZEVEDO TEIXEIRA BASTO**

outubro de 2020

# DESENVOLVIMENTO DE UMA FERRAMENTA DE SIMULAÇÃO PARA APOIO À TOMADA DE DECISÃO EM OFICINA AUTOMÓVEL

Daniel Jorge Ferreira de Azevedo Teixeira Basto

**2020**

Instituto Superior de Engenharia do Porto

Departamento de Engenharia Mecânica

isen

P.PORTO

# DESENVOLVIMENTO DE UMA FERRAMENTA DE SIMULAÇÃO PARA APOIO À TOMADA DE DECISÃO EM OFICINA AUTOMÓVEL

Daniel Jorge Ferreira de Azevedo Teixeira Basto

Estudante n.º 1160068

Dissertação apresentada ao Instituto Superior de Engenharia do Porto para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia e Gestão Industrial, realizada sob a orientação do Doutor Luís Carlos Ramos Nunes Pinto Ferreira e coorientação do Doutor António José Galvão Ramos

**2020**

Instituto Superior de Engenharia do Porto

Departamento de Engenharia Mecânica

isen

P.PORTO

## **AGRADECIMENTOS**

Como todos os trabalhos mais exigentes, não seria possível chegar ao final sem algumas ajudas, por vezes em momentos chave, e similarmente, este trabalho não foge a esse princípio. Desta forma pretendo aqui deixar um agradecimento a todas as pessoas que me ajudaram neste caminho e tornaram possível a realização desta dissertação.

Ao Professor Doutor Luís Pinto Ferreira do Instituto Superior de Engenharia do Porto (ISEP), deixo um especial agradecimento, por ter aceitado orientar o trabalho que desenvolvi, pela sua disponibilidade, e a constante e fantástica motivação para o progresso do mesmo. Agradeço também ao coorientador do ISEP, Professor Doutor António Galvão Ramos, pelas perspetivas e esclarecimentos transmitidos que permitiram melhorar o trabalho desenvolvido.

Naturalmente não posso deixar de agradecer à minha mulher Fátima e aos meus filhos, Nuno e Hugo, pela sua paciência e tolerância às minhas incontáveis ausências familiares, e especial motivação nesta reta final.

Obrigado a todos.

página propositadamente em branco

## RESUMO

A existência de concorrência entre empresas na produção de bens, ou fornecimentos de serviços, está intrínseca nas atividades de todos nós diariamente. Usualmente as empresas tentam criar fatores diferenciadores recorrendo à melhoria dos produtos ou serviços fornecidos, ou então pelo preço ou mesmo pelo fator do tempo de entrega do produto ou execução do serviço. Logicamente todas as empresas procuram conseguir fatores de vantagem sobre os competidores diretos, que permitam melhorar algum dos pontos já referidos.

As ferramentas de simulação como suporte à tomada de decisão, podem ser a vantagem à qual as empresas podem recorrer, por forma a estudarem novos processos ou métodos que permitam alavancar algum (ou mesmo todos) os fatores já referidos.

Como objetivo desta dissertação, foi criada uma ferramenta de apoio à decisão aplicada a uma oficina automóvel que permitiu estudar o resultado da aplicação de algumas estratégias, no funcionamento da mesma e avaliar o seu impacte no resultado atingido. O simulador criado foi desenvolvido recorrendo-se ao *software* Arena para implementação do modelo lógico, e recorrendo ao ambiente de Visual Basic também do *software* Arena, foi criada uma interface para o utilizador, que isola o mesmo das atividades de criação e programação do modelo. Como se pretende que o simulador seja de utilização comum (dado que isola o utilizador da programação de simuladores) foi dada grande atenção à similaridade entre módulos, tanta na apresentação dos dados, como no funcionamento dos botões disponíveis.

Desta forma pretende-se contribuir para o desenvolvimento e utilização de simuladores como suporte à tomada da decisão.

### PALAVRAS-CHAVE

Arena, Estratégia de controlo, Oficina automóvel, Suporte à decisão

página propositadamente em branco

## ABSTRACT

The existence of competition between companies in the production of goods, or the provision of services, is intrinsic in the activities of all of us on a daily basis. Usually companies try to create differentiating factors by resorting to the improvement of the products or services provided, or by the price or even by the factor of the time of delivery of the product or execution of the service. Logically, all companies seek to obtain advantage factors over their direct competitors, which allow to improve some of the points already mentioned.

Simulation tools as a support for decision making can be the advantage that companies can use, to study new processes or methods that allow leveraging some (or even all) of the factors already mentioned.

As a goal of this dissertation, a decision support tool applied to a car repair shop was created, which made it possible to study the result of the application of some strategies, in the operation of the same and to evaluate its impact on the result achieved. The simulator created was developed using the Arena software to implement the logical model, and using the Visual Basic environment also from the Arena software, an user interface was created, which isolates it from the model's creation and programming activities. As the simulator is intended to be in common use (given that it isolates the user from simulator programming) great attention was paid to the similarity between modules, both in the presentation of the data and in the operation of the available buttons.

In this way it is intended to contribute to the development and use of simulators to support decision making.

### KEYWORDS

Arena, Auto repair shop, Control strategy, Decision support

página propositadamente em branco

# ÍNDICE

RESUMO .....	I
ABSTRACT .....	III
ÍNDICE DE FIGURAS .....	IX
ÍNDICE DE TABELAS .....	XIII
LISTAS DE SIGLAS .....	XV
1. INTRODUÇÃO .....	17
1.1. Enquadramento e pertinência .....	17
1.2. Objetivos de investigação.....	19
1.3. Opções metodológicas .....	19
1.4. Estrutura do trabalho .....	20
2. REVISÃO BIBLIOGRÁFICA.....	23
2.1. Evolução da utilização da simulação .....	23
2.2. Metodologias de simulação .....	23
2.2.1. Sistemas dinâmicos .....	24
2.2.2. Dinâmica de sistemas.....	25
2.2.3. Baseada em agentes .....	25
2.2.4. Eventos discretos .....	26
2.3. Classificação de sistemas.....	27
2.3.1. Determinístico/Estocástico .....	27
2.3.2. Estático/Dinâmico .....	28
2.3.3. Discreto/Contínuo.....	28
2.4. Vantagens e desvantagens da simulação.....	28
2.5. Fases de um projeto de simulação.....	30
2.5.1. Primeira etapa – formulação do problema.....	30
2.5.2. Segunda etapa – recolha de informação .....	31
2.5.3. Etapas três, quatro e cinco.....	31
2.5.4. Sexta etapa – experimentação.....	31
2.5.5. Sétima etapa – apresentação de resultados.....	32
2.6. Estudos recorrendo à simulação em setores de serviços .....	32
2.6.1. Setor logística – otimização de rotas .....	33
2.6.2. Setor produção – problema de performance .....	33
2.6.3. Setor saúde – problema de capacidade.....	33
2.6.4. Setor comercial – caixas pagamento em supermercado .....	34
3. SISTEMA DE APOIO À DECISÃO DESENVOLVIDO .....	35
3.1. Descrição de funcionamento de oficina genérica .....	35
3.2. Módulos e funcionamento do sistema de apoio à decisão.....	36

---

3.2.1. Módulo inicial.....	38
3.2.2. Módulo de configuração das estratégias.....	38
3.2.3. Módulo de configuração da chegada dos clientes.....	39
3.2.4. Módulo de configuração dos tempos de processamento dos diferentes processos da oficina.....	40
3.2.5. Módulo de configuração dos tempos de rota.....	42
3.2.6. Módulo de configuração dos parâmetros de simulação.....	43
3.3. Animação do modelo.....	44
3.4. Modelo lógico do sistema de apoio à decisão desenvolvido.....	44
3.4.1. Modelo lógico.....	44
3.4.2. Rotas e estacionamento.....	49
3.5. Programa de controlo da interface do utilizador.....	50
3.5.1. Fluxograma do funcionamento.....	50
3.5.2. Controlo do módulo inicial.....	51
3.5.3. Controlo do módulo de estratégias.....	54
3.5.4. Controlo dos restantes módulos – Botão <i>Read Variables</i> .....	55
3.5.5. Controlo dos restantes módulos – Seleção de distribuição de tempo.....	58
3.5.6. Controlo dos restantes módulos – Botão <i>Change</i> .....	61
4. CONCLUSÕES E TRABALHO FUTURO.....	65
4.1. Resultado atingido.....	65
4.2. Conclusões finais.....	65
4.3. Caracter inovador do trabalho.....	66
4.4. Limitações e trabalho futuro.....	66
REFERÊNCIAS BIBLIOGRÁFICAS.....	69
APÊNDICE A – CÓDIGO DESENVOLVIDO.....	73

página propositadamente em branco



## ÍNDICE DE FIGURAS

Figura 1 - Método hipotético-dedutivo (Marconi & Lakatos, 2003) .....	20
Figura 2 - Artigos publicados por década (Mourtzis et al., 2014) .....	23
Figura 3 - Métodos para estudo de sistemas, adaptado de Law e Kelton (1991) .....	24
Figura 4 - Representação matemática de um sistema "bola saltitante" (Borshchev & Filippov, 2004) .....	25
Figura 5 - Representação de um sistema como dinâmica de sistemas. (Borshchev & Filippov, 2004) .....	25
Figura 6 - Modelo genérico de modelação baseada em agentes (Borshchev & Filippov, 2004) .....	26
Figura 7 - Representação de balcão de atendimento bancário (Borshchev & Filippov, 2004).....	27
Figura 8 - Esquema de classificações adaptado (Rossetti, 2016) .....	27
Figura 9 - Etapas de um projeto de simulação (A M Law, 2008) .....	30
Figura 10 - Utilização de técnicas de simulação (Jahangirian et al., 2010) .....	33
Figura 11 - Fluxograma simples funcionamento oficina automóvel .....	35
Figura 12 - Fluxograma global .....	37
Figura 13 - Módulo inicial do simulador.....	38
Figura 14 - Módulos disponíveis.....	38
Figura 15 - Módulo das estratégias de simulação.....	39
Figura 16 - Módulo Customer Arrivals .....	40
Figura 17 - Lista das distribuições de tempo disponíveis.....	40
Figura 18 - Módulo Process Delay.....	41
Figura 19 - Parâmetros das distribuições.....	42
Figura 20 - Módulo Route Time .....	43
Figura 21 - Módulo Simulation Parameters .....	43
Figura 22 - Simulador em execução .....	44
Figura 23 - Modelo lógico da receção .....	45
Figura 24 - Definição do processo de receção .....	46
Figura 25 - Bloco de decisão do serviço a executar .....	46
Figura 26 - Bloco de execução dos serviços .....	47
Figura 27 - Definição do processo de General Repairs .....	47
Figura 28 - Saída das entidades do simulador.....	48
Figura 29 - Parametrização de variáveis .....	48
Figura 30 - Modelo lógico global.....	49
Figura 31 - Representação oficina.....	50
Figura 32 - Fluxograma dos módulos .....	51
Figura 33 - Fluxograma MultiPage .....	52
Figura 34 - Excerto de código do MultiPage .....	53
Figura 35 - Botões de Run Simulation e Exit .....	53
Figura 36 - Código do botão Exit .....	53
Figura 37 - Código do botão Run.....	54
Figura 38 - Fluxograma do módulo das estratégias .....	54
Figura 39 - Código de carregamento de valores das estratégias .....	55
Figura 40 - Fluxograma botão Read Variables .....	56

---

Figura 41 - Excerto do código do botão Read Variables .....	58
Figura 42 - Fluxograma seleção da distribuição de tempo .....	59
Figura 43 - Excerto de código de seleção de distribuição de tempo .....	60
Figura 44 - Fluxograma alto nível do botão change .....	61
Figura 45 - Fluxograma da validação dos valores .....	62
Figura 46 - Código para gravação dos novos valores para os parâmetros.....	64
Figura 47 - Código para os campos de introdução de valores .....	64

página propositadamente em branco



## ÍNDICE DE TABELAS

Tabela 1 - Características das entidades.....	45
Tabela 2 - Características dos processos.....	47
Tabela 3 - Recursos configurados.....	48

página propositadamente em branco

## LISTAS DE SIGLAS

### Lista de Siglas

---

AB	<i>Agent Based</i>
DE	<i>Discrete Event</i>
DS	<i>Dynamical Systems</i>
ISEP	Instituto Superior de Engenharia do Porto
P.Porto	Instituto Politécnico do Porto

---

página propositadamente em branco

# 1. INTRODUÇÃO

Este capítulo refere-se à introdução ao tema do trabalho desenvolvido nesta dissertação, apresentando-se nos subtópicos seguintes o enquadramento e pertinência, quais os objetivos a atingir, as opções metodológicas, e a estrutura desta dissertação.

## 1.1. Enquadramento e pertinência

As simulações têm origem na necessidade de prever o comportamento futuro, sendo estas necessidades tão antigas como a história da humanidade. Podemos recolher da história mundial, alguns exemplos nomeadamente no campo militar onde, por exemplo, existe a necessidade de cálculos relacionados com trajetórias de balística (Williams & Williams, 1993).

A aplicação das técnicas de simulação, era utilizada sobretudo nos sectores da produção, militar e transporte (White & Ingalls, 2018), no entanto a sua utilização generalizou-se a praticamente todas as áreas do conhecimento, estando bem implementada no sector industrial, e em franco desenvolvimento no setor de serviços (Sakurada & Miyake, 2009).

Deslocando-se rapidamente o foco para os tempos atuais, consegue-se, com relativa facilidade executar simulações para qualquer área de atividade humana, graças aos avanços da tecnologia nomeadamente, decorrente do elevado poder de cálculo de dispositivos eletrónicos como *smartphones*, *tablets* e microcomputadores pessoais (Tan, Chua, Yong, & Wu, 2009) e também da evolução constante dos programas de simulação que levam muitas vezes ao limite, o poder de cálculo disponibilizado.

No âmbito deste trabalho, considera-se que o termo simulação se refere à execução num sistema computacional digital (Wagner, 2016), de um programa ou conjunto de programas em que as variáveis escolhidas apresentam os resultados aproximados aos do sistema real (caso exista) que representa.

Todo o desenvolvimento do trabalho efetuado, tem por base a aplicação das técnicas de simulação para as áreas de serviços. Constata-se que há alguns setores que como o da saúde que recorrem frequentemente à utilização de simuladores (Robinson, Radnor, Burgess, & Worthington, 2012) na vertente de instalações para tratamento de doentes. Desde há vários anos que a disposição das unidades de tratamento (salas, equipamento, etc.), são simuladas previamente por forma a se conseguir simplificar a movimentação dos utentes mantendo, no entanto, o desempenho das atividades de tratamento (Ramis, Palma, & Baesler, 2001). Deste modo evitam-se alterações posteriores com eventuais fechos para obras e respetivos custos, ou ainda perdas de clientes e subsequentes perdas de receita.

Outro exemplo ainda, são as grandes superfícies comerciais, nas quais as disposições dos expositores, produtos, acessos, caixas de pagamento, etc. são simuladas e estudadas, por forma a ser possível aos clientes a circulação no estabelecimento com uma boa fluidez, mesmo nos momentos de maior afluência (Souza, Tosoni, Canassa, & Morais, 2015).

Ainda dentro das áreas de serviços, boa parte das atividades ligadas aos setores financeiros, recorrem a simuladores como forma de prever os rendimentos futuros, sendo facilmente possível ter acesso a simuladores financeiros em qualquer *smartphone* como por exemplo um simulador de

crédito (habitação, automóvel, ou outros), ou ainda para análise de ciclos económicos (Abreu, Costa, & Brinca, 2019).

São várias as vantagens da utilização de técnicas de simulação já referidas, resumindo-se as mesmas na seguinte lista:

- Poder estudar a reorganização de áreas e circuitos de movimentação de pessoas ou objetos, sem parar serviços (por exemplo de estrangulamento no atendimento, tratamento, etc.) para teste de novas disposições (Mourtzis, Doukas, & Bernidaki, 2014);
- Identificar situações que permitem simular novos cenários que em potencial irão corrigir as situações encontradas nos sistemas reais (Ferreira, Gómez, Lourido, Quintas, & Tjahjono, 2012), traduzindo-se em poupanças de tempo, ou aumentando conforto para os clientes;
- Num curto espaço de tempo, comprimir períodos de tempo, podendo dessa forma obter uma visualização de um possível resultado futuro, ou similarmente estudar o comportamento de um sistema por um extenso período de tempo (Ferreira et al., 2012).

Não obstante as vantagens acima enumeradas, existem alguns contrapontos a considerar:

- De uma forma geral, os programas informáticos utilizados para simulação, se bem que sejam cada vez mais fáceis de programar, ainda necessitam que o seu utilizador tenha algum conhecimento, por forma ao modelo ser corretamente construído (Shannon, 1998);
- Em função do que se pretende simular, pode ser muito dispendioso a manutenção de uma equipa de consultores para a criação/manutenção do modelo (Shannon, 1998);
- Erros no modelo podem ser altamente lesivos, dado que o resultado ou resultados das simulações, podem induzir em erro os decisores (Averill M. Law, 2019);
- No caso de existir um ambiente real, é possível aferir o modelo criado, mas no caso de um ambiente que nunca existiu antes, será necessário envolver consultores, técnicos ou as pessoas com mais conhecimento na área em estudo, por forma a minimizar erros na criação do modelo (Averill M. Law, 2019).

Decorrente da análise da literatura efetuada, poucos setores de serviços para além dos referidos beneficiam das vantagens das ferramentas de simulação por forma a suportar a tomada de decisão. Considerando que o custo de processamento computacional e do software tem vindo a baixar ao longo dos anos, poderão ser os fatores desconhecimento, ou dificuldade de utilização dos programas, os maiores obstáculos á divulgação da simulação como ferramenta de apoio à decisão.

Com este trabalho, pretende-se resolver parcialmente algumas das desvantagens acima referidas, ou seja, pretende-se desenvolver um simulador que reflita o funcionamento de várias atividades simples (receção, encaminhamento, e atividade ou processo) aplicadas a uma oficina automóvel, com uma interface para o utilizador, simples e coerente, mas no entanto com a possibilidade de ser facilmente adaptável a diferentes empresas de serviços, evitando desta forma, a necessidade de recorrer a técnicos especializados para criação de novo modelo específico.

Conforme já referido o simulador criado reflete o funcionamento de uma oficina automóvel, podendo ser operado por pessoas não especializadas no tema (simulação), como por exemplo, um gestor de oficina automóvel (admitindo que não terá grandes conhecimentos de programação de modelos de simulação).

## 1.2. Objetivos de investigação

Com este trabalho pretende-se o desenvolvimento de um simulador que permita o suporte à tomada de decisão relativa ao funcionamento de uma oficina automóvel.

Por forma a se conseguir atender ao objetivo acima referido, coloca-se como objetivo geral a criação de um simulador que possa ser utilizado sem grandes conhecimentos de programação.

Partindo deste objetivo geral, será necessário atingir os vários objetivos sucintamente descritos:

- Criação de um simulador e respetivos interfaces autoexplicativos que representem as possíveis atividades ou processos da operação da oficina.
- Execução de diferentes cenários de simulação, avaliando os resultados por forma a identificar pontos de melhoria em potencial.
- Criação de animação gráfica do funcionamento de uma oficina automóvel.

O ponto de partida para este trabalho teve por base um estudo efetuado na disciplina de Gestão de Operações e Serviços, da licenciatura em Engenharia Mecânica Automóvel do Instituto Superior de Engenharia do Porto.

## 1.3. Opções metodológicas

Dado que para este trabalho, o objetivo é o simulador *'per se'*, foi aplicada a metodologia descrita nos parágrafos seguintes. Adicionalmente é feita na parte final deste subtópico, referência ao método hipotético-dedutivo, que basicamente é aplicado ao resultado do funcionamento do simulador, ou seja, criam-se hipóteses que se pensa atingirem determinado resultado, que por sua vez poderão ser validados (ou não) pelos resultados do simulador.

Numa primeira fase foi feita a revisão bibliográfica sobre o tema da simulação, constituída por várias partes como a sua evolução, metodologias, classificação de sistemas, fases de um projeto de simulação, e aplicação prática. Seguidamente numa segunda fase foram definidos os processos genéricos ao funcionamento de uma oficina automóvel, como receção, encaminhamento e execução da atividade ou processo.

O modelo lógico foi desenvolvido na terceira fase, correspondendo à base do funcionamento do simulador, sendo também desenvolvidos os vários módulos e interfaces que permitem a iteração por parte do utilizador.

Na quarta fase foram implementadas estratégias de simulação que permitem mais facilmente ao utilizador a obtenção de resultados da execução do simulador. Seguidamente, numa quinta fase foram executadas várias simulações, tendo por base as estratégias definidas, obtendo-se resultados diferentes para cada estratégia conforme esperado.

Conforme descrito no início deste subtópico, tendo por base esta ferramenta os utilizadores poderão pôr em prática o método hipotético-dedutivo. Este método combina os métodos dedutivo e indutivo e ainda de acordo com Gil (2008), o problema surge quando não existem dados para explicar um determinado fenómeno, então para explicar o problema formulam-se hipóteses que por sua vez deverão ser criticadas “com vista à eliminação do erro” segundo (Marconi & Lakatos, 2003, p. 95), dando no entanto origem a novos problemas. Na Figura 1, é esquematizado o método hipotético dedutivo.

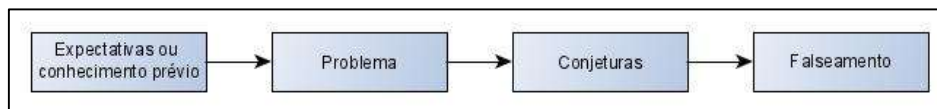


Figura 1 - Método hipotético-dedutivo (Marconi & Lakatos, 2003)<sup>1</sup>

Em termos práticos, o problema surge de conhecimentos ou teorias já pré-existentes, dando origem a novas teorias (no esquema apresentado como “conjeturas”), sendo que estas últimas são sujeitas a testes. De acordo com Marconi e Lakatos, “Se a hipótese não supera os testes, estará falseada, refutada, e exige nova reformulação do problema e da hipótese, que, se superar os testes rigorosos, estará corroborada, confirmada provisoriamente, não definitivamente como querem os indutivistas” (2003, p. 96).

Partindo então do método hipotético-dedutivo, a abordagem à investigação será quantitativa, dado que se vai estudar a relação entre as variáveis dependentes e a variável independente. Desta forma será utilizado o plano verdadeiramente experimental que é definido por três aspetos segundo Almeida e Freire (2007), primeiramente assegurar que é possível manipular a variável independente e fixar os seus valores, em segundo lugar alterar a variável independente por forma a contrastar os efeitos nas variáveis dependentes. Segundo Gil (2008), serão necessários dois grupos, um de controlo e um segundo experimental. Ainda de acordo com Almeida *et al.* (2007, p. 95) “O que se pretende é que a variância dos resultados na variável dependente seja exclusivamente, ou o mais possível, associada aos valores assumidos pela variável independente”.

Os simuladores prestam-se especialmente a este trabalho, pois todas as variáveis são conhecidas e controladas, sendo possível alterar apenas uma das variáveis, e observar os efeitos dessa alteração nos resultados obtidos.

#### 1.4. Estrutura do trabalho

O tema fundamental deste trabalho é o desenvolvimento de um simulador que suporte a tomada de decisão relativo ao funcionamento de uma oficina automóvel, organizando-se esta dissertação da seguinte forma.

No primeiro capítulo, “Introdução”, procurou-se introduzir o tema da dissertação fazendo o seu enquadramento no contexto atual aproveitando para realçar a sua pertinência, quais os objetivos que se pretendem atingir, as opções metodológicas adotadas e a própria estrutura da dissertação.

No segundo capítulo, “Revisão Bibliográfica”, observa-se a evolução da área da simulação, descrevem-se várias metodologias de simulação, assim como a classificação sobre o funcionamento de sistemas para os quais se pretendam criar modelos de simulação. Adicionalmente referem-se ainda as vantagens e desvantagens decorrentes da utilização da simulação, e as respetivas fases de um projeto de simulação. Faz-se ainda uma breve referência à utilização da simulação por sectores, e apresentam-se também algumas utilizações da simulação em sectores específicos.

No capítulo três, “Sistema de Apoio à Decisão Desenvolvido”, faz-se uma breve descrição do funcionamento genérico de uma oficina automóvel, descrevendo os seus processos mais básicos. Aborda-se o sistema desenvolvido, sendo apresentados os vários módulos e as respetivas variáveis que os compõem. Também neste capítulo é apresentada a forma de lançar a execução do simulador

<sup>1</sup> Fluxograma criado com o *software* yEd Version 3.19.1.1

e a consequente animação gráfica. Adicionalmente apresenta-se o modelo lógico do simulador, nos seus vários componentes e a descrição dos mesmos, e por último descreve-se um pouco mais os pontos chave da interface de utilizador desenvolvido.

No quarto capítulo, “Conclusões e Trabalho Futuro” apresenta-se o resultado deste trabalho, as conclusões, o contributo inovador do mesmo, limitações e também propostas de trabalho a desenvolver futuramente.

Na parte final deste trabalho, são apresentados as referências bibliográficas e o apêndice com o código desenvolvido.



## 2. REVISÃO BIBLIOGRÁFICA

Nos subtópicos seguintes, vão-se introduzir alguns conceitos base sobre simulação, ainda que de forma muito sucinta. No subtópico 2.1 constata-se a evolução das técnicas de simulação em termos globais. No subtópico 2.2 apresentam-se quatro métodos de simulação sendo um pouco mais extensa no último apresentado (eventos discretos), dado que o desenvolvimento deste trabalho terá por base esse método. O subtópico 2.3 aborda a classificação dos sistemas de forma genérica. Já o subtópico 2.4 descreve de forma um pouco mais alargada (relativamente ao já apresentado), as vantagens da simulação, e também algumas desvantagens. Seguidamente no subtópico 2.5 apresentam-se as fases de um projeto de implementação de técnicas de modelação/simulação. O subtópico 2.6 é referente a publicações e estudos das técnicas de simulação a áreas de setores de serviços.

### 2.1. Evolução da utilização da simulação

De uma forma global denota-se o crescimento da utilização de técnicas de simulação computacional para todas as áreas de atividade humana. Segundo Mourtziz *et al.* (2014) a quantidade de artigos sobre simulação têm crescido exponencialmente desde os anos 70, conforme se pode constatar na Figura 2.

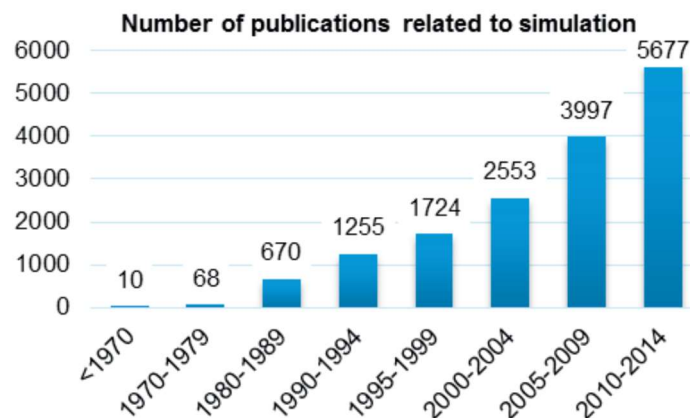


Figura 2 - Artigos publicados por década (Mourtziz et al., 2014)

Não obstante a evolução apresentada sobre as técnicas de simulação, também se denota crescimento a nível mundial do setor de serviços, cujos sistemas são também bastante complexos. Decorre do exposto que estes fatores têm contribuído para estudos e pesquisas da aplicação da simulação no setor de serviços estando no entanto, menos desenvolvidos do que para outros setores como por exemplo a produção e a logística (Sakurada & Miyake, 2009).

### 2.2. Metodologias de simulação

Existem várias abordagens possíveis relativamente ao estudo de um sistema, sendo representado na Figura 3, uma possível organização.

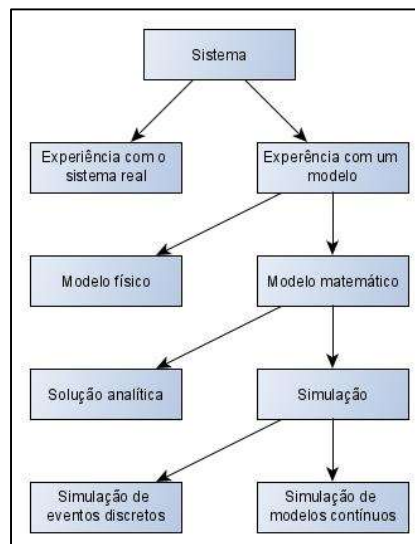


Figura 3 - Métodos para estudo de sistemas, adaptado de Law e Kelton (1991) <sup>2</sup>

De acordo com Banks, a simulação é “... a imitação da operação de um processo ou sistema real ao longo do tempo” (2000, p. 1), sendo possível inferir que a realização de experiências sobre sistemas ou processos reais poderão não ser consideradas simulações, isto é, são de facto experimentos. No entanto, a partir de um modelo (físico ou matemático) que represente um sistema ou processo real, é possível analisar e descrever o comportamento de um sistema e eventualmente colocar questões do que aconteceria em determinada circunstância.

De acordo com a Figura 3, a modelação associada a modelo físico pode-se explicar facilmente com o exemplo do túnel de vento (White & Ingalls, 2018). Estes tipos de simuladores são utilizados para testes de aerodinâmicas de peças, tanto para indústria aeronáutica como automóvel, e ainda outros sistemas.

Já no campo da modelação matemática, existem vários métodos que se podem utilizar. Se considerarmos o caso das soluções analíticas, então o sistema a representar deverá ser muito simples, para permitir a criação de um modelo matemático que represente o mesmo ou seja passível de serem utilizadas técnicas de programação linear inteira (DOS SANTOS, 2010). Em função dos valores de entrada corresponderem aos cenários a simular, obter-se-ão as soluções que correspondem à resposta do sistema naquelas circunstâncias.

No entanto, de uma forma geral os sistemas do mundo real (existentes ou a criar) são normalmente bastantes complexos e necessitam de métodos matemáticos mais poderosos que permitam chegar às soluções a analisar. Para este tipo de simulações, existem métodos mais aplicados a sistemas contínuos, como os Sistemas Dinâmicos (*Dynamic Systems* - DS) e a Dinâmica de Sistemas (*Systems Dynamics* - SD), ou então métodos de eventos discretos (*Discrete Event* - DE) e baseada em agentes (*Agent Based* - AB) (Borshchev & Filippov, 2004).

### 2.2.1. Sistemas dinâmicos

Os sistemas dinâmicos são sistemas que representam matematicamente o mundo físico (Borshchev & Filippov, 2004), e por isso as suas variáveis são as grandezas físicas em estudo. Na Figura 4,

<sup>2</sup> Fluxograma criado com o *software* yEd Version 3.19.1.1

representa-se, a título de exemplo, um diagrama de blocos de uma bola saltitante, sendo que em cada bloco temos a relação matemática entre as grandezas, aceleração (gravidade na figura 4), velocidade e posição. Com este modelo é possível calcular (ou prever) toda a trajetória da bola.

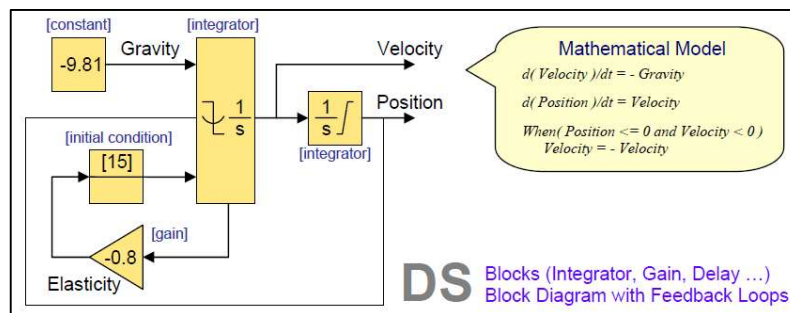


Figura 4 - Representação matemática de um sistema "bola saltitante" (Borshchev & Filippov, 2004)

## 2.2.2. Dinâmica de sistemas

Já na dinâmica de sistemas são estudadas as características da realimentação da informação, que podem amplificar ou atrasar as interações, influenciando o sucesso ou insucesso do sistema. Este tipo de modelo tem bastante aplicação em sistemas sociais, ecológicos, entre outros (ver Figura 5). Para ser possível representar o sistema, o nível de abstração aumenta necessariamente refletindo uma visão agregada de políticas. De acordo com Borshchev (2004), a modelação de um sistema em dinâmica de sistemas, passa pela descrição do comportamento do mesmo através da iteração de vários circuitos de realimentação, estejam os mesmos a equilibrar ou reforçar o sistema e também estruturas que introduzem atrasos.

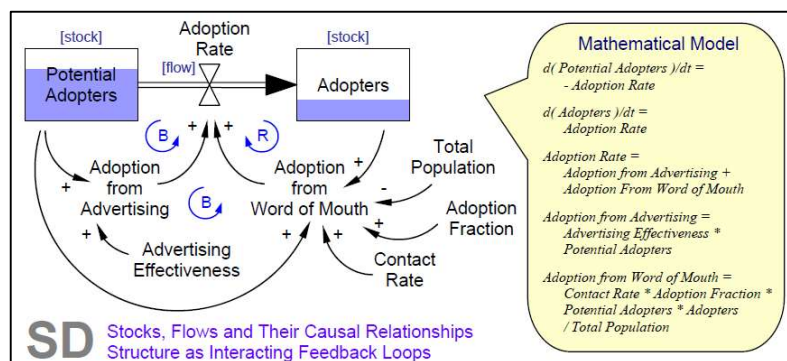


Figura 5 - Representação de um sistema como dinâmica de sistemas. (Borshchev & Filippov, 2004)

## 2.2.3. Baseada em agentes

Já na simulação baseada em agentes, cada entidade tem o seu comportamento definido a nível individual, tendo a capacidade de comunicar interagindo com outros agentes ou com o ambiente de acordo com as suas regras de comportamento. O comportamento global do sistema, é o resultado do comportamento global que resulta das interações de dos agentes que podem ir das dezenas aos milhões (Borshchev & Filippov, 2004).

Na Figura 6 representa-se uma arquitetura genérica de um modelo baseado em agentes. Neste modelo, os seres humanos simulados são modelados como agentes, interagindo com outros agentes (seres humanos) e também com o meio ambiente. O meio ambiente tem um papel crucial

neste tipo de modelação que deve ser tido em conta, pois pode influenciar todo o comportamento da simulação. A mesma Figura 6, representa um sistema que modela a difusão de um produto no mercado.

É de realçar que nesta forma de representação de sistemas, os resultados obtidos não revelam indivíduos ou entidades, sendo desta forma considerado que o comportamento é igual para toda uma população. Desta forma o modelador necessita de estudar e compreender as dependências estruturais do modelo e a qualidade dos dados para as mesmas.

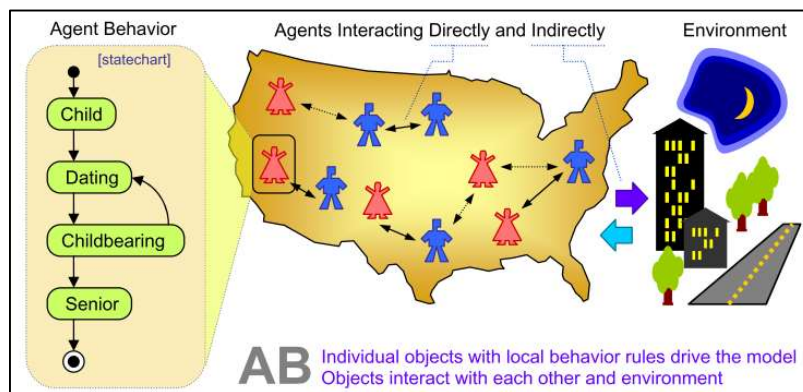


Figura 6 - Modelo genérico de modelação baseada em agentes (Borshchev & Filippov, 2004)

#### 2.2.4. Eventos discretos

Relativamente a eventos discretos, um sistema que seja representado por esta metodologia, é um sistema cujo estado vai mudando em momentos discretos no tempo, ou seja, não há variação contínua do estado das variáveis. O tempo neste tipo de modelação é um componente crítico, pois os estados do sistema mudam apenas cada vez que há um evento de tempo (Carson, 2005).

Neste tipo de modelação, existem as chamadas entidades (contendo cada uma o seu conjunto de propriedades ou atributos) que representam um indivíduo ou peça individual que se vão deslocando ao longo da simulação, sendo afetadas por outras entidades ou recursos (White & Ingalls, 2018).

Os chamados recursos são os elementos no modelo que prestam um serviço, isto é, podem ser uma máquina que executa uma transformação no produto, ou um balcão de atendimento que atende o indivíduo (White & Ingalls, 2018).

Adicionalmente existem também as filas de espera, surgindo sempre que um conjunto de entidades sofrem condições idênticas. Na Figura 7 pode-se, a título de exemplo, visualizar um sistema que representa de um balcão de atendimento de um banco.

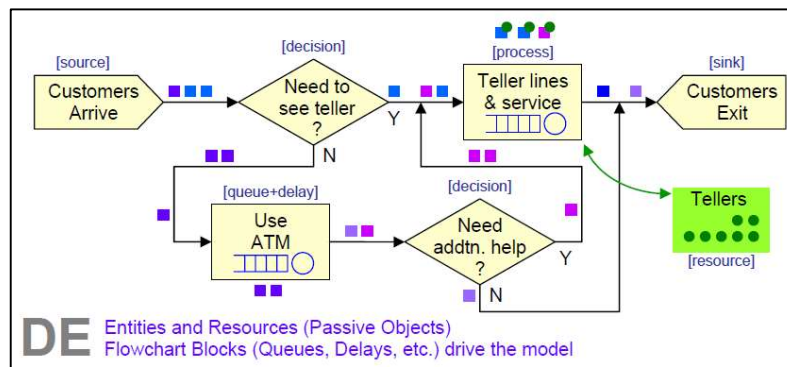


Figura 7 - Representação de balcão de atendimento bancário (Borshchev & Filippov, 2004)

## 2.3. Classificação de sistemas

Relativamente à classificação de sistemas, os modelos de simulação podem ser classificados de várias formas, apresentando-se um esquema adaptado de Rossetti (2016) na Figura 8. Seguidamente explicam-se as classificações apresentadas.

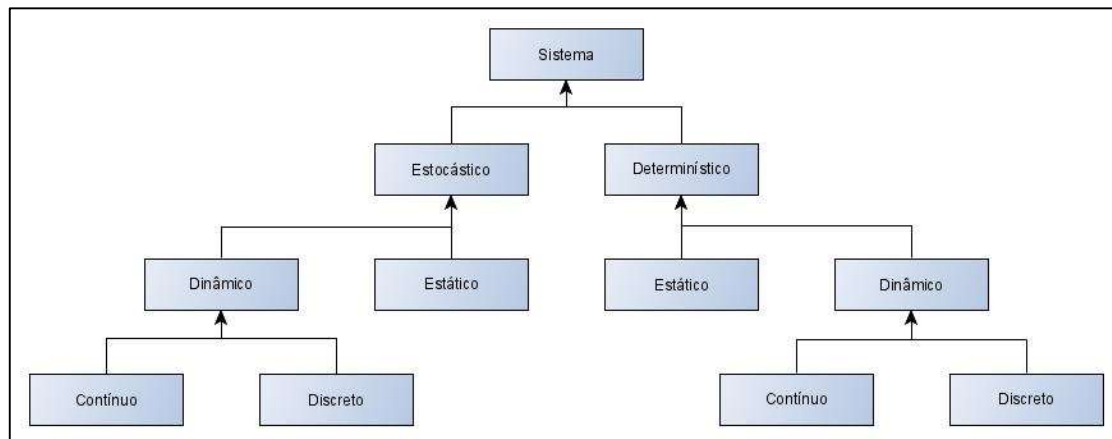


Figura 8 - Esquema de classificações adaptado (Rossetti, 2016)<sup>3</sup>

### 2.3.1. Determinístico/Estocástico

Um modelo determinístico, é um modelo no qual os valores parametrizados nas entradas são constantes. Adicionalmente, todas as relações são exatas não existindo aleatoriedade. Por outro lado, num modelo estocástico algumas das variáveis têm alguma incerteza, como por exemplo num banco, cuja chegada clientes é aleatória e adicionalmente o tempo de serviço pode variar (Kelton, Sadowski, & Zupick, 2015).

Por forma a modelar com o maior realismo possível, um modelo pode ter entradas determinísticas nalguns componentes e estocásticas noutro componentes, conseguindo-se desta forma introduzir alguma incerteza que usualmente está presente no mundo real. De acordo com Kelton (2015), um modelo estocástico tem sempre incerteza nos resultados, obrigando o modelador a considerar cuidadosamente o desenho e a interpretação de cada execução da simulação.

<sup>3</sup> Fluxograma criado com o *software* yEd Version 3.19.1.1

### 2.3.2. Estático/Dinâmico

Para os modelos estáticos, o fator tempo não é considerado, sendo possível utilizar como exemplo os modelos que fazem uso da programação linear. Nestes modelos os resultados podem ser obtidos por técnicas analíticas, sendo até possível nalgumas situações, atingir a solução com uma folha de cálculo (Kelton et al., 2015). No entanto para a maior parte dos sistemas do mundo real, é necessário considerar o fator tempo, e nesse caso o modelo será dinâmico pois este fator influencia o comportamento do modelo ao longo da simulação. Num modelo de simulação dinâmico, o tempo é a principal variável independente, sendo as outras variáveis funções do tempo, logo variáveis dependentes.

### 2.3.3. Discreto/Contínuo

Os modelos dinâmicos são classificados de acordo com os instantes de tempo em que as variáveis de estado têm os seus valores alterados. Considera-se então que num modelo discreto, o estado das variáveis dependentes variam em pontos específicos do tempo simulado (tempo de evento), como por exemplo num sistema de produção em que as peças entram e saem dos sistema em momentos pré-determinados (Kelton et al., 2015). Por outro lado, é considerado um modelo contínuo aquele que as variáveis dependentes se alteram continuamente ao longo do tempo simulado, como por exemplo um reservatório de água em que a mesma está sempre a entrar e a sair, conforme ocorre precipitação e evaporação (Kelton et al., 2015).

## 2.4. Vantagens e desvantagens da simulação

De acordo com Banks (2000), o número de empresas a recorrer às ferramentas de modelação e simulação tem vindo a crescer de forma constante. Estes crescimentos têm sido impulsionados pelo aumento das capacidades dos computadores atuais e dos programas de simulação.

Existem várias vantagens na utilização da simulação como forma de melhorar ou prever comportamentos futuros, no entanto, existem também algumas desvantagens, que se não forem corretamente endereçadas, podem provocar prejuízos ou mesmo perdas.

Seguidamente apresentam-se um conjunto de vantagens sobre a utilização da simulação:

- Refere Banks (2000), que recorrendo à simulação é possível estudar o resultado de alterações ou adições sem a necessidade de comprometer recursos para efetuar as mesmas, como por exemplo, obras para uma nova máquina, ou um novo posto de atendimento.
- Numa simulação, é possível controlar o tempo, ou seja, podemos nalguns minutos executar uma simulação de anos, ou podemos fazer precisamente o contrário, demorar horas a executar uma simulação de um sistema que ocorre em minutos, por forma a estudá-la mais pormenorizadamente (Shannon, 1998) (Banks, 2000) (Carson, 2005), como por exemplo, para descobrir porque é que determinado fenómeno acontece no mundo real. Nestes casos pode-se reconstruir o cenário numa simulação e analisar o que acontece passo a passo (Shannon, 1998) (Banks, 2000).

- Dentro do mesmo registo, a simulação permite identificar restrições e diagnosticar problemas, como por exemplo estrangulamentos em linhas de produção (Shannon, 1998) (Ferreira et al., 2012), que normalmente são efeitos resultantes de causas que a simulação ajuda a descobrir (Banks, 2000).
- Uma das maiores vantagens da simulação, assumindo que o modelo se encontra devidamente validado, é a possibilidade de explorar novos processos ou métodos operacionais, ou ainda novas políticas, evitando disrupções no funcionamento do sistema (Shannon, 1998) (Banks, 2000) (Carson, 2005).
- Uma simulação de um sistema, permite visualizar como o sistema funciona de facto e perceber quais as variáveis que mais afetam o desempenho (Shannon, 1998), adicionalmente com a utilização das ferramentas de animação que os softwares de simulação atualmente dispõe, mais facilmente ficam expostas falhas que de outra forma não seriam detetadas (Banks, 2000).
- O comportamento de um sistema pode ser estudado tendo vista alguns cenários futuros, do género “qual é a resposta do sistema se a procura aumentar 10%”, e nesse sentido já ter alterações estudadas e programadas, caso o cenário aconteça (Shannon, 1998) (Banks, 2000). Este tipo de simulação também ajuda a definir as especificações necessárias para as ditas alterações (Carson, 2005).
- Conforme refere Banks, o custo típico de um estudo recorrendo a simulação é substancialmente menos de 1% do custo de implementação de um desenho ou redesenho. Por contrapartida, o custo de uma alteração após implementação pode ser tão grande que a simulação pode ser vista como um bom investimento (Banks, 2000) (Carson, 2005).

Não obstante as vantagens possíveis de retirar de modelação e simulação de sistemas, na realidade existem também algumas desvantagens aqui apresentadas:

- De uma forma geral, a modelação e respetiva simulação necessitam de conhecimentos especializados por parte do modelador (Shannon, 1998) (Banks, 2000).
- Adicionalmente o modelador deverá ter um grande conhecimento sobre o sistema que esta a modelar (Shannon, 1998), sendo mais habitual que seja uma equipa multidisciplinar a estudar o sistema e a criar o modelo (Carson, 2005). A manutenção de uma equipa deste género, pode ser algo dispendioso.
- De acordo com Banks (2000), os resultados de uma simulação podem ser de difícil interpretação, assumindo que os modelos são estocásticos por forma a serem mais similares ao mundo real.
- Se bem que as ferramentas de simulação, de forma geral já estão dotadas de capacidade de análise dos resultados, permitindo uma análise extensiva dos dados, na realidade o operador tem de compreender o processo de análise (Carson, 2005).
- Por vezes é difícil obter dados confiáveis para introduzir no modelo, sendo que o mesmo não consegue compensar esses dados, e consequentemente os resultados não terão qualquer interesse prático (Shannon, 1998).
- A definição inicial do problema é fundamental para evitar em retrocessos no projeto de modelação. Acontece por vezes, que mesmo quem está a solicitar a modelação e simulação, não tem bem claro o que pretende obter com a mesma, obrigando a redesenhar partes do modelo (Carson, 2005).

## 2.5. Fases de um projeto de simulação

Existem várias formas de abordar um projeto de simulação, existindo na literatura da especialidade diferentes propostas de abordagem diferindo na quantidade de etapas e objetivos de cada uma. Na Figura 9 seguinte representa-se a abordagem mais vulgarmente utilizada (seguida neste trabalho), constituída por sete etapas (A M Law, 2008). Seguidamente descrevem-se sucintamente cada uma das etapas.

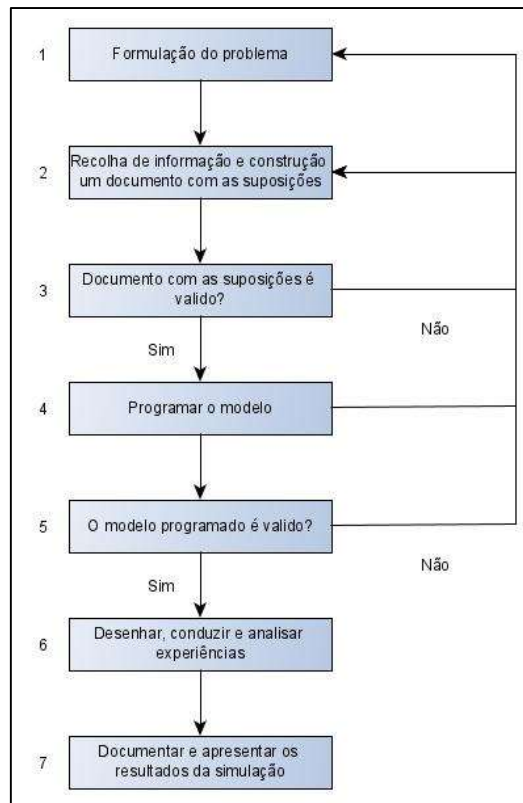


Figura 9 - Etapas de um projeto de simulação (A M Law, 2008) <sup>4</sup>

### 2.5.1. Primeira etapa – formulação do problema

Nesta etapa, relativa a formulação do problema, é o momento de arranque do projeto, onde o problema poderá ser apresentado de forma superficial, ou sem termos quantitativos. Dada a importância desta etapa torna-se necessário colocar algumas questões (A M Law, 2008) (Shannon, 1998), como por exemplo:

- Qual é o objetivo geral do estudo a efetuar?
- Qual o âmbito do modelo a criar?
- Quais as questões a responder ou a decisão a tomar?
- Qual a informação necessária para tomar a decisão?
- Quais os critérios precisos para a tomada da decisão?
- Quais os sistemas e configurações a modelar?
- Qual é o prazo para o desenvolvimento do estudo e quais os recursos necessários?

<sup>4</sup> Fluxograma criado com o *software* yEd Version 3.19.1.1

### 2.5.2. Segunda etapa – recolha de informação

Para a recolha de dados e criação de um documento com as suposições ou premissas do modelo a criar, deverão ser efetuados alguns passos (A M Law, 2008) (Carson, 2005) (Banks, 2000) (Shannon, 1998).

- Deverão ser recolhidas informações dos procedimentos operacionais e do *layout* do sistema.
- Recolha de dados específicos para a parametrização da modelação, como por exemplo distribuições de probabilidade.
- Os dados deverão ser quantitativos e não resumos ou compilações de dados antigos.
- As suposições sobre o modelo, algoritmos e os resumos de dados deverão ficar escritos, no entanto, o seu detalhe depende de alguns fatores como: objetivos do projeto, disponibilidade de dados, prazo para o projeto disponibilidade financeira entre outros.
- O documento a criar deverá ser isento de termos técnicos, por forma a ser de fácil compreensão por toda a equipa e principais interessados no estudo.

### 2.5.3. Etapas três, quatro e cinco

Na etapa três – validação do documento com as suposições, o qual deverá ser totalmente percorrido por toda a equipa que está a trabalhar sobre o estudo, assim como, por quem solicitou a utilização da simulação (A M Law, 2008). Na eventualidade de se descobrirem algumas falhas, retorna-se então ao ponto anterior para as corrigir. Só se deve avançar para a etapa seguinte após todos estarem de acordo com o documento.

Relativamente à quarta etapa, correspondente à programação do modelo, é momento em que de facto se passa à programação do modelo, podendo ser numa linguagem de programação generalista (ex.: C, C++, etc.) ou então recorre-se a um *software* de simulação comercial (ex.: Arena, AnyLogic, etc.). Segundo Law (2008), nesta etapa, também é feita a verificação do simulador, ou seja, a verificação do correto funcionamento do código (ausência de *bugs*).

A validação do modelo é efetuada na quinta etapa, sendo um passos mais importantes no projeto de simulação, pois de acordo com Banks (2000) é a determinação do modelo conceptual que representa com precisão o sistema real. Ainda de acordo com o mesmo autor, se houver um sistema real, então os resultados do mesmo deverão ser utilizados para comparação com os obtidos na simulação. Para Law (2008) , mesmo que exista o sistema real e os dados possam ser comparados, os analistas da equipa e especialistas em simulação deverão rever os resultados para verificar a razoabilidade dos mesmos. Adicionalmente, Carson (2005) refere que sendo possível deverão ser usadas animações e outras ferramentas visuais, que permitam visualizar as suposições já definidas.

### 2.5.4. Sexta etapa – experimentação

O modelo já deverá estar devidamente validado e, portanto, pronto a simular um sistema real, ou então a simular o comportamento do sistema sob outras condições que não as já conhecidas. Conforme referem vários autores (A M Law, 2008) (Carson, 2005) (Banks, 2000) e (Shannon, 1998),

nesta fase existem ainda algumas questões a definir, que poderão até já ter sido endereçadas no documento das suposições definido na etapa dois. As questões mais comuns são:

- O tempo de execução da simulação.
- O período de aquecimento (se aplicável).
- Quantidade de replicações.
- Parâmetros de inicialização e de finalização.

Ainda nesta etapa deve ser considerada a análise dos resultados obtidos, e conseqüentemente a necessidade de efetuar novas experiências (A M Law, 2008), ou seja, executar novamente o simulador para obtenção de novos resultados e validar potenciais melhorias a introduzir no sistema real, seja por redução de estrangulamentos, ou filas de espera, aumento de recursos, etc.

### **2.5.5. Sétima etapa – apresentação de resultados**

A última etapa de um projeto de implementação de simulação, trata da documentação do projeto e apresentação dos resultados da simulação. Segundo Law (2008), na documentação final deverão constar, o documento com as suposições iniciais, a descrição do programa criado, os dados do estudo da simulação, sejam os iniciais ou os alterados para testes de hipóteses, assim como os resultados obtidos. Para um projeto sobre um estudo de um sistema recorrendo à simulação ser considerado completo, os seus resultados têm que ser compreendidos, aceites e também utilizados (Shannon, 1998).

Também deverão fazer parte da apresentação dos resultados, as animações computadorizadas do sistema em estudo no simulador, pois as mesmas permitem transmitir mais facilmente e de forma mais credível ao decisor a informação dos resultados obtidos (Johnson & Poorte, 1988). Adicionalmente a representação animada do sistema, permite visualizar o fluxo das entidades no sistema, filas de espera, alterações de estado e outros dados, que aliás devem ser utilizadas ao longo do desenvolvimento do simulador e estudo de novos cenários.

## **2.6. Estudos recorrendo à simulação em setores de serviços**

De acordo com Jahangirian, constata-se que existe uma tendência de diversificação das áreas de atividade, relativamente à utilização das técnicas de simulação (Jahangirian, Eldabi, Naseer, Stergioulas, & Young, 2010). Não obstante a diversificação, conforme se pode constatar na Figura 10, o setor da produção é o que mais se destaca na utilização de técnicas de simulação.

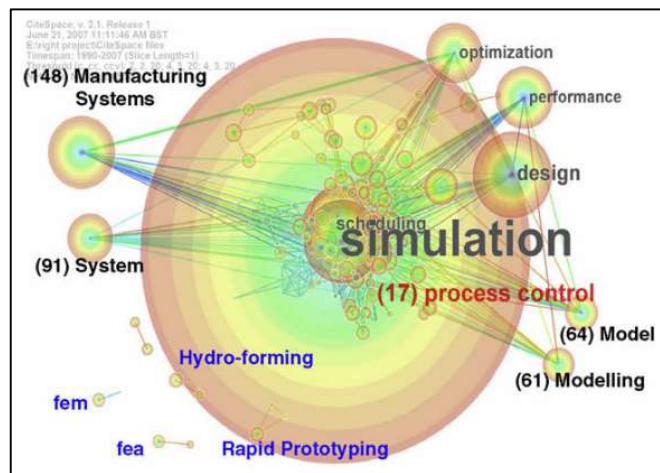


Figura 10 - Utilização de técnicas de simulação (Jahangirian et al., 2010)

Na secção seguinte apresentam-se algumas utilizações de técnicas de simulação para alguns setores.

### 2.6.1. Setor logística – otimização de rotas

No trabalho desenvolvido por Castañé et al. (2019), a simulação é utilizada pelos decisores como ferramenta que suporta a decisão relativa à geração de planos de ação diários, aplicados a técnicos que efetuam intervenções em equipamentos geograficamente dispersos. A ferramenta, pode considerar para um técnico, os serviços de manutenção de equipamento que são planeados no tempo, mas também as situações de avarias sendo que estas podem ocorrer em qualquer altura, e logo obrigando à interrupção do trabalho planeado. No artigo apresentado, é possível observar para os cenários criados (adicionando ou retirando técnicos) e as situações de trabalho não planeado e efetuar comparações com o que de facto se passa no trabalho do dia a dia.

### 2.6.2. Setor produção – problema de performance

No estudo seguidamente referido, é efetuado um estudo sobre uma rede fechada de montagem automóvel (Ferreira et al., 2012). Nesta publicação são referidos quatro circuitos em malha fechada em que as correias de transporte funcionam como armazenagem temporária entre postos. Da observação do funcionamento foi constatado que qualquer bloqueio da linha, produz efeitos que se propagam tanto para montante como para jusante do ponto de bloqueio. Com o estudo efetuado, foi verificado que através da variação do comprimento das correias de transporte (que funcionam como *buffers*) é possível melhorar a performance de todo o sistema.

### 2.6.3. Setor saúde – problema de capacidade

A utilização de técnicas de simulação no setor da saúde, é uma ferramenta valiosa na tomada de decisões em serviços de saúde (Silva, Caulliriaux, & Spiegel, 2015). Segundo os autores, o setor de

serviços (no qual a saúde está inserida) têm algumas particularidades que necessitam de alguma atenção, nomeadamente:

- Os serviços são intangíveis, ou seja, não são passíveis de armazenamento. Uma hora de técnico ou especialista desperdiçada, dificilmente será recuperada.
- O fornecimento do serviço, requer simultaneamente o cliente (neste caso paciente) e o fornecedor do serviço (médico, enfermeiro, técnico, etc.).
- A inexistência de dois serviços iguais.
- A medição da qualidade de serviço é extremamente difícil, pois é dependente do cliente e logo altamente subjetiva.

O problema proposto aos autores, estava relacionado com o elevado grau de absentismo dos pacientes, as longas filas de espera para marcações de consultas e a elevada quantidade (cerca de 80%) de pacientes caracterizados como “pouco urgente” ou “não urgente”.

Após a criação do modelo, levantamento dos dados de funcionamento do sistema em estudo, e validação do mesmo, a equipa estudou vários cenários alternativos, sugerindo um deles a nível de distribuição de médicos, que reduziu o tempo de espera dos pacientes, como o mais vantajoso para implementação.

Referindo outro trabalho ainda na mesma área, no estudo seguidamente resumido, a simulação foi aplicada para efeitos de dimensionamento de capacidade num centro de cirurgia ambulatório (Ramis et al., 2001). Neste caso foi efetuado a recolha de dados no local em termos de tempos para várias atividades, sendo os mesmos validados pelos profissionais do centro. Após a construção e validação do modelo, foram executados cenários alternativos, em que se constata que com o mesmo de número de quartos/camas, mas dando-lhes funções específicas é possível aumentar a quantidade de pacientes atendidos por dia.

#### **2.6.4. Setor comercial – caixas pagamento em supermercado**

No caso seguinte, a simulação é utilizada para ajustar o funcionamento de caixas de supermercado (Souza et al., 2015), especialmente nos momentos de maior fluxo. Adicionalmente, no decurso do estudo efetuado para a criação do modelo, constatou-se que no local existia uma caixa de pagamento que se encontrava pouco visível, e conseqüentemente os clientes continuavam a esperar noutra fila enquanto a caixa de pagamento em causa estava vazia.

Os casos apresentados, demonstram o potencial das técnicas computacionais de simulação em responder às situações de otimização. No primeiro caso a distribuição dos recursos (médicos) pelos serviços, permitiu diminuir o tempo de espera dos pacientes. No segundo caso o aumento da quantidade de cirurgias diárias, e o último caso que logo com o levantamento para o estudo permitiu identificar a situação da falta de visibilidade de um caixa, e os resultados do modelo em virtude da redistribuição das caixas de atendimento.

### 3. SISTEMA DE APOIO À DECISÃO DESENVOLVIDO

Nesta secção é descrito no subtópico 3.1 o funcionamento de uma oficina automóvel genérica, apresentando-se no subtópico 3.2 os módulos do simulador e também o seu funcionamento. No subtópico 3.3 aborda-se a animação da simulação ficando para o subtópico 3.4 as componentes de modelação lógica do funcionamento do simulador e por último, no subtópico 3.5, excertos principais do código desenvolvido para a interface com o utilizador.

#### 3.1. Descrição de funcionamento de oficina genérica

De uma forma geral, é possível decompor as atividades principais do funcionamento de uma oficina em alguns blocos básicos, constatando-se também que os mesmos serão similares entre vários setores de atividade. Na Figura 11 os retângulos representam algumas atividades ou processos, que como será expectável, necessitam de algum tempo para a sua execução, podendo este tempo variar um pouco, ou seja, não será um tempo constante.

Atualmente, muitas oficinas funcionam com marcação prévia, traduzindo-se na programação dos trabalhos previstos com alguma antecedência assim como o agendamento da entrada da viatura na oficina, não obstante, há sempre necessidade de efetuar a receção da mesma e validação dos trabalhos a executar, sendo esta atividade representada no retângulo – Receção Viatura.

Seguidamente a viatura será encaminhada para o conjunto de postos que irá desempenhar o serviço previsto, sendo considerados os serviços de troca de óleo, reparações gerais e serviços rápidos, representados nos respetivos retângulos. Após a execução do serviço, a viatura seguirá para entrega ao cliente.

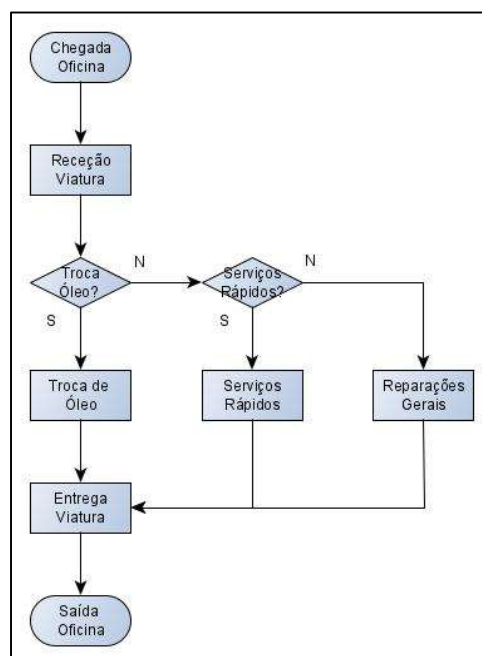


Figura 11 - Fluxograma simples funcionamento oficina automóvel<sup>5</sup>

<sup>5</sup> Fluxograma criado com o *software* yEd Version 3.19.1.1

Não foi representado neste trabalho a área da reparação de viaturas decorrente de colisões, pois usualmente envolve companhias de seguros e respetivas peritagens, que são fatores externos e não controláveis. Eventualmente trabalhos de chapa e pintura serão considerados nas reparações gerais, após ser tratada a parte burocrática entre as várias entidades. Como informação adicional é de referir que no âmbito deste trabalho, o simulador é estocástico, pois os dados de entrada os serviços a prestar são representados por funções com distribuição normal ou outras mais adequadas. Complementarmente o objetivo é representar o melhor possível o mundo real, logo, o fator tempo estará presente implicando que o modelo é dinâmico, e como por sua vez, as mudanças das variáveis ocorrem em momentos discretos, é um modelo de eventos discretos, criado no *software* Arena.

### **3.2. Módulos e funcionamento do sistema de apoio à decisão**

No desenvolvimento desta ferramenta de simulação de apoio à decisão, foi dada muita atenção à coerência da interface do utilizador, nomeadamente os módulos, localização de botões e apresentação em geral, por forma ao seu funcionamento ser similar em qualquer ponto, facilitando dessa forma a sua utilização.

O simulador é desenvolvido sobre o *software* Arena, tendo sido criado uma interface com o utilizador onde são apresentados os parâmetros possíveis de alteração para a criação de cenários. Nesta interface, estão já integradas funcionalidades que detetam erros ou omissões de valores nas caixas de diálogo que vão sendo apresentadas em função das opções tomadas pelo utilizador.

Na figura seguinte apresenta-se o fluxograma global da interface do utilizador. Este é um fluxograma de alto nível, que faz referência aos módulos desenvolvidos. O desenvolvimento dos módulos é abordado na subsecção 3.5.

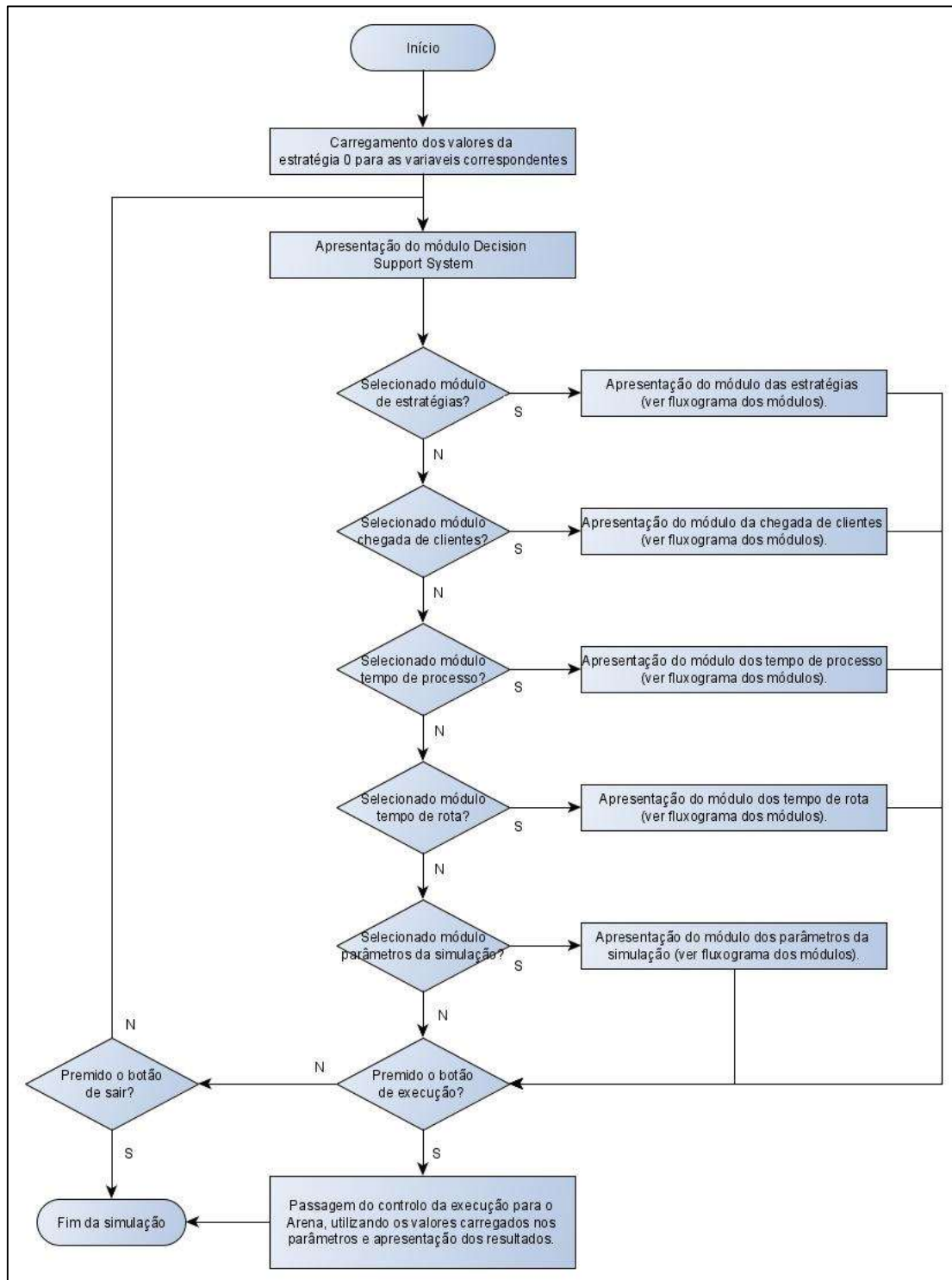


Figura 12 - Fluxograma global<sup>6</sup>

Desta forma o simulador é constituído por vários módulos, descrevendo-se nos subtópicos seguintes cada um deles.

<sup>6</sup> Fluxograma criado com o *software* yEd Version 3.19.1.1

### 3.2.1. Módulo inicial

Na Figura 13, apresenta-se o ecrã inicial, onde é apresentado um aspeto gráfico de uma potencial oficina automóvel, uma descrição da forma de lançar a execução de uma simulação na caixa *Running a Simulation*, e mais alguma informação acerca do objetivo do simulador e do seu desenvolvimento.

De salientar que é neste módulo que se encontram os botões de saída do simulador e também de execução de uma simulação.

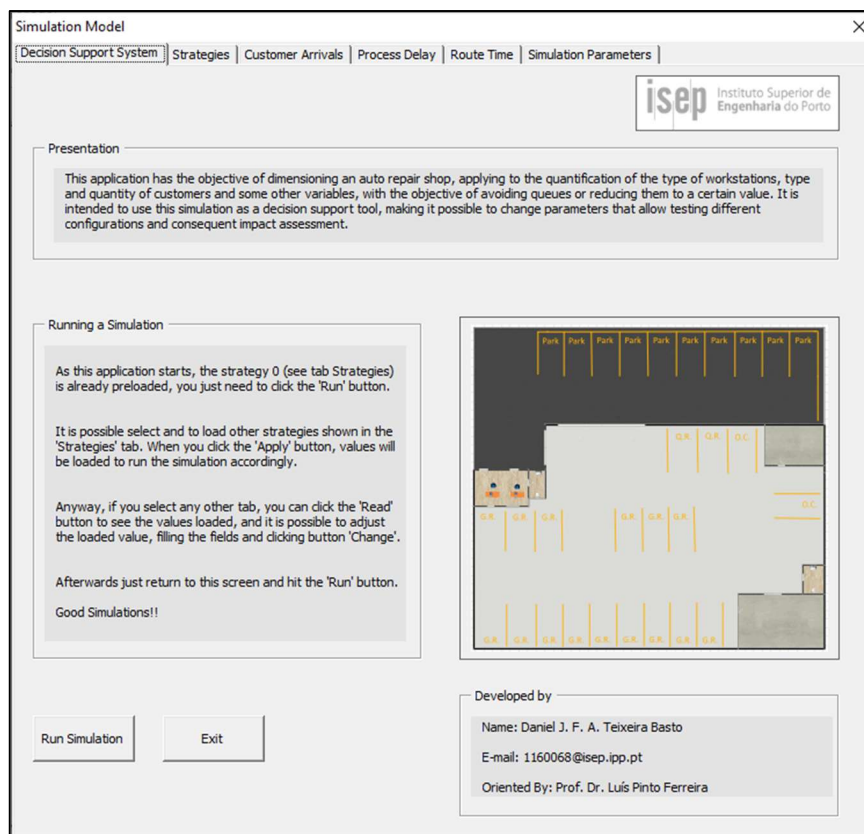


Figura 13 - Módulo inicial do simulador

Conforme já referido o simulador é constituído por vários módulos como, as estratégias já pré-programadas (*Strategies*), a chegada dos clientes (*Customer Arrivals*), tempos das atividades (*Process Delay*), etc. de acordo com a Figura 14.



Figura 14 - Módulos disponíveis

### 3.2.2. Módulo de configuração das estratégias

Apresenta-se na Figura 15 o aspeto do módulo das estratégias possíveis de simulação sem outras alterações de parâmetros. Conforme é possível observar, estão já pré-programadas três

estratégias, sendo que a *Strategy 0 (default)* já está carregada, sendo possível executar de imediato uma simulação e obter resultados.

As outras estratégias alteram as capacidades dos processos conforme o seu nome indica, ou seja, a *Strategy 1 (starving)*, limita drasticamente a capacidade em cada processo, o que implicará nos resultados obtidos, o aumento da quantidade das viaturas nas filas de espera e também o respetivo aumento de tempo de espera. Já a estratégia seguinte, carrega os parâmetros de forma precisamente contrária, sendo possível observar nos resultados da simulação, que existe muita folga no funcionamento da oficina.

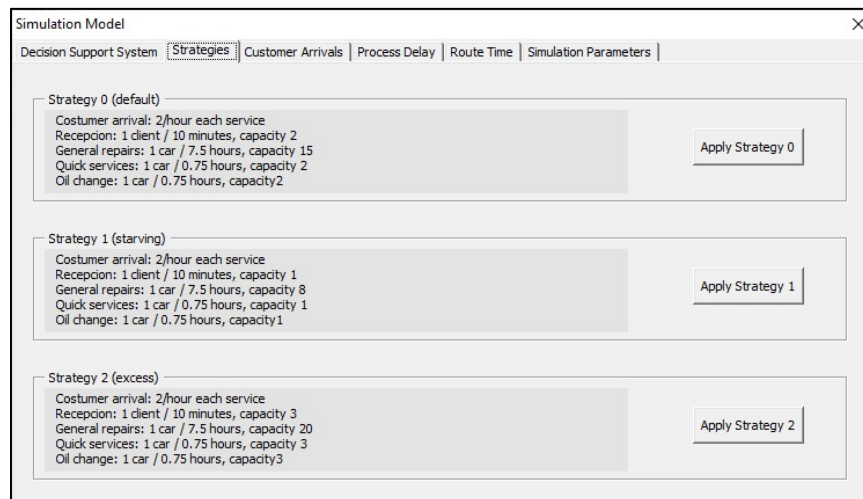


Figura 15 - Módulo das estratégias de simulação

O utilizador pode optar por qualquer uma das estratégias, sendo os valores carregados para os parâmetros respetivos. Nos restantes módulos (*Customer Arrivals* e *Process Delay*) é possível visualizar e alterar os valores da estratégia escolhida, dando então origem a um novo cenário customizado. Caso seja necessário voltar a carregar a uma estratégia, bastará aceder a este módulo e premir o botão correspondente, sendo então recarregados novamente os valores indicados, nos parâmetros respetivos.

### 3.2.3. Módulo de configuração da chegada dos clientes

Neste módulo que é apresentado na Figura 16, é possível alterar os valores relativos à chegada de clientes para os respetivos serviços. Para esse efeito o utilizador deverá premir o botão de *Read Variables*, sendo os valores de cada parâmetro apresentado imediatamente e abrindo simultaneamente as caixas de diálogo para inserção de novos valores. Caso o utilizador pretenda alterar algum dos parâmetros relativo a um dos tipos de cliente (ou mesmo todos), deverá preencher os campos respetivos e premir o botão *Change...* relativo ao serviço que está a alterar.

Figura 16 - Módulo Customer Arrivals

Conforme já referido, na eventualidade do utilizador escolher outra estratégia (no respetivo módulo), os valores dos parâmetros podem alterar de acordo com a estratégia selecionada. No caso particular do módulo de *Customer Arrivals*, para todas as estratégias optou-se por manter os valores para os parâmetros, por forma a ser possível constatar facilmente o resultado do funcionamento por defeito, e os resultados com as outras estratégias (altamente limitado na capacidade, ou com bastante folga).

Para a situação de alteração dos parâmetros, as distribuições de tempo disponíveis neste módulo (sendo iguais para os restantes módulos) apresentam-se na Figura 17.

Figura 17 - Lista das distribuições de tempo disponíveis

Após a seleção de uma distribuição de tempo, as caixas de diálogo imediatamente abaixo, ajustam-se por forma a receber os valores de acordo com a distribuição escolhida.

### 3.2.4. Módulo de configuração dos tempos de processamento dos diferentes processos da oficina

Na Figura 18 apresentada na página seguinte, o funcionamento é idêntico ao descrito no módulo anterior, ou seja, após premir o botão *Read Variables*, são apresentados os valores dos parâmetros que estão programados e simultaneamente são abertas as caixas de diálogo para inserção de novos valores.

Para cada uma das atividades apresentadas, é possível definir a capacidade de funcionamento e o tempo que demora a atividade. Adicionalmente, por forma a ser possível introduzir aleatoriedade

nos processos, existe a possibilidade de definir o tempo recorrendo às distribuições disponíveis já apresentadas na Figura 17.

The screenshot displays the 'Simulation Model' window with the 'Process Delay' tab selected. The interface is organized into two columns: 'Actual Value' and 'New Value'. Each column contains settings for four process types: Reception, General Repairs, Quick Services, and Oil Change. For each process, the 'Actual Value' column shows the current configuration, while the 'New Value' column shows the configuration after a change. A dropdown menu is open in the 'New Value' section for the 'Reception' process, listing available time distributions: Normal, Exponential, Poisson, LogNormal, Uniform, and Triangular. A 'Change Reception Process' button is located to the right of the dropdown. At the bottom center, there is a 'Read Variables' button.

Process	Section	Capacity	Time Distribution	Parameters: Mean	Parameters: StdDev
Reception	Actual Value	2	Normal	10 (Minutes)	5
	New Value		Normal		
General Repairs	Actual Value	15	Normal	7.5 (Hours)	2
	New Value				
Quick Services	Actual Value	2	Normal	0.75 (Hours)	0.25
	New Value				
Oil Change	Actual Value	2	Normal	0.75 (Hours)	0.25
	New Value				

Figura 18 - Módulo Process Delay

Assim que for selecionada uma distribuição, serão apresentadas novas caixas de diálogo de acordo com os parâmetros que são necessários carregar para essa mesma distribuição. Na Figura 19 apresenta-se um ecrã com várias distribuições diferentes e respetivos parâmetros. Este modelo de funcionamento relativos às distribuições, é idêntico em todos os módulos.

The screenshot displays the 'Simulation Model' window with the 'Simulation Parameters' tab selected. It is divided into two columns: 'Actual Value' and 'New Value'. Each process has a set of parameters including Capacity, Time Distribution, Mean, and StdDev. The 'New Value' column is highlighted with red boxes, indicating the current configuration for each process.

Process	Actual Value	New Value
Reception	Capacity: 2, Distribution: Normal, Mean: 10, StdDev: 5	Capacity: [ ], Distribution: Exponenc, Mean: [ ]
General Repairs	Capacity: 15, Distribution: Normal, Mean: 7.5, StdDev: 2	Capacity: [ ], Distribution: Normal, Mean: [ ], StdDev: [ ]
Quick Services	Capacity: 2, Distribution: Normal, Mean: 0.75, StdDev: 0.25	Capacity: [ ], Distribution: Uniform, Mean: [ ], Maximum: [ ]
Oil Change	Capacity: 2, Distribution: Normal, Mean: 0.75, StdDev: 0.25	Capacity: [ ], Distribution: Triangular, Mean: [ ], Mode: [ ], Maximum: [ ]

Figura 19 - Parâmetros das distribuições

### 3.2.5. Módulo de configuração dos tempos de rota

No módulo de *Route Time* apresentado na Figura 20 e cujo funcionamento é idêntico ao anteriores, é possível definir tempos de deslocação das viaturas dentro da oficina, entre o parque de estacionamento interno e os postos de trabalho. Existe novamente a possibilidade de introduzir alguma aleatoriedade por via das distribuições cujo funcionamento já foi descrito.

Relativamente aos parâmetros de *Route Time*, como os tempos de deslocação dentro da oficina são bastantes pequenos relativamente aos dos processos, optou-se por não refletir os mesmos nas estratégias pré-programadas. Adicionalmente não são aplicáveis parâmetros como capacidade ou quantidade.

Activity	Actual Value	New Value
Reception -> Internal Park	Time Distribution: Normal Parameters: Mean (Minutes): 2, StdDev: 0.5	Time Distribution: Uniform Parameters: Minimum: 2, Maximum: 3
Internal Park -> General Repairs	Time Distribution: Normal Parameters: Mean (Minutes): 5, StdDev: 1	Time Distribution: [Dropdown]
Internal Park -> Quick Repairs	Time Distribution: Normal Parameters: Mean (Minutes): 2, StdDev: 0.5	Time Distribution: [Dropdown]
Internal Park -> Oil Change	Time Distribution: Normal Parameters: Mean (Minutes): 3, StdDev: 0.5	Time Distribution: [Dropdown]

Figura 20 - Módulo Route Time

### 3.2.6. Módulo de configuração dos parâmetros de simulação

Na Figura 21 apresenta-se o módulo em que é possível alterar parâmetros da simulação, sendo o funcionamento idêntico aos anteriores. Os parâmetros são praticamente autoexplicativos, no entanto, relativamente aos dois primeiros parâmetros, caso sejam colocados a *true*, as variáveis e as estatísticas recolhidas serão inicializadas a cada replicação, podendo desta forma interferir com os resultados esperados na execução da simulação.

Parameter	Actual Value	New Value
Initialize System Between Replications	FALSE	False
Initialize Statistics Between Replications	FALSE	False
Number Of Replications	1	1
Hours Per Day	8	8
Replication Length	5	5
Replication Length Time Units	Days	Days
Warm Up Period	0	0
Warm Up Period Time Units	Minutes	Minutes

Figura 21 - Módulo Simulation Parameters

### 3.3. Animação do modelo

Após a alteração dos valores de acordo com o cenário pretendido, o utilizador deverá voltar ao módulo *Decision Support System*, e premir o botão *Run Simulation*. O simulador irá arrancar com uma animação gráfica similar à Figura 22 abaixo apresentada.

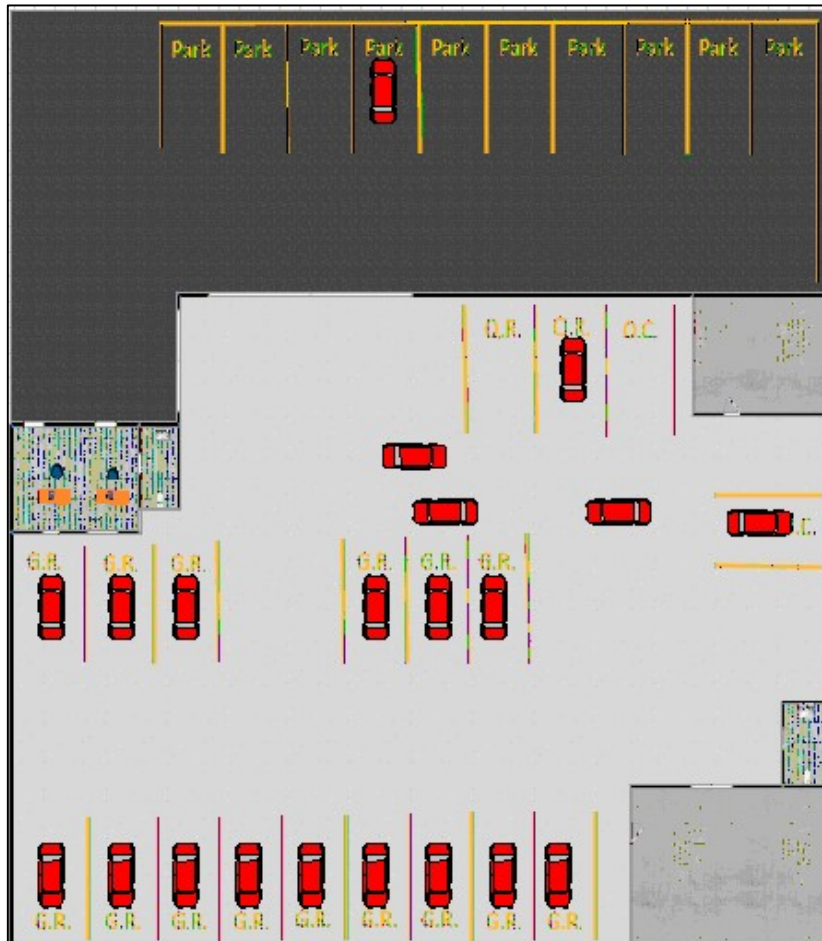


Figura 22 - Simulador em execução

### 3.4. Modelo lógico do sistema de apoio à decisão desenvolvido

Neste tópico apresenta-se o modelo lógico subjacente ao funcionamento do simulador, nas suas componentes principais, apresenta-se também o esquema subjacente ao estacionamento das viaturas e postos de trabalho. Adicionalmente apresentam-se o fluxograma de base dos módulos e parte do código desenvolvido para controlo da interface com o utilizador.

#### 3.4.1. Modelo lógico

Conforme apresentado na Figura 11 (subtópico 3.1.), o sistema inicia com a chegada de viaturas e o seu registo na atividade da receção. Por forma a ser possível introduzir variações nos cenários, foi definida uma entidade para cada tipo de serviço (troca de óleo, serviços rápidos ou reparações

gerais), sendo desta forma possível parametrizar a quantidade de entidades que entram no modelo e qual a respetiva distribuição no tempo (exponencial, normal ou outra).

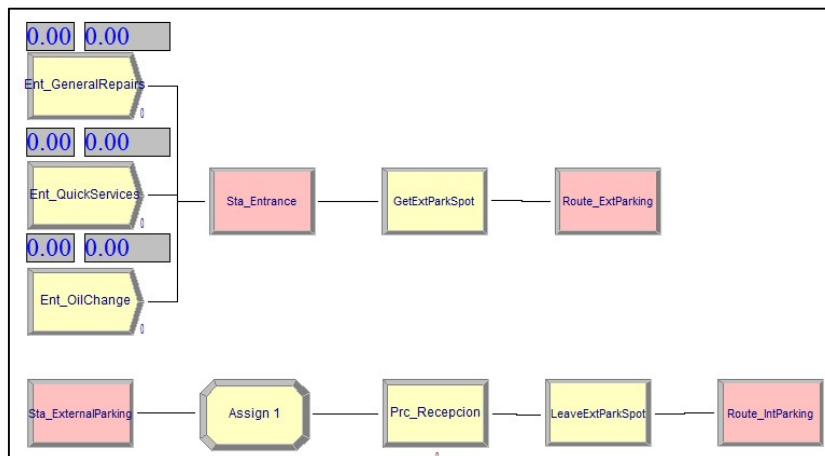


Figura 23 - Modelo lógico da recepção

Conforme se pode observar na Figura 23, estão representadas as três entidades (representam as viaturas) que entram no sistema e são processadas no modelo. Por forma a se conseguir aleatoriedade no estacionamento das viaturas e respetiva visualização na animação, foram introduzidas as funções de *Seize* (GetExtParkSpot), *Assign* (Assign 1) e *Release* (LeaveExtParkSpot).

Na Tabela 1 apresentam-se as características das entidades, sendo de realçar a utilização de variáveis (passíveis de serem alteradas via interface do utilizador) no campo *Expression* e *Entities per Arrival* por forma a permitir a criação de novos cenários.

Tabela 1 - Características das entidades

	Name	Entity Type	Type	Expression	Units	Entities per Arrival	Max Arrivals	First Creation
1	Ent_GeneralRepair	Entity 1	Expression	Eval(Alter_Time_Entity1)	Hours	Alter_Quantity_Entity1	Infinite	0.0
2	Ent_QuickServices	Entity 2	Expression	Eval(Alter_Time_Entity2)	Hours	Alter_Quantity_Entity2	Infinite	0.0
3	Ent_OilChange	Entity 3	Expression	Eval(Alter_Time_Entity3)	Hours	Alter_Quantity_Entity3	Infinite	0.0

Na Figura 24 apresentam-se as definições do processo da recepção, na qual se pode constatar no campo *Expression* a definição de uma variável por forma a ser possível a alteração via interface do utilizador.

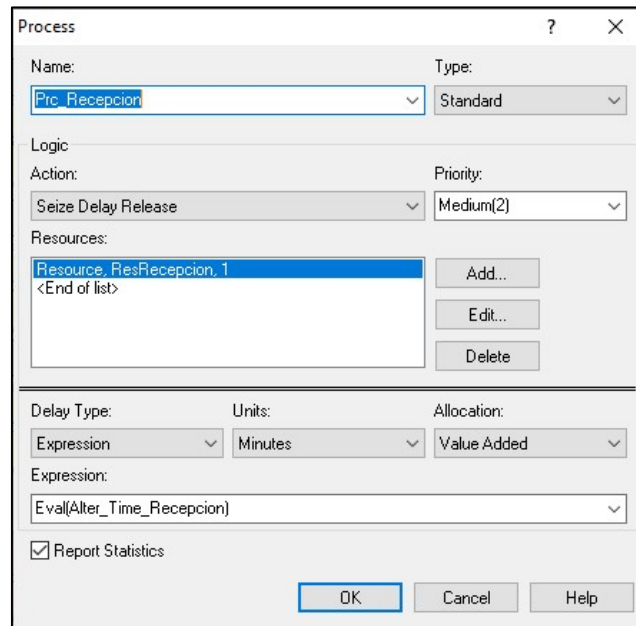


Figura 24 - Definição do processo de recepção

Na caixa de diálogo dos *Resources* ainda na Figura 24, constata-se que o recurso tem o nome *ResRecepcion* e uma quantidade de 1. Esta quantidade representa a disponibilidade do recurso, sendo que um valor inferior a 1 não permitiria a utilização do recurso no seu máximo. Há, no entanto, forma de alterar a capacidade através da interface de utilizador, ou seja, mesmo sendo apenas um processo de recepção, é possível parametrizá-lo por forma a simular vários em paralelo (simulando, portanto, a quantidade de rececionistas).

O passo seguinte no modelo lógico é a decisão do serviço a executar e respetivo encaminhamento para o processo respetivo, ou seja, o serviço a efetuar na viatura. Na Figura 25, apresenta-se o ponto de decisão e seguidamente o bloco do tipo *Seize* (blocos cujo nome começa por *Get...*) necessários para a criação da aleatoriedade na animação relativo ao posto de trabalho a ocupar. Para cada serviço disponibilizado é necessário um bloco *Seize*.

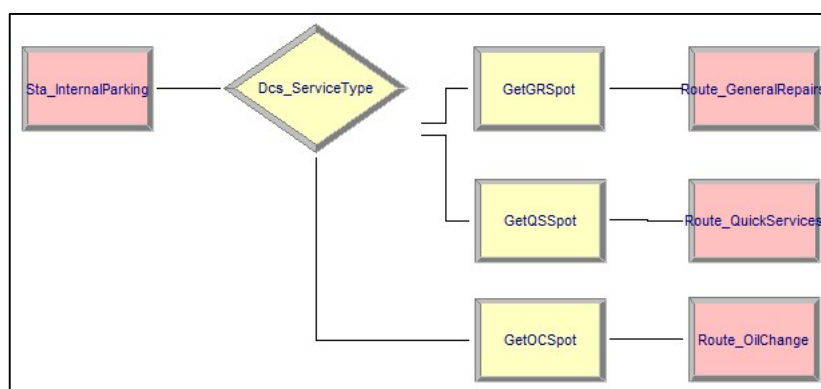


Figura 25 - Bloco de decisão do serviço a executar

A partir do momento que a entidade é encaminhada para o serviço respetivo, o mesmo será executado de acordo com o a parametrização do respetivo processo. Na Figura 26 representa-se o bloco lógico desta fase, onde se observa a utilização de blocos do tipo *Assign* (blocos com o mesmo

nome e número sequencial), e blocos do tipo Release (blocos com o nomes *Leave...*) necessários para a animação da simulação.

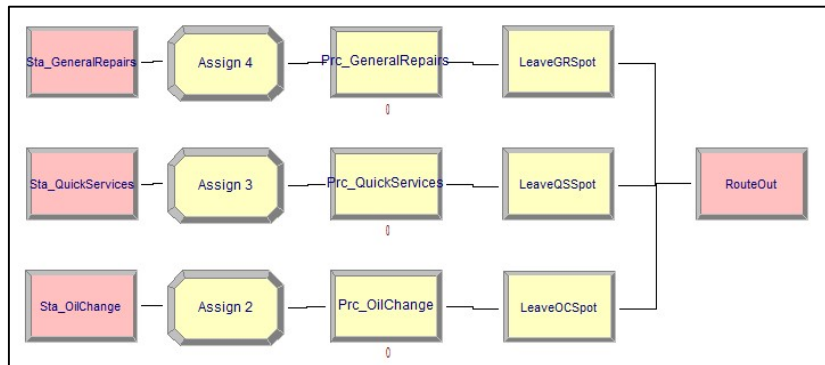


Figura 26 - Bloco de execução dos serviços

Na Figura 27, é possível ver as parametrizações carregadas para o processo de *General Repairs*, sendo possível a alteração dos valores através da interface do utilizador.

Figura 27 - Definição do processo de *General Repairs*

A definição dos processos é similar entre os mesmos, resumindo-se a sua parametrização na Tabela 2 apresentada seguidamente.

Tabela 2 - Características dos processos

	Name	Type	Action	Priority	Resources	Delay Type	Units	Allocation	Expression	Report Statistics
1	Prc_Recepcion	Standard	Seize Delay Release	Medium(2)	1 rows	Expression	Minutes	Value Added	Eval(Alter_Time_Recepcion)	<input checked="" type="checkbox"/>
2	Prc_GeneralRepairs	Standard	Seize Delay Release	Medium(2)	1 rows	Expression	Hours	Value Added	Eval(Alter_Time_GeneralRepairs)	<input checked="" type="checkbox"/>
3	Prc_OilChange	Standard	Seize Delay Release	Medium(2)	1 rows	Expression	Hours	Value Added	Eval(Alter_Time_OilChange)	<input checked="" type="checkbox"/>
4	Prc_QuickServices	Standard	Seize Delay Release	Medium(2)	1 rows	Expression	Hours	Value Added	Eval(Alter_Time_QuickServices)	<input checked="" type="checkbox"/>

Seguidamente as entidades saem do modelo de simulação, sendo este bloco do modelo lógico apresentado na Figura 28.

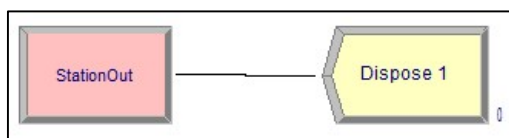


Figura 28 - Saída das entidades do simulador

Na Tabela 3 apresentam-se as parametrizações dos recursos que são utilizados em cada processo, e onde é possível constatar a capacidade que está parametrizada, correspondendo por sua vez à estratégia 0 (estratégia que está pré-selecionada por defeito).

Tabela 3 - Recursos configurados

	Name	Type	Capacity	Busy / Hour	Idle / Hour	Per Use	StateSet Name	Initial State	Failures	Report Statistics
1	ResRecepcion	Fixed Capacity	2	0.0	0.0	0.0			0 rows	<input checked="" type="checkbox"/>
2	RscGeneralRepairs	Fixed Capacity	15	0.0	0.0	0.0			0 rows	<input checked="" type="checkbox"/>
3	RscOilChange	Fixed Capacity	2	0.0	0.0	0.0			0 rows	<input checked="" type="checkbox"/>
4	RscQuickServices	Fixed Capacity	2	0.0	0.0	0.0			0 rows	<input checked="" type="checkbox"/>

Conforme já referido, foram definidas várias variáveis que podem ser alteradas via interface do utilizador. Na Figura 29 seguinte apresentam-se apenas algumas e em sobreposição o valor de uma variável de tempo, visualizando-se que segue uma distribuição normal, de média 1 (hora) e desvio padrão de 0.1.

	Name	Rows	Columns	Data Type	Clear Option	File Name	Initial Values	Report Statistics
1	Alter_Quantity_Entity1			Real	System		1 rows	<input type="checkbox"/>
2	Alter_Time_Entity1			String	System		1 rows	<input type="checkbox"/>
3	Alter_Quantity_Entity2			Real	System		1 rows	<input type="checkbox"/>
4	Alter_Time_Entity2			String	System		1 rows	<input type="checkbox"/>
5	Alter_Quantity_Entity3			Real	System		1 rows	<input type="checkbox"/>
6	Alter_Time_Entity3			String	System		1 rows	<input type="checkbox"/>
7	Alter_Time_Recepcion			String	System		1 rows	<input type="checkbox"/>

Figura 29 - Parametrização de variáveis

Na Figura 30, apresenta-se o aspeto global do modelo lógico criado para este simulador, onde é possível visualizar simultaneamente os vários blocos que compõe o modelo lógico.

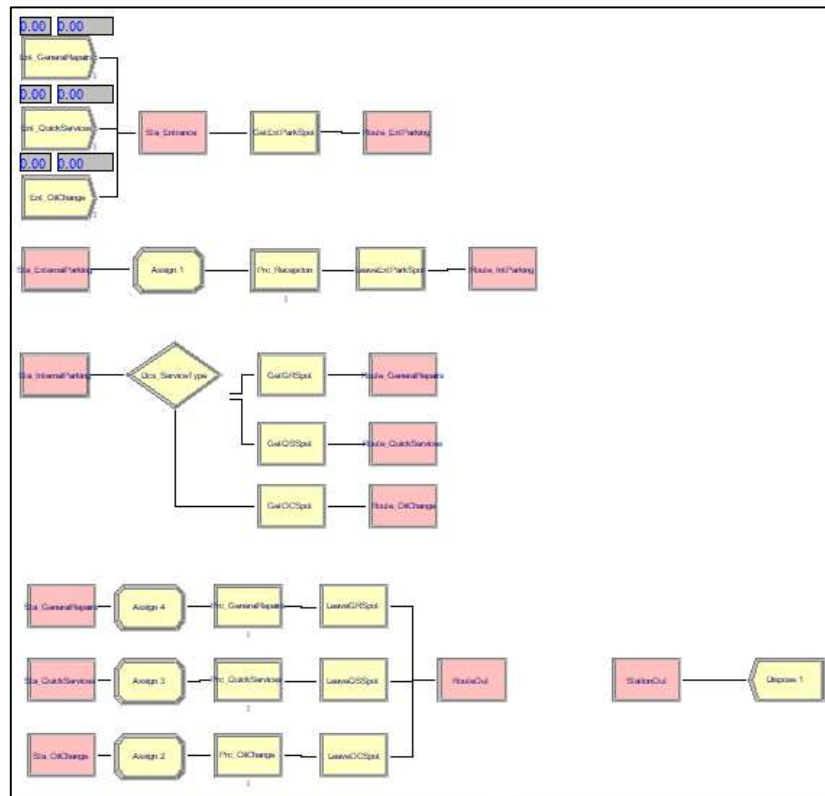


Figura 30 - Modelo lógico global

### 3.4.2. Rotas e estacionamento

Por forma a ser possível criar uma animação que represente o funcionamento de uma oficina automóvel no simulador, procedeu-se à criação de um desenho que representa uma oficina em potencial.

Na Figura 31, a entrada das viaturas é efetuada na zona assinalada com o número um (a verde), e vão parquear de forma aleatória nos lugares disponíveis na zona identificada com o número dois. Seguidamente há um compasso de espera que corresponde ao processo da receção da viatura. Estando o processo de receção concluído, a viatura segue para a zona da intervenção correspondente ao serviço programado, sendo a zona três correspondente aos serviços rápidos, a zona quatro à troca de óleo, e a zona 5 às reparações gerais. A zona seis corresponde à receção, e a sete aos armazéns de peças, pneus, óleos e outras. A saída das viaturas acontece na zona oito.

As viaturas deslocam-se sobre as linhas a azul, o que aplicado a um desenho de oficina real, permitiria visualizar zonas de estrangulamento na passagem das mesmas.

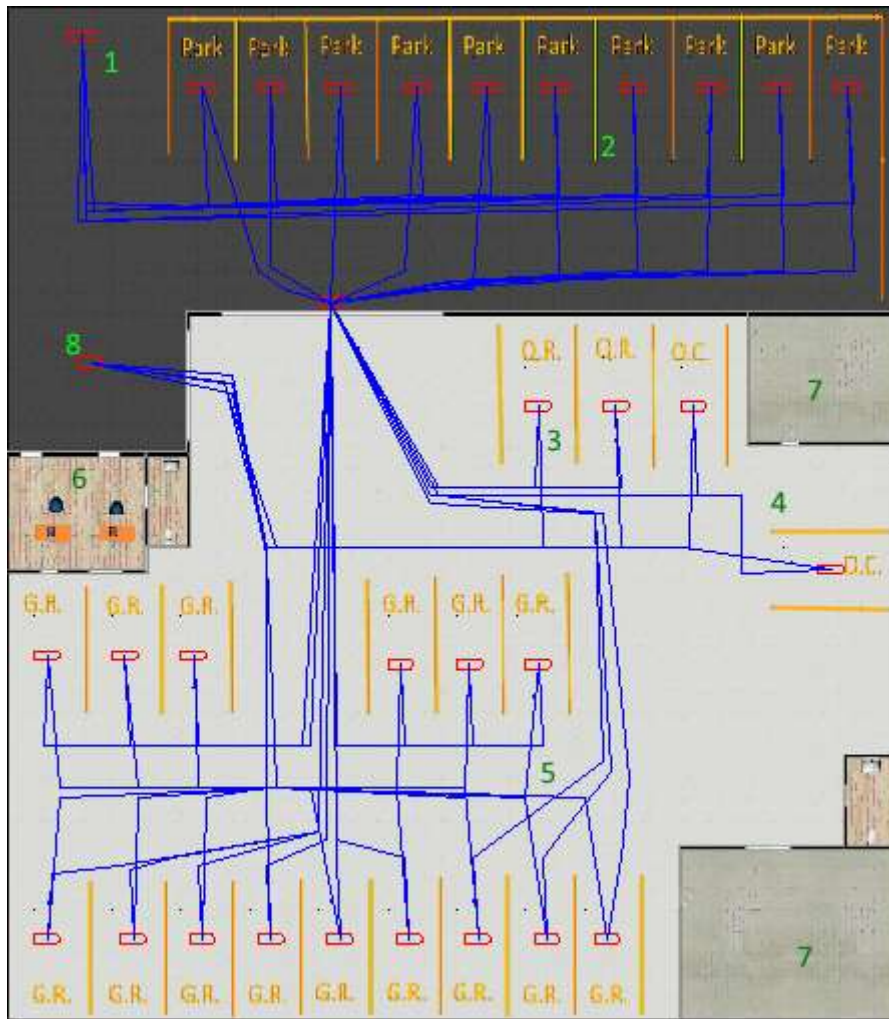


Figura 31 - Representação oficina

### 3.5. Programa de controlo da interface do utilizador

Nesta secção apresentam-se o funcionamento interno dos módulos desenvolvidos e os excertos de código principais.

#### 3.5.1. Fluxograma do funcionamento

Os vários ecrãs da interface com o utilizador já foram apresentados no tópico 4.1, tendo sido dada grande importância à coerência do funcionamento dos vários módulos e respetiva apresentação. Na Figura 32 apresenta-se o fluxograma que guiou o desenvolvimento dos vários módulos, havendo apenas diferenças na quantidade e nomes das variáveis tratadas em cada um.

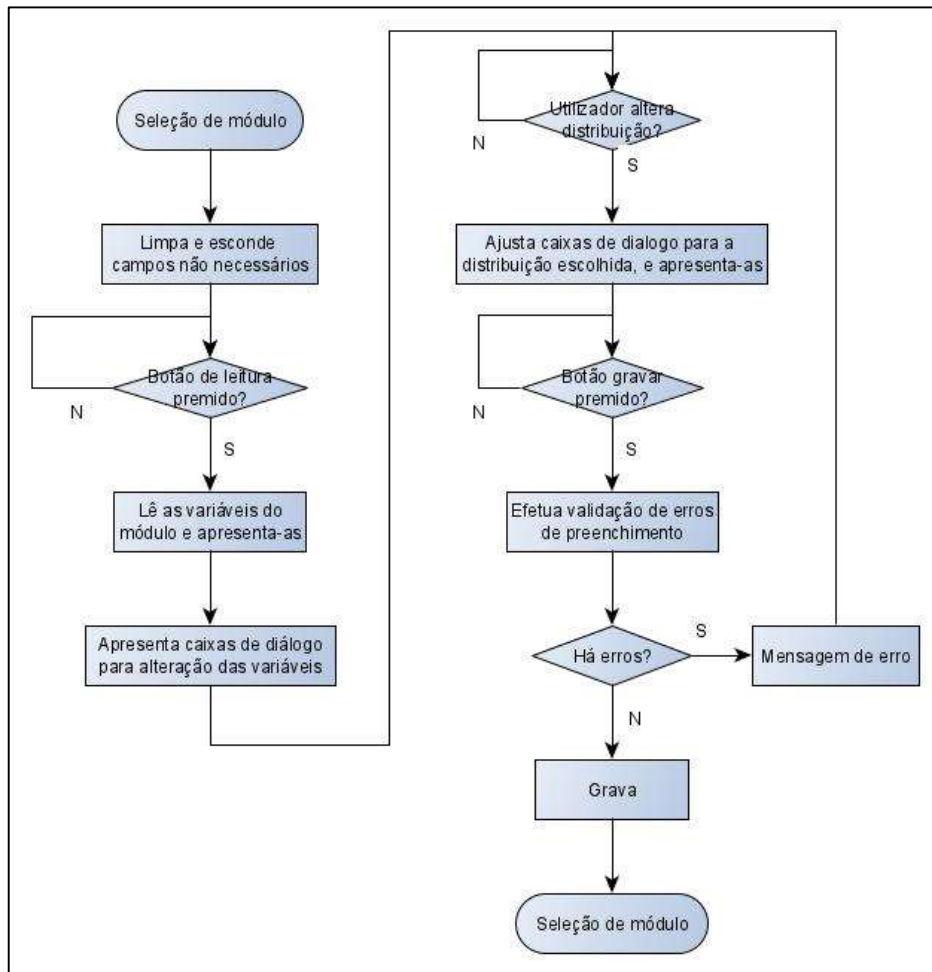


Figura 32 - Fluxograma dos módulos <sup>7</sup>

Decorrente do próprio conceito de programação por objetos, em que cada um dos objetos tem as suas propriedades e respetivos métodos, basta mudar o foco entre módulos ou até mesmo entre campos, para provocar eventos que por sua vez despoletam ações programadas.

### 3.5.2. Controlo do módulo inicial

Conforme já mostrado na secção 3.2.1 Módulo inicial, pretende-se ter uma interface para o utilizador muito simples e coerente. Por forma a se conseguir o aspeto apresentado, foi utilizada a opção de *Multipage* no ambiente de programação Visual Basic, permitindo desta forma a colocação dos módulos por separadores, com a vantagem adicional da possibilidade de execução de algumas ações sempre que há nova seleção de módulo. Na Figura 33 é representado o fluxograma das ações sempre que um módulo é selecionado.

<sup>7</sup> Fluxograma criado com o *software* yEd Version 3.19.1.1

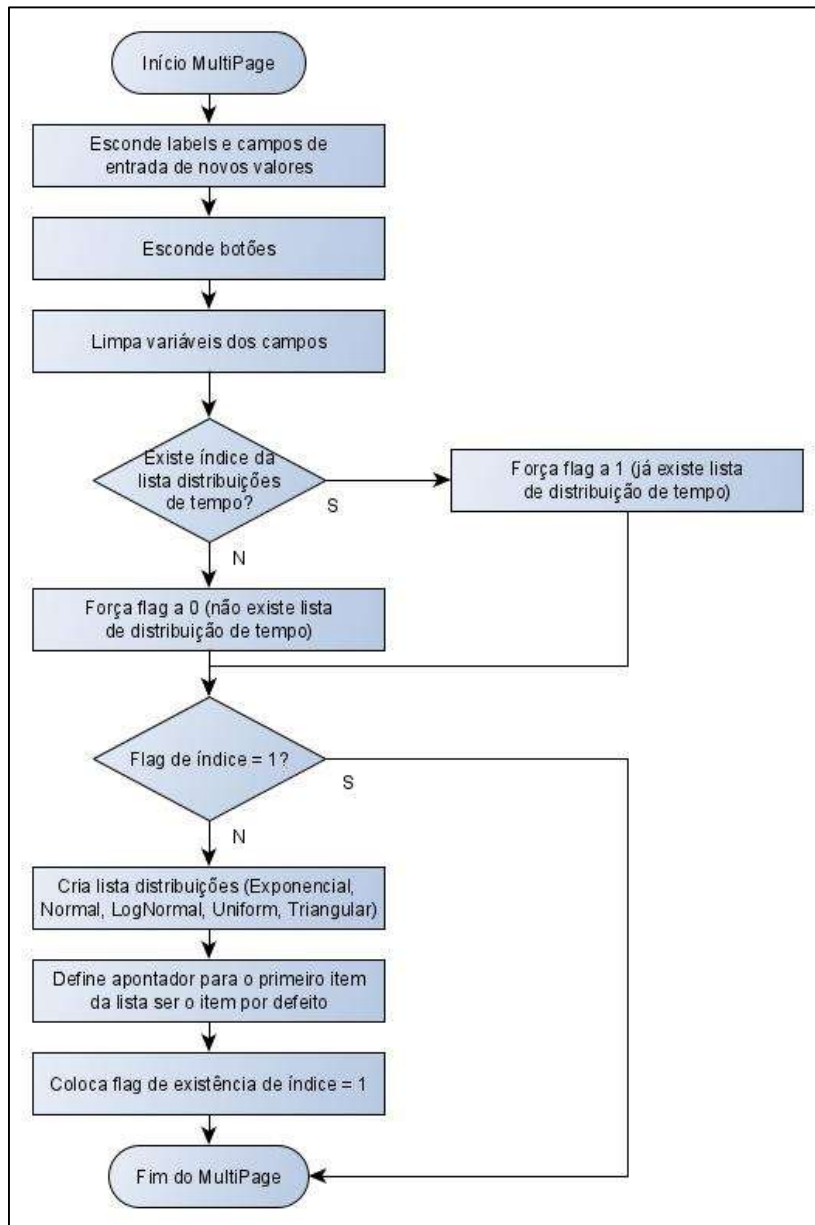


Figura 33 - Fluxograma MultiPage<sup>8</sup>

Seguidamente representa-se na Figura 34 excerto do código estruturado no fluxograma já indicado. Em termos práticos o código abaixo executa a normalização de um módulo, ou seja, de cada vez que se seleciona um módulo, as *labels* e caixas de diálogo são refrescadas (escondidas ou mostradas), limpando as variáveis das caixas (para o caso de terem algum valor carregado), e por último garantindo que as *ComboBox* das listas de distribuição de tempo, tenham essas mesmas listas devidamente carregadas.

```

Private Sub MultiPage1_Change()

UserForm1.Label7.Visible = False
...
UserForm1.TextBox1.Visible = False
...

```

<sup>8</sup> Fluxograma criado com o *software* yEd Version 3.19.1.1

```

UserForm1.CommandButton4.Visible = False
...
UserForm1.TextBox1.value = ""

Var_Combobox1 = UserForm1.Combobox1.ListIndex
Select Case Var_Combobox1
    Case -1
        Var_Combobox1Existe = 0
    Case 0
        Var_Combobox1Existe = 1
    Case 1
        Var_Combobox1Existe = 1
    Case 2
        Var_Combobox1Existe = 1
    Case 3
        Var_Combobox1Existe = 1
    Case 4
        Var_Combobox1Existe = 1
    Case 5
        Var_Combobox1Existe = 1
    Case 6
        Var_Combobox1Existe = 1
    Case Else
        Var_Combobox1Existe = 0
End Select

If Var_Combobox1Existe = 0 Then
    UserForm1.Combobox1.AddItem "" 'ListIndex = 0
    UserForm1.Combobox1.AddItem "Exponencial" 'ListIndex = 1
    UserForm1.Combobox1.AddItem "Poisson" 'ListIndex = 2
    UserForm1.Combobox1.AddItem "Normal" 'ListIndex = 3
    UserForm1.Combobox1.AddItem "LogNormal" 'ListIndex = 4
    UserForm1.Combobox1.AddItem "Uniform" 'ListIndex = 5
    UserForm1.Combobox1.AddItem "Triangular" 'ListIndex = 6
    'Set combo box to first entry
    UserForm1.Combobox1.ListIndex = 0
    Var_Combobox1Existe = 1
End If

```

Figura 34 - Excerto de código do MultiPage

No módulo inicial, estão presentes os botões de *Run Simulation* e *Exit* (ver Figura 35), que executam o que o próprio nome indica.



Figura 35 - Botões de Run Simulation e Exit

```

Private Sub CommandButton1_Click()

Dim m As Model
Dim s As SIMAN

Set m = ThisDocument.Model
Set s = m.SIMAN

m.Pause
m.End

UserForm1.Hide

End Sub

```

Figura 36 - Código do botão Exit

Ao premir o botão *Exit*, é ativado o objeto (neste caso o modelo do Arena) via os métodos *Pause* e *End* como se pode observar na Figura 36. Quando chega ao momento da instrução de esconder o *form* implicando a devolução do controlo ao *software* Arena, a simulação não será executada.

Como se pode ver na Figura 37, caso seja premido o botão *Run*, apenas é devolvido o controlo ao *software* Arena, dando dessa forma início à execução da simulação.

```
Private Sub CommandButton2_Click()

Dim m As Model
Dim s As SIMAN

Set m = ThisDocument.Model
Set s = m.SIMAN

UserForm1.Hide

End Sub
```

Figura 37 - Código do botão Run

### 3.5.3. Controlo do módulo de estratégias

O módulo das estratégias representa-se pelo fluxograma apresentado na Figura 38, apresentando-se seguidamente excerto de código de um dos botões de ativação de estratégia na Figura 39.

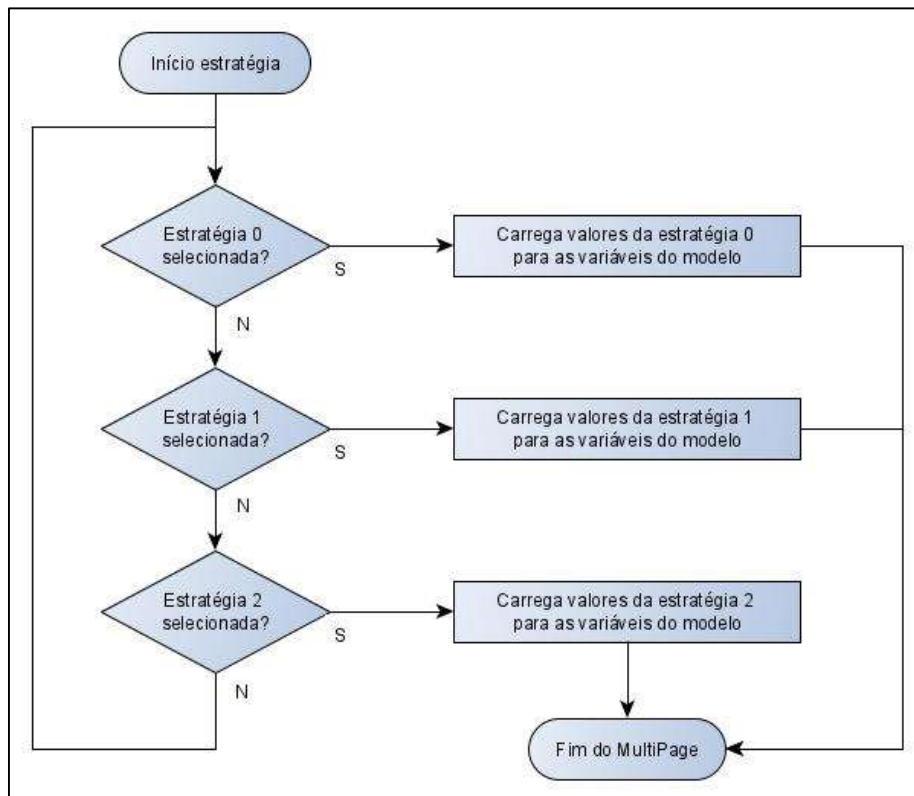


Figura 38 - Fluxograma do módulo das estratégias<sup>9</sup>

<sup>9</sup> Fluxograma criado com o *software* yEd Version 3.19.1.1

O código para ativação da estratégia é constituído por três blocos, facilmente visíveis na Figura 39, sendo o primeiro bloco relativo às variáveis que controlam a quantidade de viaturas que chegam à oficina, o segundo bloco é relativo às variáveis do tempo do processamento da receção da viatura, e o último bloco trata por sua vez das variáveis do tempo de processamento para cada tipo de serviço. A escolher a estratégia os valores serão carregados para as variáveis do *software* Arena.

```
Private Sub CommandButton19_Click()
Dim m As Model
Dim s As SIMAN

Set m = ThisDocument.Model
Set s = m.SIMAN

s.VariableArrayValue(s.SymbolNumber("Alter_Quantity_Entity1")) = "2"
s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity1")) = "Norm(1,0.1)"
s.VariableArrayValue(s.SymbolNumber("Alter_Quantity_Entity2")) = "2"
s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity2")) = "Norm(1,0.1)"
s.VariableArrayValue(s.SymbolNumber("Alter_Quantity_Entity3")) = "2"
s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity3")) = "Norm(1,0.1)"

s.ResourceCapacity(s.SymbolNumber("ResRecepcion")) = "2"
s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Recepcion")) = "Norm(10,5)"

s.ResourceCapacity(s.SymbolNumber("RscGeneralRepairs")) = "15"
s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_GeneralRepairs")) = "Norm(7.5,2)"
s.ResourceCapacity(s.SymbolNumber("RscQuickServices")) = "2"
s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_QuickServices")) = "Norm(0.75,0.25)"
s.ResourceCapacity(s.SymbolNumber("RscOilChange")) = "2"
s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_OilChange")) = "Norm(0.75,0.25)"

End Sub
```

Figura 39 - Código de carregamento de valores das estratégias

### 3.5.4. Controlo dos restantes módulos – Botão *Read Variables*

O botão *Read Variables*, é o único botão visível assim que se seleciona qualquer um dos restantes módulos, por forma a manter a simplicidade da interface do utilizador, ou seja, todos os campos para preenchimento, *labels* e botões são escondidos até serem necessários.

A partir do momento que qualquer um dos módulos (chegada de clientes, tempo de processamento, tempo de rota, parâmetros de simulação) é selecionado, o programa fica a aguardar que seja premido o botão *Read Variables*. A partir desse momento, entra em ação o algoritmo do fluxograma apresentado na Figura 40.

O módulo vai ler os valores das variáveis que estão carregadas no Arena, e de acordo com a distribuição que está carregada vai ajustar os campos e *labels* que são apresentadas. A título de exemplo se a distribuição for normal, irão ser apresentadas as *labels* média e desvio padrão, e no caso de ser uma distribuição uniforme, as *labels* ficarão como mínimo e máximo.

Seguidamente à atividade de mostrar os campos e *labels*, é apresentado um campo de entrada de dados e uma *ComboBox* para valores a introduzir, que serão tratados no próximo tópico.

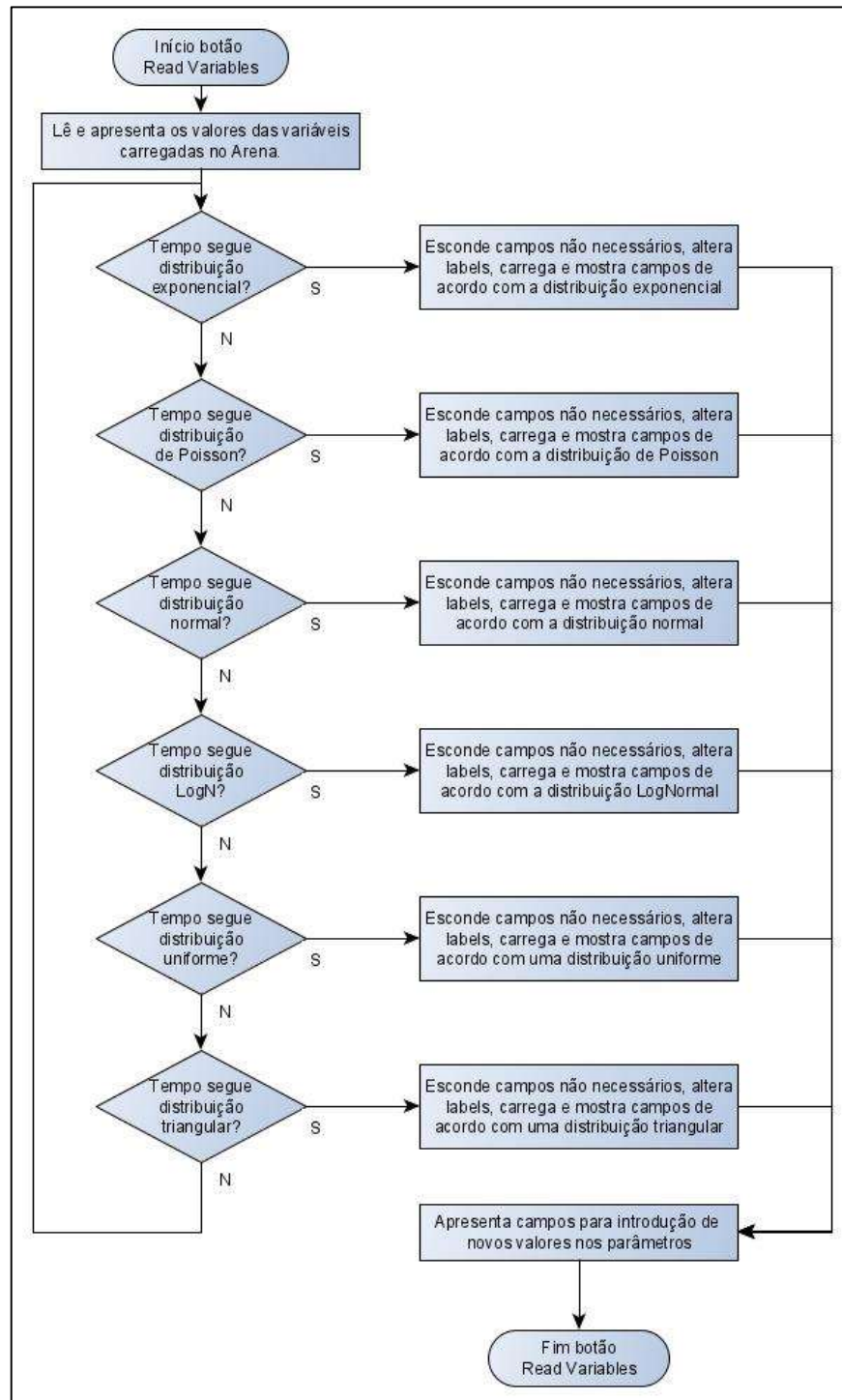


Figura 40 - Fluxograma botão Read Variables <sup>10</sup>

Na Figura 41, é apresentado um excerto do código do botão *Read Variables*. O código é igual para os vários tipos de clientes (reparações gerais, serviços rápidos, troca de óleo) adicionando mais as variáveis respetivas para os serviços em causa.

<sup>10</sup> Fluxograma criado com o software yEd Version 3.19.1.1

```

Private Sub CommandButton3_Click()

Dim m As Model
Dim s As SIMAN

Set m = ThisDocument.Model
Set s = m.SIMAN

UserForm1.TextBox1.value = s.VariableArrayValue(s.SymbolNumber("Alter_Quantity_Entity1"))
UserForm1.TextBox1.Visible = True

Var_ExpressionGR = s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity1"))
Var_DistributionGR = Left(Var_ExpressionGR, 4)

Select Case Var_DistributionGR
    Case "Expo", "EXPO"
        UserForm1.Label8.Visible = False
        UserForm1.Label9.Visible = False
        UserForm1.TextBox4.Visible = False
        UserForm1.TextBox5.Visible = False
        UserForm1.TextBox2.value = "Exponencial"
        UserForm1.TextBox3.value = Val(Mid(Var_ExpressionGR, 6))
        UserForm1.Label7.Caption = "Mean"
        UserForm1.Label7.Visible = True
        UserForm1.TextBox2.Visible = True
        UserForm1.TextBox3.Visible = True
    Case "Pois", "POIS"
        UserForm1.Label8.Visible = False
        UserForm1.Label9.Visible = False
        UserForm1.TextBox4.Visible = False
        UserForm1.TextBox5.Visible = False
        UserForm1.TextBox2.value = "Poisson"
        UserForm1.TextBox3.value = Val(Mid(Var_ExpressionGR, 6))
        UserForm1.Label7.Caption = "Mean"
        UserForm1.Label7.Visible = True
        UserForm1.TextBox2.Visible = True
        UserForm1.TextBox3.Visible = True
    Case "Norm", "NORM"
        UserForm1.Label9.Visible = False
        UserForm1.TextBox5.Visible = False
        UserForm1.TextBox2.value = "Normal"
        UserForm1.TextBox3.value = Val(Mid(Var_ExpressionGR, 6))
        UserForm1.TextBox4.value = Val(Mid(Var_ExpressionGR, InStr(Var_ExpressionGR, ",") +
1))
        UserForm1.Label7.Caption = "Mean"
        UserForm1.Label7.Visible = True
        UserForm1.Label8.Caption = "StdDev"
        UserForm1.Label8.Visible = True
        UserForm1.TextBox2.Visible = True
        UserForm1.TextBox3.Visible = True
        UserForm1.TextBox4.Visible = True
    Case "Logn", "LOGN"
        UserForm1.Label9.Visible = False
        UserForm1.TextBox5.Visible = False
        UserForm1.TextBox2.value = "LogNormal"
        UserForm1.TextBox3.value = Val(Mid(Var_ExpressionGR, 6))
        UserForm1.TextBox4.value = Val(Mid(Var_ExpressionGR, InStr(Var_ExpressionGR, ",") +
1))
        UserForm1.Label7.Caption = "LogMean"
        UserForm1.Label7.Visible = True
        UserForm1.Label8.Caption = "LogStd"
        UserForm1.Label8.Visible = True
        UserForm1.TextBox2.Visible = True
        UserForm1.TextBox3.Visible = True
        UserForm1.TextBox4.Visible = True
    Case "Unif", "UNIF"
        UserForm1.Label9.Visible = False

```

```

        UserForm1.TextBox5.Visible = False
        UserForm1.TextBox2.value = "Uniform"
        UserForm1.TextBox3.value = Val(Mid(Var_ExpressionGR, 6))
        UserForm1.TextBox4.value = Val(Mid(Var_ExpressionGR, InStr(Var_ExpressionGR, ",") +
1))
        UserForm1.Label7.Caption = "Minimum"
        UserForm1.Label7.Visible = True
        UserForm1.Label8.Caption = "Maximum"
        UserForm1.Label8.Visible = True
        UserForm1.TextBox2.Visible = True
        UserForm1.TextBox3.Visible = True
        UserForm1.TextBox4.Visible = True
    Case "Tria", "TRIA"
        UserForm1.TextBox2.value = "Triangular"
        UserForm1.TextBox3.value = Val(Mid(Var_ExpressionGR, 6))
        UserForm1.TextBox4.value = Val(Mid(Var_ExpressionGR, InStr(Var_ExpressionGR, ",") +
1))
        UserForm1.TextBox5.value = Val(Mid(Var_ExpressionGR, InStr(InStr(Var_ExpressionGR,
",") + 1, Var_ExpressionGR, ",") + 1))
        UserForm1.Label7.Caption = "Minimum"
        UserForm1.Label7.Visible = True
        UserForm1.Label8.Caption = "Mode"
        UserForm1.Label8.Visible = True
        UserForm1.Label9.Caption = "Maximum"
        UserForm1.Label9.Visible = True
        UserForm1.TextBox2.Visible = True
        UserForm1.TextBox3.Visible = True
        UserForm1.TextBox4.Visible = True
        UserForm1.TextBox5.Visible = True
End Select

UserForm1.TextBox17.Visible = True
UserForm1.TextBox26.Visible = True
UserForm1.TextBox30.Visible = True
UserForm1.ComboBox1.Visible = True
UserForm1.ComboBox2.Visible = True
UserForm1.ComboBox3.Visible = True

End Sub

```

Figura 41 - Excerto do código do botão Read Variables

### 3.5.5. Controlo dos restantes módulos – Seleção de distribuição de tempo

A partir do momento que é selecionada uma nova distribuição de tempo, entra em ação o algoritmo apresentado na Figura 42. De acordo com o fluxograma, são disponibilizados os campos e respetivas *Labels* para preenchimento, de acordo com a distribuição escolhida. Adicionalmente é apresentado o botão de *Change* relativo à variável a mudar, que irá gravar os valores dos campos, nas variáveis dos parâmetros. A seleção da distribuição de tempo, só pode ser feita selecionando a partir da lista disponível, evitando-se desta forma erros de digitação.

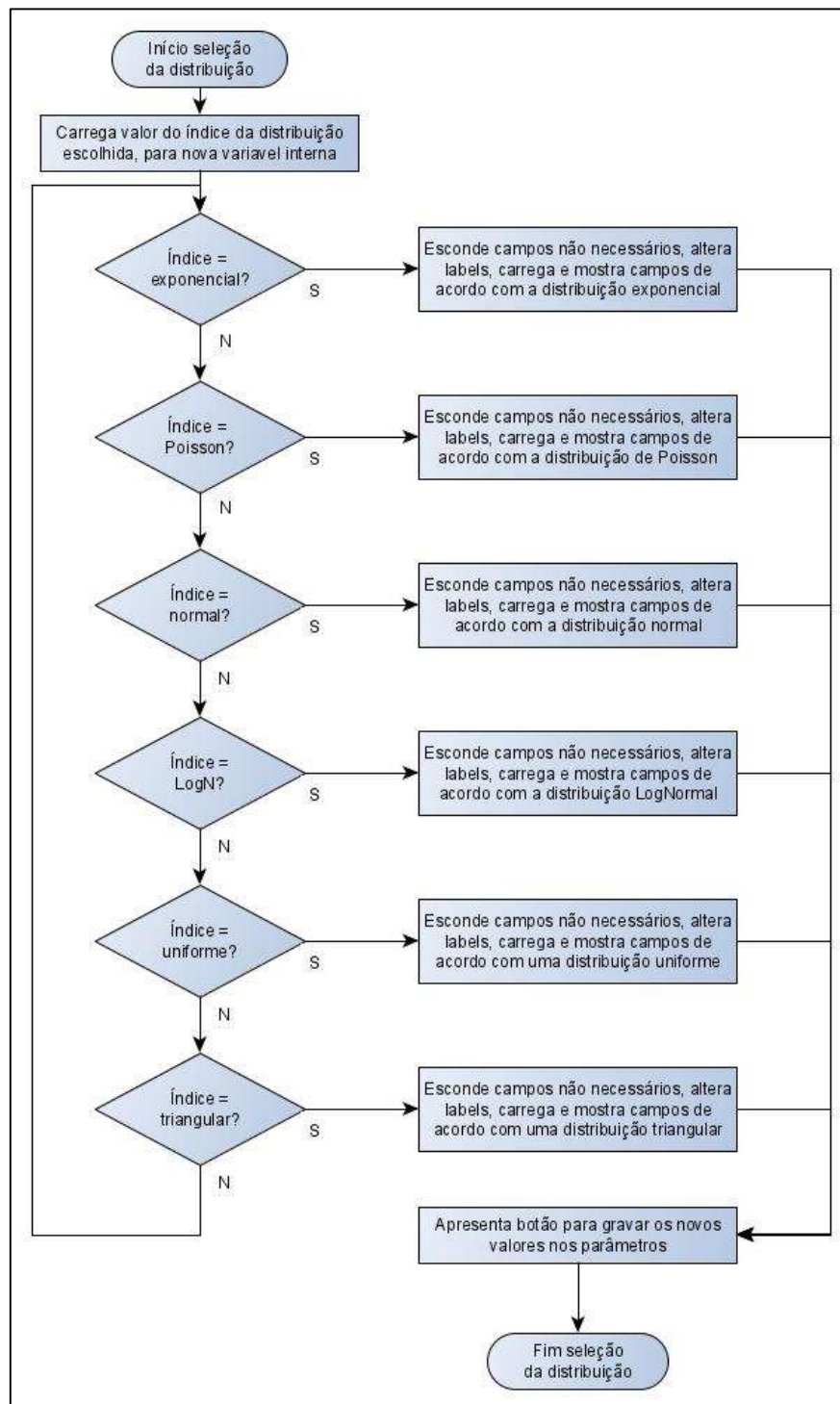


Figura 42 - Fluxograma seleção da distribuição de tempo <sup>11</sup>

Por forma a manter a coerência entre os vários módulos, a cada seleção diferente da distribuição de tempo, os campos imediatamente abaixo são ajustados sendo também apresentado o botão da gravação dos novos valores conforme se pode ver no excerto de código da Figura 43.

```

Private Sub ComboBox1_Change()
    Var_Combobox1Alterada = UserForm1.ComboBox1.ListIndex
  
```

<sup>11</sup> Fluxograma criado com o *software* yEd Version 3.19.1.1

```
Select Case Var_Combobox1Alterada
  Case 1 'Exponencial
    UserForm1.Label28.Visible = False
    UserForm1.Label29.Visible = False
    UserForm1.TextBox20.Visible = False
    UserForm1.TextBox21.Visible = False
    UserForm1.Label27.Caption = "Mean"
    UserForm1.Label27.Visible = True
    UserForm1.TextBox19.Visible = True
    UserForm1.CommandButton4.Visible = True
  Case 2 'Poisson
    UserForm1.Label28.Visible = False
    UserForm1.Label29.Visible = False
    UserForm1.TextBox20.Visible = False
    UserForm1.TextBox21.Visible = False
    UserForm1.Label27.Caption = "Mean"
    UserForm1.Label27.Visible = True
    UserForm1.TextBox19.Visible = True
    UserForm1.CommandButton4.Visible = True
  Case 3 'Normal
    UserForm1.Label29.Visible = False
    UserForm1.TextBox21.Visible = False
    UserForm1.Label27.Caption = "Mean"
    UserForm1.Label27.Visible = True
    UserForm1.Label28.Caption = "StdDev"
    UserForm1.Label28.Visible = True
    UserForm1.TextBox19.Visible = True
    UserForm1.TextBox20.Visible = True
    UserForm1.CommandButton4.Visible = True
  Case 4 'LogNormal
    UserForm1.Label29.Visible = False
    UserForm1.TextBox21.Visible = False
    UserForm1.Label27.Caption = "LogMean"
    UserForm1.Label27.Visible = True
    UserForm1.Label28.Caption = "LogStd"
    UserForm1.Label28.Visible = True
    UserForm1.TextBox19.Visible = True
    UserForm1.TextBox20.Visible = True
    UserForm1.CommandButton4.Visible = True
  Case 5 'Uniform
    UserForm1.Label29.Visible = False
    UserForm1.TextBox21.Visible = False
    UserForm1.Label27.Caption = "Minimum"
    UserForm1.Label27.Visible = True
    UserForm1.Label28.Caption = "Maximum"
    UserForm1.Label28.Visible = True
    UserForm1.TextBox19.Visible = True
    UserForm1.TextBox20.Visible = True
    UserForm1.CommandButton4.Visible = True
  Case 6 'Triangular
    UserForm1.Label27.Caption = "Minimum"
    UserForm1.Label27.Visible = True
    UserForm1.Label28.Caption = "Mode"
    UserForm1.Label28.Visible = True
    UserForm1.Label29.Caption = "Maximum"
    UserForm1.Label29.Visible = True
    UserForm1.TextBox19.Visible = True
    UserForm1.TextBox20.Visible = True
    UserForm1.TextBox21.Visible = True
    UserForm1.CommandButton4.Visible = True
End Select

End Sub
```

Figura 43 - Excerto de código de seleção de distribuição de tempo

### 3.5.6. Controlo dos restantes módulos – Botão *Change*

O botão de *Change*, que vai ficando disponível conforme o utilizador seleciona uma distribuição, faz entrar em ação o algoritmo de alto nível apresentado na Figura 44. Na Figura 45 é apresentado o algoritmo onde é feita a validação dos valores a carregar para os parâmetros de acordo com a distribuição escolhida, e em caso de omissão, ou valores trocados é devolvida mensagem ao utilizador para verificar ou corrigir o valor.

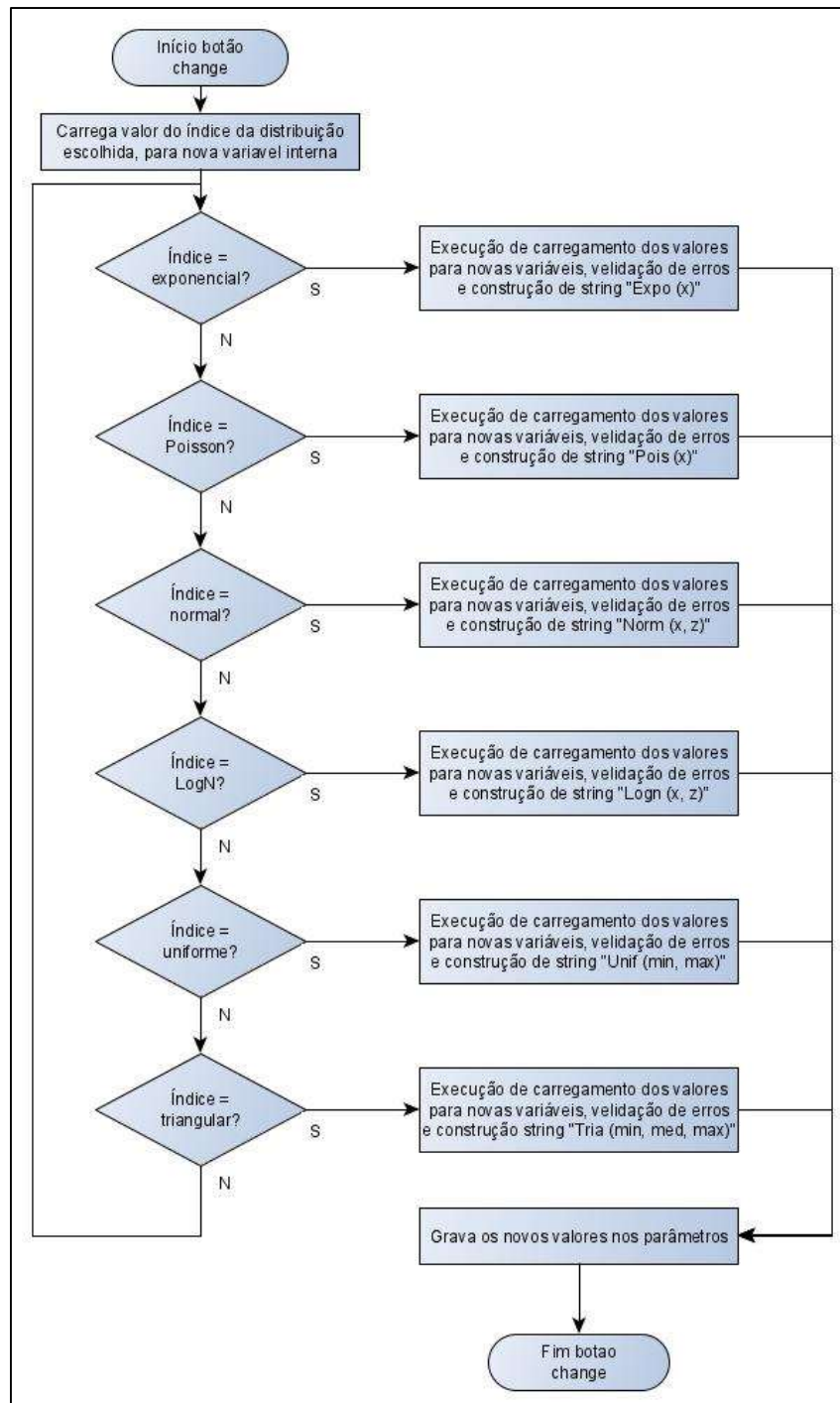


Figura 44 - Fluxograma alto nível do botão *change* <sup>12</sup>

<sup>12</sup> Fluxograma criado com o *software* yEd Version 3.19.1.1

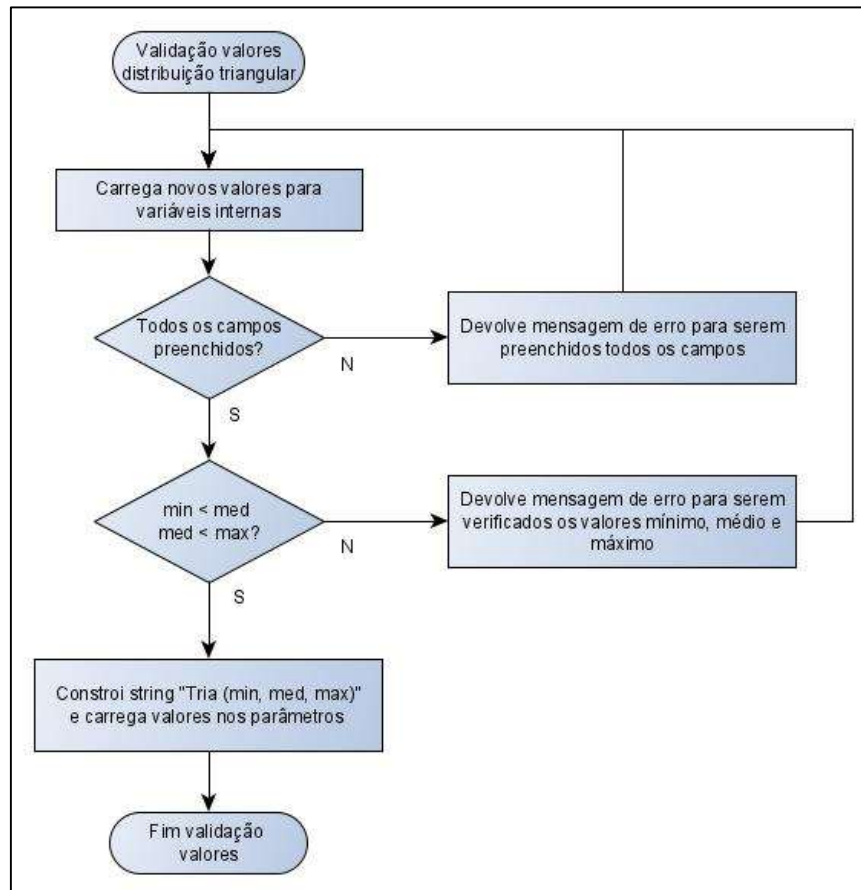


Figura 45 - Fluxograma da validação dos valores <sup>13</sup>

Na Figura 46, apresenta-se o código de gravação dos valores introduzidos nos campos pelo utilizador, executando validações e no caso de existirem erros de preenchimento, devolve mensagens com o erro respetivo.

```

Private Sub CommandButton11_Click()

Dim m As Model
Dim s As SIMAN

Set m = ThisDocument.Model
Set s = m.SIMAN

Var_Combobox4Index = UserForm1.Combobox4.ListIndex

Select Case Var_Combobox4Index
    Case 1 'Selecionada distribuição Exponencial
        Var_TempTextBox47 = Val(UserForm1.TextBox47.value) 'capacidade da recepção
        Var_TempTextBox48 = Val(UserForm1.TextBox48.value) 'Média da distribuição escolhida
        If Var_TempTextBox47 <> 0 And Var_TempTextBox48 <> 0 Then
            Var_TempConstructRecep = "Expo" & "(" & Var_TempTextBox48 & ")"
            s.ResourceCapacity(s.SymbolNumber("ResRepcion")) = Var_TempTextBox47
            s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Repcion")) = Var_TempConstructRecep
        Else
            MsgBox "Please insert values in all text boxes"
        End If
    End Select
  
```

<sup>13</sup> Fluxograma criado com o software yEd Version 3.19.1.1

```

Case 2      'Selecionada distribuição de Poisson
Var_TempTextBox47 = Val(UserForm1.TextBox47.value) 'Capacidade da recepção
Var_TempTextBox48 = Val(UserForm1.TextBox48.value) 'Média da distribuição escolhida
If Var_TempTextBox47 <> 0 And Var_TempTextBox48 <> 0 Then
    Var_TempConstructRecep = "Pois" & "(" & Var_TempTextBox48 & ")"
    s.ResourceCapacity(s.SymbolNumber("ResRecepcion")) = Var_TempTextBox47
    s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Recepcion")) =
Var_TempConstructRecep
Else
    MsgBox "Please insert values in all text boxes"
End If
Case 3      'Selecionada distribuição Normal
Var_TempTextBox47 = Val(UserForm1.TextBox47.value) 'Capacidade da recepção
Var_TempTextBox48 = Val(UserForm1.TextBox48.value) 'Média da distribuição escolhida
Var_TempTextBox49 = Val(UserForm1.TextBox49.value) 'Desvio standard
'Valida existência de valores e em caso afirmativo procede à alteração
If Var_TempTextBox47 <> 0 And Var_TempTextBox48 <> 0 And Var_TempTextBox49 <> 0 Then
    Var_TempConstructRecep = "Norm" & "(" & Var_TempTextBox48 & "," &
Var_TempTextBox49 & ")"
    s.ResourceCapacity(s.SymbolNumber("ResRecepcion")) = Var_TempTextBox47
    s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Recepcion")) =
Var_TempConstructRecep
Else
    MsgBox "Please insert values in all text boxes"
End If
Case 4      'Selecionada distribuição LogNormal
Var_TempTextBox47 = Val(UserForm1.TextBox47.value) 'Quantidade de clientes
Var_TempTextBox48 = Val(UserForm1.TextBox48.value) 'Média da distribuição escolhida
Var_TempTextBox49 = Val(UserForm1.TextBox49.value) 'Desvio standard
If Var_TempTextBox47 <> 0 And Var_TempTextBox48 <> 0 And Var_TempTextBox49 <> 0 Then
    Var_TempConstructRecep = "Logn" & "(" & Var_TempTextBox48 & "," &
Var_TempTextBox49 & ")"
    s.ResourceCapacity(s.SymbolNumber("ResRecepcion")) = Var_TempTextBox47
    s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Recepcion")) =
Var_TempConstructRecep
Else
    MsgBox "Please insert values in all text boxes"
End If
Case 5      'Selecionada distribuição Uniform
Var_TempTextBox47 = Val(UserForm1.TextBox47.value) 'Quantidade de clientes
Var_TempTextBox48 = Val(UserForm1.TextBox48.value) 'Minimum
Var_TempTextBox49 = Val(UserForm1.TextBox49.value) 'Maximum
If Var_TempTextBox47 <> 0 And Var_TempTextBox48 <> 0 And Var_TempTextBox49 <> 0 Then
    If (Var_TempTextBox48 < Var_TempTextBox49) Then
        Var_TempConstructRecep = "Unif" & "(" & Var_TempTextBox48 & "," &
Var_TempTextBox49 & ")"
        s.ResourceCapacity(s.SymbolNumber("ResRecepcion")) = Var_TempTextBox47
        s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Recepcion")) =
Var_TempConstructRecep
    Else
        MsgBox "Minimum value should be inferior than maximum"
    End If
Else
    MsgBox "Please insert values in all text boxes"
End If
Case 6      'Selecionada distribuição Triangular
Var_TempTextBox47 = Val(UserForm1.TextBox47.value) 'Quantidade de clientes
Var_TempTextBox48 = Val(UserForm1.TextBox48.value) 'Minimum
Var_TempTextBox49 = Val(UserForm1.TextBox49.value) 'Mode
Var_TempTextBox50 = Val(UserForm1.TextBox50.value) 'Maximum
If Var_TempTextBox47 <> 0 And Var_TempTextBox48 <> 0 And Var_TempTextBox49 <> 0 And
Var_TempTextBox50 <> 0 Then
    If (Var_TempTextBox48 < Var_TempTextBox49) And (Var_TempTextBox49 <
Var_TempTextBox50) Then
        Var_TempConstructRecep = "Tria" & "(" & Var_TempTextBox48 & "," &
Var_TempTextBox49 & "," & Var_TempTextBox50 & ")"
        s.ResourceCapacity(s.SymbolNumber("ResRecepcion")) = Var_TempTextBox47

```

```

        s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Recepcion")) =
Var_TempConstructRecep
        Else
            MsgBox "Please review Minimum, Mode and Maximum values"
        End If
    Else
        MsgBox "Please insert values in all text boxes"
    End If
Case Else
    MsgBox "Please confirm distribution selection"
End Select
End Sub

```

*Figura 46 - Código para gravação dos novos valores para os parâmetros*

De forma a simplificar as validações e erros de introdução de valores, todos os campos de entrada digitada são apenas de valores numéricos e apenas permitem a digitação de números e ponto decimal, todos os restantes caracteres não são inscritos, dando origem a sinal sonoro. Apresenta-se na Figura 47 o código subjacente a todos os campos de entrada de dados em causa.

```

Private Sub TextBox49_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

Select Case KeyAscii
    Case vbKey0 To vbKey9
    Case vbKeyBack, vbKeyClear, vbKeyDelete
    Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
    Case Else
        KeyAscii = 0
        Beep
    End Select
End Sub

```

*Figura 47 - Código para os campos de introdução de valores*

## 4. CONCLUSÕES E TRABALHO FUTURO

A presente dissertação foi desenvolvida no âmbito do curso de Mestrado em Engenharia e Gestão Industrial, ministrado no Instituto Superior de Engenharia do Porto, apresentando-se neste tópico, o resultado atingido no subtópico 4.1. No subtópico 4.2 as conclusões retiradas deste trabalho, o contributo deste trabalho no subtópico 4.3, e no subtópico 4.4 limitações encontradas e perspectivas de trabalhos a desenvolver.

### 4.1. Resultado atingido

Conforme indicado em 1.2., o resultado principal deste trabalho, é a criação de uma ferramenta baseada num simulador que suporte a tomada de decisão para uma oficina automóvel. De acordo com o pretendido, o simulador desenvolvido inicia com uma interface de utilizador que traduz as necessidades das entradas para o modelo, em variáveis mais correntes que permitem a sua utilização por pessoas sem grandes conhecimentos de programação de modelos/simuladores.

Ainda com o intuito de facilidade de utilização, foi mantida a coerência de apresentação e funcionamento entre módulos e adicionalmente criadas estratégias que podem ser facilmente aplicadas e alteradas, potenciando a criação de cenários de acordo com as necessidades do utilizador.

Para permitir maior facilidade de visualização da organização de uma oficina, foi criada uma planta de uma potencial oficina e sobrepôs-se a animação das viaturas, programando-se a deslocação das mesmas entre estacionamentos e postos de trabalho, onde é executado o serviço programado. A atribuição dos estacionamentos e postos de trabalho é feita aleatoriamente. Ainda com a animação é possível observar se há alguma espécie de estrangulamento físico à movimentação de viaturas.

Adicionalmente aos resultados acima referidos, é de salientar que os processos base (receção, encaminhamento, processamento/execução) são facilmente adaptáveis para aplicação a outros setores de serviços.

### 4.2. Conclusões finais

Com o presente trabalho foi desenvolvido uma ferramenta de simulação para suporte à tomada de decisão de uma oficina automóvel. A ferramenta foi desenvolvida recorrendo ao *software* Arena, sendo constituída por dois grandes blocos, o modelo lógico que contém a própria lógica de funcionamento e os vários componentes necessários para suporte à animação, e a interface do utilizador que pretende ocultar a complexidade da programação de modelos e simuladores do utilizador convencional. Adicionalmente foram pré-programadas três estratégias, sendo desta forma possível executar o simulador de imediato e obter resultados, ou caso o utilizador pretenda, carregar os parâmetros de qualquer uma das estratégias e seguidamente alterar os valores no módulo respetivo, criando assim cenários cujos resultados poderão ser comparados entre si.

Por experiência própria do autor deste trabalho (adquirida no âmbito deste Mestrado), constata-se que softwares mais atuais (comparando o *software* Arena ao *software* Delmia Quest), são mais fáceis de trabalhar, conseguindo-se com maior facilidade criar modelos e efetuar simulações muito rapidamente.

Não obstante a facilidade de utilização, mantém-se a necessidade de conhecimentos específicos na área da simulação, como por exemplo a utilização de nomes genéricos como entidades e recursos, a criação de rotas para as entidades seguirem, distribuições de tempo e forma de as parametrizar, que representam o mundo real, mas tem de ser corretamente programadas no sistema computacional.

Adicionalmente, a necessidade de conhecimentos na área do que se pretende representar, é essencial, por forma a validar e confirmar a correta representação num modelo computacional de um sistema real, ou seja, um especialista em softwares de simulação, poderá ter dificuldade em representar corretamente um sistema real se não tiver ajudas.

Em suma, *softwares* de simulação como o Arena, Delmia Quest, Any Logic e outros, ainda não se podem considerar de utilização geral. Provavelmente em futuras versões dos softwares acima referidos e outros ainda, a facilidade de utilização irá aumentar. Adicionalmente para a criação de ferramentas de simulação, serão ainda necessárias equipas com várias valências, por forma a garantir o correto funcionamento.

### 4.3. Caracter inovador do trabalho

Verificou-se na literatura da especialidade, a ausência de ferramentas de simulação para muitos setores de atividade, como no caso deste trabalho que foi orientado a uma oficina automóvel. A maior parte dos serviços de utilização corrente, utilizam três processos básicos, nomeadamente a receção, o encaminhamento e por sua vez a execução do pedido. Esta ferramenta implementa esses processos básicos, sendo facilmente adaptável a outras atividades.

A animação criada permite a visualização do estado da oficina a cada momento, permitindo desta forma ajudar à interpretação dos resultados obtidos.

Adicionalmente todos os valores passíveis de parametrização estão disponíveis para seleção e alteração por parte do utilizador, retirando a necessidade de conhecimentos específicos de modelação e simulação sendo, portanto, de fácil utilização. Desta forma este trabalho contribui para a divulgação de sistemas computacionais de simulação para o suporte à tomada de decisão.

### 4.4. Limitações e trabalho futuro

Durante a realização deste trabalho, a maior dificuldade esteve relacionada com a escassez de informação sobre a forma de como o *software* Arena recebe, ou entrega as variáveis aos módulos da interface do utilizador em *Visual Basic*, ou seja, a praticamente inexistente informação sobre os objetos do SIMAN (linguagem interna do Arena) e os métodos para aceder ou alterar valores das propriedades dos mesmos.

Na vertente de modelação e simulação, seria de considerar para trabalhos futuros, a adição da parametrização que permite valorizar os serviços efetuados, ou seja, acrescentar informação de tempo *Value Added* ou *Non Value Added*. Permitir a possibilidade de uma viatura poder ter mais de um serviço contratado, ou seja, na eventualidade de ter sido programado uma troca de óleo, também pode precisar de trocar uma lâmpada (serviço rápido) e, portanto, manter-se no modelo para uma nova iteração, eventualmente sem ser necessário voltar à receção. Eventualmente para as viaturas que seguem para as reparações gerais, isto já não se aplicaria. Em termos de apresentação de resultados, será de considerar a introdução de *Key Performance Indicators* (KPIs), por forma a facilitar a leitura dos resultados das execuções das simulações e ainda nesta vertente, a criação de relatórios com formato pré-definidos, ajudando ao mesmo objetivo de leitura e compreensão dos resultados obtidos.

Numa vertente de utilizador da ferramenta, será de considerar a adição da possibilidade de gravar os cenários criados, e a possibilidade de carregar cenários anteriormente gravados.



## REFERÊNCIAS BIBLIOGRÁFICAS

- Abreu, F. F. T. de, Costa, L. F., & Brinca, P. (2019). What has been driving the fluctuations in the portuguese economy? A business-cycle accounting approach. Obtido de <http://hdl.handle.net/10400.5/16571>
- Almeida, L. S., & Freire, T. (2007). *Metodologia da Investigação em Psicologia e Educação*. Braga: Psiquilíbrios Edições.
- Banks, J. (2000). Introduction to simulation. Em *2000 Winter Simulation Conference Proceedings (Cat. No.00CH37165)* (Vol. 1, pp. 9–16). IEEE. <https://doi.org/10.1109/WSC.2000.899690>
- Borshchev, A., & Filippov, A. (2004). From system dynamics and discrete event to practical agent based modeling: reasons, techniques, tools. *Proceedings of the 22nd international conference of the system dynamics society, Vol. 22*.
- Carson, J. S. (2005). Introduction to modeling and simulation. Em *Proceedings of the Winter Simulation Conference, 2005*. (pp. 8--pp).
- Castane, G. G., Simonis, H., Brown, K. N., Lin, Y., Ozturk, C., Garraffa, M., & Antunes, M. (2019). Simulation-Based Optimization Tool for Field Service Planning. Em *Proceedings - Winter Simulation Conference* (Vol. 2019-Decem, pp. 1684–1695). Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/WSC40007.2019.9004869>
- DOS SANTOS, M. (2010). O uso da Programação Linear Inteira (PLI) no Apoio à Decisão e a Otimização do Mix de Produção. *Anais do Encontro Nacional de Engenharia de Produção: ABEPRO*.
- Ferreira, L., Gómez, E., Lourido, G., Quintas, J., & Tjahjono, B. (2012). Analysis and optimisation of a network of closed-loop automobile assembly line using simulation. *International Journal of Advanced Manufacturing Technology*, 59(1–4), 351–366. <https://doi.org/10.1007/s00170-011-3502-4>
- Gil, A. C. (2008). *Métodos e técnicas de pesquisa social* (6 ed.). São Paulo: Editora Atlas S.A.
- Jahangirian, M., Eldabi, T., Naseer, A., Stergioulas, L. K., & Young, T. (2010). Simulation in manufacturing and business: A review. *European Journal of Operational Research*, 203(1), 1–13. <https://doi.org/10.1016/J.EJOR.2009.06.004>
- Johnson, M. E., & Poorte, J. P. (1988). A hierarchical approach to computer animation in simulation modeling. *SIMULATION*, 50(1), 30–36. <https://doi.org/10.1177/003754978805000105>
- Kelton, W. D., Sadowski, R. P., & Zupick, N. B. (2015). *Simulation with Arena* (6th ed.). New York: McGraw-Hill Education.
- Law, A M. (2008). How to build valid and credible simulation models. *2008 Winter Simulation Conference, Simulation Conference, 2008. WSC 2008. Winter*. <https://doi.org/10.1109/WSC.2008.4736054>
- Law, Averill M. (2019). How to Build Valid and Credible Simulation Models. Em *Proceedings - Winter Simulation Conference* (Vol. 2019-Decem, pp. 1402–1414). Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/WSC40007.2019.9004789>
- Marconi, M. de A., & Lakatos, E. M. (2003). *Fundamentos da metodologia científica* (5 ed.). São Paulo: Editora Atlas S.A.
- Mourtzis, D., Doukas, M., & Bernidaki, D. (2014). Simulation in manufacturing: Review and challenges. Em *Procedia CIRP* (Vol. 25, pp. 213–229). Elsevier B.V.

- <https://doi.org/10.1016/j.procir.2014.10.032>
- Ramis, F. J., Palma, J. L., & Baesler, F. F. (2001). The Use of Simulation for Process Improvement at an Ambulatory Surgery Center. Em *Proceedings of the 33Nd Conference on Winter Simulation* (pp. 1401–1404). Washington, DC, USA: IEEE Computer Society. Obtido de <http://dl.acm.org/citation.cfm?id=564124.564328>
- Robinson, S., Radnor, Z. J., Burgess, N., & Worthington, C. (2012). SimLean: Utilising simulation in the implementation of lean in healthcare. *European Journal of Operational Research*, 219(1), 188–197. <https://doi.org/10.1016/J.EJOR.2011.12.029>
- Rossetti, M. D. (2016). *Simulation modeling and Arena* (2nd ed.). Hoboken: John Wiley & Sons Inc.
- Sakurada, N., & Miyake, D. I. (2009). Aplicação de simuladores de eventos discretos no processo de modelagem de sistemas de operações de serviços. *Gestao e Producao*, 16(1), 25–43. <https://doi.org/10.1590/s0104-530x2009000100004>
- Shannon, R. E. (1998). Introduction to the Art and Science of Simulation. Em *Proceedings of the 30th Conference on Winter Simulation* (pp. 7–14). Los Alamitos, CA, USA: IEEE Computer Society Press. Obtido de <http://dl.acm.org/citation.cfm?id=293172.293175>
- Silva, S. B., Caulliraux, H. M., & Spiegel, T. (2015). Modelo de simulação para auxílio em planejamento e gestão de serviços de saúde. *Sistemas & Gestão 10 (2015)*, 608–616.
- Souza, A. C., Tosoni, L. A., Canassa, R. C., & Morais, M. D. F. (2015). Estudo de filas em caixas rápidos de um supermercado utilizando simulação computacional. *Revista Latino-Americana de Inovação e Engenharia de Produção*, 3(4), 203. <https://doi.org/10.5380/relainep.v3i4.43619>
- Tan, W. S., Chua, S. L., Yong, K. W., & Wu, T. S. (2009). Impact of pharmacy automation on patient waiting time: an application of computer simulation. *Annals Academy of Medicine Singapore*, 38(6), 501.
- Wagner, G. (2016). Introduction to simulation using JavaScript. Em *2016 Winter Simulation Conference (WSC)* (pp. 148–162). <https://doi.org/10.1109/WSC.2016.7822086>
- White, K. P., & Ingalls, R. G. (2018). THE BASICS OF SIMULATION. Em *2018 Winter Simulation Conference (WSC)* (pp. 147–161). <https://doi.org/10.1109/WSC.2018.8632271>
- Williams, M. R., & Williams, M. R. (1993). The origins, uses, and fate of the edvac. *IEEE ANN. HIST. COMPUT.* Obtido de <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.705.4726>





## APÊNDICE A – CÓDIGO DESENVOLVIDO

```
Private Sub ModelLogic_RunBeginSimulation()  
  
Dim m As Model  
Dim s As SIMAN  
  
Set m = ThisDocument.Model  
Set s = m.SIMAN  
  
UserForm1.MultiPage1.value = 0  
UserForm1.Show  
  
End Sub  
  
Private Sub ComboBox1_Change()  
  
Var_Combobox1Alterada = UserForm1.ComboBox1.ListIndex  
  
Select Case Var_Combobox1Alterada  
    Case 1  
        UserForm1.Label28.Visible = False  
        UserForm1.Label29.Visible = False  
        UserForm1.TextBox20.Visible = False  
        UserForm1.TextBox21.Visible = False  
        UserForm1.Label27.Caption = "Mean"  
        UserForm1.Label27.Visible = True  
        UserForm1.TextBox19.Visible = True  
        UserForm1.CommandButton4.Visible = True  
    Case 2  
        UserForm1.Label28.Visible = False  
        UserForm1.Label29.Visible = False  
        UserForm1.TextBox20.Visible = False  
        UserForm1.TextBox21.Visible = False  
        UserForm1.Label27.Caption = "Mean"  
        UserForm1.Label27.Visible = True  
        UserForm1.TextBox19.Visible = True  
        UserForm1.CommandButton4.Visible = True  
    Case 3  
        UserForm1.Label29.Visible = False  
        UserForm1.TextBox21.Visible = False  
        UserForm1.Label27.Caption = "Mean"  
        UserForm1.Label27.Visible = True  
        UserForm1.Label28.Caption = "StdDev"  
        UserForm1.Label28.Visible = True  
        UserForm1.TextBox19.Visible = True  
        UserForm1.TextBox20.Visible = True  
        UserForm1.CommandButton4.Visible = True  
    Case 4  
        UserForm1.Label29.Visible = False  
        UserForm1.TextBox21.Visible = False  
        UserForm1.Label27.Caption = "LogMean"  
        UserForm1.Label27.Visible = True  
        UserForm1.Label28.Caption = "LogStd"  
        UserForm1.Label28.Visible = True  
        UserForm1.TextBox19.Visible = True  
        UserForm1.TextBox20.Visible = True  
        UserForm1.CommandButton4.Visible = True  
    Case 5  
        UserForm1.Label29.Visible = False  
        UserForm1.TextBox21.Visible = False
```

```

    UserForm1.Label27.Caption = "Minimum"
    UserForm1.Label27.Visible = True
    UserForm1.Label28.Caption = "Maximum"
    UserForm1.Label28.Visible = True
    UserForm1.TextBox19.Visible = True
    UserForm1.TextBox20.Visible = True
    UserForm1.CommandButton4.Visible = True
Case 6
    UserForm1.Label27.Caption = "Minimum"
    UserForm1.Label27.Visible = True
    UserForm1.Label28.Caption = "Mode"
    UserForm1.Label28.Visible = True
    UserForm1.Label29.Caption = "Maximum"
    UserForm1.Label29.Visible = True
    UserForm1.TextBox19.Visible = True
    UserForm1.TextBox20.Visible = True
    UserForm1.TextBox21.Visible = True
    UserForm1.CommandButton4.Visible = True
End Select

End Sub

Private Sub ComboBox10_Change()

Var_Combobox10Alterada = UserForm1.ComboBox10.ListIndex

Select Case Var_Combobox10Alterada
    Case 1
        UserForm1.Label144.Visible = False
        UserForm1.Label143.Visible = False
        UserForm1.TextBox94.Visible = False
        UserForm1.TextBox93.Visible = False
        UserForm1.Label142.Caption = "Mean"
        UserForm1.Label142.Visible = True
        UserForm1.TextBox95.Visible = True
        UserForm1.CommandButton14.Visible = True
    Case 2
        UserForm1.Label144.Visible = False
        UserForm1.Label143.Visible = False
        UserForm1.TextBox94.Visible = False
        UserForm1.TextBox93.Visible = False
        UserForm1.Label142.Caption = "Mean"
        UserForm1.Label142.Visible = True
        UserForm1.TextBox95.Visible = True
        UserForm1.CommandButton14.Visible = True
    Case 3
        UserForm1.Label143.Visible = False
        UserForm1.TextBox93.Visible = False
        UserForm1.Label142.Caption = "Mean"
        UserForm1.Label142.Visible = True
        UserForm1.Label144.Caption = "StdDev"
        UserForm1.Label144.Visible = True
        UserForm1.TextBox95.Visible = True
        UserForm1.TextBox94.Visible = True
        UserForm1.CommandButton14.Visible = True
    Case 4
        UserForm1.Label143.Visible = False
        UserForm1.TextBox93.Visible = False
        UserForm1.Label142.Caption = "LogMean"
        UserForm1.Label142.Visible = True
        UserForm1.Label144.Caption = "LogStd"
        UserForm1.Label144.Visible = True
        UserForm1.TextBox95.Visible = True
        UserForm1.TextBox94.Visible = True
        UserForm1.CommandButton14.Visible = True
    Case 5
        UserForm1.Label143.Visible = False

```

```
UserForm1.TextBox93.Visible = False
UserForm1.Label142.Caption = "Minimum"
UserForm1.Label142.Visible = True
UserForm1.Label144.Caption = "Maximum"
UserForm1.Label144.Visible = True
UserForm1.TextBox95.Visible = True
UserForm1.TextBox94.Visible = True
UserForm1.CommandButton14.Visible = True
Case 6
UserForm1.Label142.Caption = "Minimum"
UserForm1.Label142.Visible = True
UserForm1.Label144.Caption = "Mode"
UserForm1.Label144.Visible = True
UserForm1.Label143.Caption = "Maximum"
UserForm1.Label143.Visible = True
UserForm1.TextBox95.Visible = True
UserForm1.TextBox94.Visible = True
UserForm1.TextBox93.Visible = True
UserForm1.CommandButton14.Visible = True
End Select

End Sub

Private Sub ComboBox11_Change()

Var_Combobox11Alterada = UserForm1.ComboBox11.ListIndex

Select Case Var_Combobox11Alterada
Case 1
UserForm1.Label150.Visible = False
UserForm1.Label149.Visible = False
UserForm1.TextBox97.Visible = False
UserForm1.TextBox84.Visible = False
UserForm1.Label148.Caption = "Mean"
UserForm1.Label148.Visible = True
UserForm1.TextBox98.Visible = True
UserForm1.CommandButton15.Visible = True
Case 2
UserForm1.Label150.Visible = False
UserForm1.Label149.Visible = False
UserForm1.TextBox97.Visible = False
UserForm1.TextBox84.Visible = False
UserForm1.Label148.Caption = "Mean"
UserForm1.Label148.Visible = True
UserForm1.TextBox98.Visible = True
UserForm1.CommandButton15.Visible = True
Case 3
UserForm1.Label149.Visible = False
UserForm1.TextBox84.Visible = False
UserForm1.Label148.Caption = "Mean"
UserForm1.Label148.Visible = True
UserForm1.Label150.Caption = "StdDev"
UserForm1.Label150.Visible = True
UserForm1.TextBox98.Visible = True
UserForm1.TextBox97.Visible = True
UserForm1.CommandButton15.Visible = True
Case 4
UserForm1.Label149.Visible = False
UserForm1.TextBox84.Visible = False
UserForm1.Label148.Caption = "LogMean"
UserForm1.Label148.Visible = True
UserForm1.Label150.Caption = "LogStd"
UserForm1.Label150.Visible = True
UserForm1.TextBox98.Visible = True
UserForm1.TextBox97.Visible = True
UserForm1.CommandButton15.Visible = True
```

```
Case 5
    UserForm1.Label149.Visible = False
    UserForm1.TextBox84.Visible = False
    UserForm1.Label148.Caption = "Minimum"
    UserForm1.Label148.Visible = True
    UserForm1.Label150.Caption = "Maximum"
    UserForm1.Label150.Visible = True
    UserForm1.TextBox98.Visible = True
    UserForm1.TextBox97.Visible = True
    UserForm1.CommandButton15.Visible = True
Case 6
    UserForm1.Label148.Caption = "Minimum"
    UserForm1.Label148.Visible = True
    UserForm1.Label150.Caption = "Mode"
    UserForm1.Label150.Visible = True
    UserForm1.Label149.Caption = "Maximum"
    UserForm1.Label149.Visible = True
    UserForm1.TextBox98.Visible = True
    UserForm1.TextBox97.Visible = True
    UserForm1.TextBox84.Visible = True
    UserForm1.CommandButton15.Visible = True
End Select

End Sub

Private Sub ComboBox2_Change()

Var_Combobox2Alterada = UserForm1.ComboBox2.ListIndex

Select Case Var_Combobox2Alterada
    Case 1
        UserForm1.Label31.Visible = False
        UserForm1.Label30.Visible = False
        UserForm1.TextBox23.Visible = False
        UserForm1.TextBox22.Visible = False
        UserForm1.Label32.Caption = "Mean"
        UserForm1.Label32.Visible = True
        UserForm1.TextBox24.Visible = True
        UserForm1.CommandButton5.Visible = True
    Case 2
        UserForm1.Label31.Visible = False
        UserForm1.Label30.Visible = False
        UserForm1.TextBox23.Visible = False
        UserForm1.TextBox22.Visible = False
        UserForm1.Label32.Caption = "Mean"
        UserForm1.Label32.Visible = True
        UserForm1.TextBox24.Visible = True
        UserForm1.CommandButton5.Visible = True
    Case 3
        UserForm1.Label30.Visible = False
        UserForm1.TextBox22.Visible = False
        UserForm1.Label32.Caption = "Mean"
        UserForm1.Label32.Visible = True
        UserForm1.Label31.Caption = "StdDev"
        UserForm1.Label31.Visible = True
        UserForm1.TextBox24.Visible = True
        UserForm1.TextBox23.Visible = True
        UserForm1.CommandButton5.Visible = True
    Case 4
        UserForm1.Label30.Visible = False
        UserForm1.TextBox22.Visible = False
        UserForm1.Label32.Caption = "LogMean"
        UserForm1.Label32.Visible = True
        UserForm1.Label31.Caption = "LogStd"
        UserForm1.Label31.Visible = True
        UserForm1.TextBox24.Visible = True
        UserForm1.TextBox23.Visible = True
```

```
UserForm1.CommandButton5.Visible = True
Case 5
UserForm1.Label30.Visible = False
UserForm1.TextBox22.Visible = False
UserForm1.Label32.Caption = "Minimum"
UserForm1.Label32.Visible = True
UserForm1.Label31.Caption = "Maximum"
UserForm1.Label31.Visible = True
UserForm1.TextBox24.Visible = True
UserForm1.TextBox23.Visible = True
UserForm1.CommandButton5.Visible = True
Case 6
UserForm1.Label32.Caption = "Minimum"
UserForm1.Label32.Visible = True
UserForm1.Label31.Caption = "Mode"
UserForm1.Label31.Visible = True
UserForm1.Label30.Caption = "Maximum"
UserForm1.Label30.Visible = True
UserForm1.TextBox24.Visible = True
UserForm1.TextBox23.Visible = True
UserForm1.TextBox22.Visible = True
UserForm1.CommandButton5.Visible = True
End Select

End Sub

Private Sub ComboBox3_Change()

Var_Combobox3Alterada = UserForm1.ComboBox3.ListIndex

Select Case Var_Combobox3Alterada
Case 1
UserForm1.Label36.Visible = False
UserForm1.Label35.Visible = False
UserForm1.TextBox27.Visible = False
UserForm1.TextBox16.Visible = False
UserForm1.Label37.Caption = "Mean"
UserForm1.Label37.Visible = True
UserForm1.TextBox28.Visible = True
UserForm1.CommandButton6.Visible = True
Case 2
UserForm1.Label36.Visible = False
UserForm1.Label35.Visible = False
UserForm1.TextBox27.Visible = False
UserForm1.TextBox16.Visible = False
UserForm1.Label37.Caption = "Mean"
UserForm1.Label37.Visible = True
UserForm1.TextBox28.Visible = True
UserForm1.CommandButton6.Visible = True
Case 3
UserForm1.Label35.Visible = False
UserForm1.TextBox16.Visible = False
UserForm1.Label37.Caption = "Mean"
UserForm1.Label37.Visible = True
UserForm1.Label36.Caption = "StdDev"
UserForm1.Label36.Visible = True
UserForm1.TextBox28.Visible = True
UserForm1.TextBox27.Visible = True
UserForm1.CommandButton6.Visible = True
Case 4
UserForm1.Label35.Visible = False
UserForm1.TextBox16.Visible = False
UserForm1.Label37.Caption = "LogMean"
UserForm1.Label37.Visible = True
UserForm1.Label36.Caption = "LogStd"
UserForm1.Label36.Visible = True
UserForm1.TextBox28.Visible = True
```

```

        UserForm1.TextBox27.Visible = True
        UserForm1.CommandButton6.Visible = True
    Case 5
        UserForm1.Label135.Visible = False
        UserForm1.TextBox16.Visible = False
        UserForm1.Label137.Caption = "Minimum"
        UserForm1.Label137.Visible = True
        UserForm1.Label136.Caption = "Maximum"
        UserForm1.Label136.Visible = True
        UserForm1.TextBox28.Visible = True
        UserForm1.TextBox27.Visible = True
        UserForm1.CommandButton6.Visible = True
    Case 6
        UserForm1.Label137.Caption = "Minimum"
        UserForm1.Label137.Visible = True
        UserForm1.Label136.Caption = "Mode"
        UserForm1.Label136.Visible = True
        UserForm1.Label135.Caption = "Maximum"
        UserForm1.Label135.Visible = True
        UserForm1.TextBox28.Visible = True
        UserForm1.TextBox27.Visible = True
        UserForm1.TextBox16.Visible = True
        UserForm1.CommandButton6.Visible = True
End Select

End Sub

Private Sub ComboBox4_Change()

Var_Combobox4Alterada = UserForm1.ComboBox4.ListIndex

Select Case Var_Combobox4Alterada
    Case 1
        UserForm1.Label172.Visible = False
        UserForm1.Label173.Visible = False
        UserForm1.TextBox49.Visible = False
        UserForm1.TextBox50.Visible = False
        UserForm1.Label171.Caption = "Mean"
        UserForm1.Label171.Visible = True
        UserForm1.TextBox48.Visible = True
        UserForm1.CommandButton11.Visible = True
    Case 2
        UserForm1.Label172.Visible = False
        UserForm1.Label173.Visible = False
        UserForm1.TextBox49.Visible = False
        UserForm1.TextBox50.Visible = False
        UserForm1.Label171.Caption = "Mean"
        UserForm1.Label171.Visible = True
        UserForm1.TextBox48.Visible = True
        UserForm1.CommandButton11.Visible = True
    Case 3
        UserForm1.Label173.Visible = False
        UserForm1.TextBox50.Visible = False
        UserForm1.Label171.Caption = "Mean"
        UserForm1.Label171.Visible = True
        UserForm1.Label172.Caption = "StdDev"
        UserForm1.Label172.Visible = True
        UserForm1.TextBox48.Visible = True
        UserForm1.TextBox49.Visible = True
        UserForm1.CommandButton11.Visible = True
    Case 4
        UserForm1.Label173.Visible = False
        UserForm1.TextBox50.Visible = False
        UserForm1.Label171.Caption = "LogMean"
        UserForm1.Label171.Visible = True
        UserForm1.Label172.Caption = "LogStd"
        UserForm1.Label172.Visible = True

```

```
UserForm1.TextBox48.Visible = True
UserForm1.TextBox49.Visible = True
UserForm1.CommandButton11.Visible = True
Case 5
UserForm1.Label73.Visible = False
UserForm1.TextBox50.Visible = False
UserForm1.Label71.Caption = "Minimum"
UserForm1.Label71.Visible = True
UserForm1.Label72.Caption = "Maximum"
UserForm1.Label72.Visible = True
UserForm1.TextBox48.Visible = True
UserForm1.TextBox49.Visible = True
UserForm1.CommandButton11.Visible = True
Case 6
UserForm1.Label71.Caption = "Minimum"
UserForm1.Label71.Visible = True
UserForm1.Label72.Caption = "Mode"
UserForm1.Label72.Visible = True
UserForm1.Label73.Caption = "Maximum"
UserForm1.Label73.Visible = True
UserForm1.TextBox48.Visible = True
UserForm1.TextBox49.Visible = True
UserForm1.TextBox50.Visible = True
UserForm1.CommandButton11.Visible = True
End Select

End Sub

Private Sub ComboBox5_Change()

Var_Combobox5Alterada = UserForm1.ComboBox5.ListIndex

Select Case Var_Combobox5Alterada
Case 1
UserForm1.Label75.Visible = False
UserForm1.Label74.Visible = False
UserForm1.TextBox52.Visible = False
UserForm1.TextBox51.Visible = False
UserForm1.Label76.Caption = "Mean"
UserForm1.Label76.Visible = True
UserForm1.TextBox53.Visible = True
UserForm1.CommandButton10.Visible = True
Case 2
UserForm1.Label75.Visible = False
UserForm1.Label74.Visible = False
UserForm1.TextBox52.Visible = False
UserForm1.TextBox51.Visible = False
UserForm1.Label76.Caption = "Mean"
UserForm1.Label76.Visible = True
UserForm1.TextBox53.Visible = True
UserForm1.CommandButton10.Visible = True
Case 3
UserForm1.Label74.Visible = False
UserForm1.TextBox51.Visible = False
UserForm1.Label76.Caption = "Mean"
UserForm1.Label76.Visible = True
UserForm1.Label75.Caption = "StdDev"
UserForm1.Label75.Visible = True
UserForm1.TextBox53.Visible = True
UserForm1.TextBox52.Visible = True
UserForm1.CommandButton10.Visible = True
Case 4
UserForm1.Label74.Visible = False
UserForm1.TextBox51.Visible = False
UserForm1.Label76.Caption = "LogMean"
UserForm1.Label76.Visible = True
UserForm1.Label75.Caption = "LogStd"
```

```

    UserForm1.Label175.Visible = True
    UserForm1.TextBox53.Visible = True
    UserForm1.TextBox52.Visible = True
    UserForm1.CommandButton10.Visible = True
Case 5
    UserForm1.Label174.Visible = False
    UserForm1.TextBox51.Visible = False
    UserForm1.Label176.Caption = "Minimum"
    UserForm1.Label176.Visible = True
    UserForm1.Label175.Caption = "Maximum"
    UserForm1.Label175.Visible = True
    UserForm1.TextBox53.Visible = True
    UserForm1.TextBox52.Visible = True
    UserForm1.CommandButton10.Visible = True
Case 6
    UserForm1.Label176.Caption = "Minimum"
    UserForm1.Label176.Visible = True
    UserForm1.Label175.Caption = "Mode"
    UserForm1.Label175.Visible = True
    UserForm1.Label174.Caption = "Maximum"
    UserForm1.Label174.Visible = True
    UserForm1.TextBox53.Visible = True
    UserForm1.TextBox52.Visible = True
    UserForm1.TextBox51.Visible = True
    UserForm1.CommandButton10.Visible = True
End Select

End Sub

Private Sub ComboBox6_Change()

Var_Combobox6Alterada = UserForm1.ComboBox6.ListIndex

Select Case Var_Combobox6Alterada
    Case 1
        UserForm1.Label180.Visible = False
        UserForm1.Label179.Visible = False
        UserForm1.TextBox55.Visible = False
        UserForm1.TextBox46.Visible = False
        UserForm1.Label181.Caption = "Mean"
        UserForm1.Label181.Visible = True
        UserForm1.TextBox56.Visible = True
        UserForm1.CommandButton9.Visible = True
    Case 2
        UserForm1.Label180.Visible = False
        UserForm1.Label179.Visible = False
        UserForm1.TextBox55.Visible = False
        UserForm1.TextBox46.Visible = False
        UserForm1.Label181.Caption = "Mean"
        UserForm1.Label181.Visible = True
        UserForm1.TextBox56.Visible = True
        UserForm1.CommandButton9.Visible = True
    Case 3
        UserForm1.Label179.Visible = False
        UserForm1.TextBox46.Visible = False
        UserForm1.Label181.Caption = "Mean"
        UserForm1.Label181.Visible = True
        UserForm1.Label180.Caption = "StdDev"
        UserForm1.Label180.Visible = True
        UserForm1.TextBox56.Visible = True
        UserForm1.TextBox55.Visible = True
        UserForm1.CommandButton9.Visible = True
    Case 4
        UserForm1.Label179.Visible = False
        UserForm1.TextBox46.Visible = False
        UserForm1.Label181.Caption = "LogMean"
        UserForm1.Label181.Visible = True

```

```
UserForm1.Label180.Caption = "LogStd"
UserForm1.Label180.Visible = True
UserForm1.TextBox56.Visible = True
UserForm1.TextBox55.Visible = True
UserForm1.CommandButton9.Visible = True
Case 5
UserForm1.Label79.Visible = False
UserForm1.TextBox46.Visible = False
UserForm1.Label81.Caption = "Minimum"
UserForm1.Label81.Visible = True
UserForm1.Label80.Caption = "Maximum"
UserForm1.Label80.Visible = True
UserForm1.TextBox56.Visible = True
UserForm1.TextBox55.Visible = True
UserForm1.CommandButton9.Visible = True
Case 6
UserForm1.Label81.Caption = "Minimum"
UserForm1.Label81.Visible = True
UserForm1.Label80.Caption = "Mode"
UserForm1.Label80.Visible = True
UserForm1.Label79.Caption = "Maximum"
UserForm1.Label79.Visible = True
UserForm1.TextBox56.Visible = True
UserForm1.TextBox55.Visible = True
UserForm1.TextBox46.Visible = True
UserForm1.CommandButton9.Visible = True
End Select

End Sub

Private Sub ComboBox7_Change()

Var_Combobox7Alterada = UserForm1.ComboBox7.ListIndex

Select Case Var_Combobox7Alterada
Case 1
UserForm1.Label197.Visible = False
UserForm1.Label196.Visible = False
UserForm1.TextBox64.Visible = False
UserForm1.TextBox63.Visible = False
UserForm1.Label198.Caption = "Mean"
UserForm1.Label198.Visible = True
UserForm1.TextBox65.Visible = True
UserForm1.CommandButton8.Visible = True
Case 2
UserForm1.Label197.Visible = False
UserForm1.Label196.Visible = False
UserForm1.TextBox64.Visible = False
UserForm1.TextBox63.Visible = False
UserForm1.Label198.Caption = "Mean"
UserForm1.Label198.Visible = True
UserForm1.TextBox65.Visible = True
UserForm1.CommandButton8.Visible = True
Case 3
UserForm1.Label196.Visible = False
UserForm1.TextBox63.Visible = False
UserForm1.Label198.Caption = "Mean"
UserForm1.Label198.Visible = True
UserForm1.Label197.Caption = "StdDev"
UserForm1.Label197.Visible = True
UserForm1.TextBox65.Visible = True
UserForm1.TextBox64.Visible = True
UserForm1.CommandButton8.Visible = True
Case 4
UserForm1.Label196.Visible = False
UserForm1.TextBox63.Visible = False
UserForm1.Label198.Caption = "LogMean"
```

```

    UserForm1.Label98.Visible = True
    UserForm1.Label97.Caption = "LogStd"
    UserForm1.Label97.Visible = True
    UserForm1.TextBox65.Visible = True
    UserForm1.TextBox64.Visible = True
    UserForm1.CommandButton8.Visible = True
Case 5
    UserForm1.Label96.Visible = False
    UserForm1.TextBox63.Visible = False
    UserForm1.Label98.Caption = "Minimum"
    UserForm1.Label98.Visible = True
    UserForm1.Label97.Caption = "Maximum"
    UserForm1.Label97.Visible = True
    UserForm1.TextBox65.Visible = True
    UserForm1.TextBox64.Visible = True
    UserForm1.CommandButton8.Visible = True
Case 6
    UserForm1.Label98.Caption = "Minimum"
    UserForm1.Label98.Visible = True
    UserForm1.Label97.Caption = "Mode"
    UserForm1.Label97.Visible = True
    UserForm1.Label96.Caption = "Maximum"
    UserForm1.Label96.Visible = True
    UserForm1.TextBox65.Visible = True
    UserForm1.TextBox64.Visible = True
    UserForm1.TextBox63.Visible = True
    UserForm1.CommandButton8.Visible = True
End Select

End Sub

Private Sub ComboBox8_Change()

Var_Combobox8Alterada = UserForm1.ComboBox8.ListIndex

Select Case Var_Combobox8Alterada
    Case 1
        UserForm1.Label133.Visible = False
        UserForm1.Label134.Visible = False
        UserForm1.TextBox87.Visible = False
        UserForm1.TextBox88.Visible = False
        UserForm1.Label132.Caption = "Mean"
        UserForm1.Label132.Visible = True
        UserForm1.TextBox86.Visible = True
        UserForm1.CommandButton12.Visible = True
    Case 2
        UserForm1.Label133.Visible = False
        UserForm1.Label134.Visible = False
        UserForm1.TextBox87.Visible = False
        UserForm1.TextBox88.Visible = False
        UserForm1.Label132.Caption = "Mean"
        UserForm1.Label132.Visible = True
        UserForm1.TextBox86.Visible = True
        UserForm1.CommandButton12.Visible = True
    Case 3
        UserForm1.Label134.Visible = False
        UserForm1.TextBox88.Visible = False
        UserForm1.Label132.Caption = "Mean"
        UserForm1.Label132.Visible = True
        UserForm1.Label133.Caption = "StdDev"
        UserForm1.Label133.Visible = True
        UserForm1.TextBox86.Visible = True
        UserForm1.TextBox87.Visible = True
        UserForm1.CommandButton12.Visible = True
    Case 4
        UserForm1.Label134.Visible = False
        UserForm1.TextBox88.Visible = False

```

```
UserForm1.Label132.Caption = "LogMean"
UserForm1.Label132.Visible = True
UserForm1.Label133.Caption = "LogStd"
UserForm1.Label133.Visible = True
UserForm1.TextBox86.Visible = True
UserForm1.TextBox87.Visible = True
UserForm1.CommandButton12.Visible = True
Case 5
UserForm1.Label134.Visible = False
UserForm1.TextBox88.Visible = False
UserForm1.Label132.Caption = "Minimum"
UserForm1.Label132.Visible = True
UserForm1.Label133.Caption = "Maximum"
UserForm1.Label133.Visible = True
UserForm1.TextBox86.Visible = True
UserForm1.TextBox87.Visible = True
UserForm1.CommandButton12.Visible = True
Case 6
UserForm1.Label132.Caption = "Minimum"
UserForm1.Label132.Visible = True
UserForm1.Label133.Caption = "Mode"
UserForm1.Label133.Visible = True
UserForm1.Label134.Caption = "Maximum"
UserForm1.Label134.Visible = True
UserForm1.TextBox86.Visible = True
UserForm1.TextBox87.Visible = True
UserForm1.TextBox88.Visible = True
UserForm1.CommandButton12.Visible = True
End Select

End Sub

Private Sub ComboBox9_Change()

Var_Combobox9Alterada = UserForm1.ComboBox9.ListIndex

Select Case Var_Combobox9Alterada
Case 1
UserForm1.Label138.Visible = False
UserForm1.Label137.Visible = False
UserForm1.TextBox90.Visible = False
UserForm1.TextBox89.Visible = False
UserForm1.Label136.Caption = "Mean"
UserForm1.Label136.Visible = True
UserForm1.TextBox91.Visible = True
UserForm1.CommandButton13.Visible = True
Case 2
UserForm1.Label138.Visible = False
UserForm1.Label137.Visible = False
UserForm1.TextBox90.Visible = False
UserForm1.TextBox89.Visible = False
UserForm1.Label136.Caption = "Mean"
UserForm1.Label136.Visible = True
UserForm1.TextBox91.Visible = True
UserForm1.CommandButton13.Visible = True
Case 3
UserForm1.Label137.Visible = False
UserForm1.TextBox89.Visible = False
UserForm1.Label136.Caption = "Mean"
UserForm1.Label136.Visible = True
UserForm1.Label138.Caption = "StdDev"
UserForm1.Label138.Visible = True
UserForm1.TextBox91.Visible = True
UserForm1.TextBox90.Visible = True
UserForm1.CommandButton13.Visible = True
Case 4
UserForm1.Label137.Visible = False
```

```

    UserForm1.TextBox89.Visible = False
    UserForm1.Label136.Caption = "LogMean"
    UserForm1.Label136.Visible = True
    UserForm1.Label138.Caption = "LogStd"
    UserForm1.Label138.Visible = True
    UserForm1.TextBox91.Visible = True
    UserForm1.TextBox90.Visible = True
    UserForm1.CommandButton13.Visible = True
Case 5
    UserForm1.Label137.Visible = False
    UserForm1.TextBox89.Visible = False
    UserForm1.Label136.Caption = "Minimum"
    UserForm1.Label136.Visible = True
    UserForm1.Label138.Caption = "Maximum"
    UserForm1.Label138.Visible = True
    UserForm1.TextBox91.Visible = True
    UserForm1.TextBox90.Visible = True
    UserForm1.CommandButton13.Visible = True
Case 6
    UserForm1.Label136.Caption = "Minimum"
    UserForm1.Label136.Visible = True
    UserForm1.Label138.Caption = "Mode"
    UserForm1.Label138.Visible = True
    UserForm1.Label137.Caption = "Maximum"
    UserForm1.Label137.Visible = True
    UserForm1.TextBox91.Visible = True
    UserForm1.TextBox90.Visible = True
    UserForm1.TextBox89.Visible = True
    UserForm1.CommandButton13.Visible = True
End Select

End Sub

Private Sub CommandButton1_Click()

    Dim m As Model
    Dim s As SIMAN

    Set m = ThisDocument.Model
    Set s = m.SIMAN

    m.Pause
    m.End

    UserForm1.Hide

End Sub

Private Sub CommandButton10_Click()

    Dim m As Model
    Dim s As SIMAN

    Set m = ThisDocument.Model
    Set s = m.SIMAN

    Var_Combobox5Index = UserForm1.Combobox5.ListIndex

    Select Case Var_Combobox5Index
        Case 1
            Var_TempTextBox54 = Val(UserForm1.TextBox54.value)
            var_temptextbox53 = Val(UserForm1.TextBox53.value)
            If Var_TempTextBox54 <> 0 And var_temptextbox53 <> 0 Then
                Var_TempConstructGR = "Expo" & "(" & var_temptextbox53 & ")"
                s.ResourceCapacity(s.SymbolNumber("RscGeneralRepairs")) = Var_TempTextBox54
                s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_GeneralRepairs")) =
            Var_TempConstructGR

```

```

Else
    MsgBox "Please insert values in all text boxes"
End If
Case 2
    Var_TempTextBox54 = Val(UserForm1.TextBox54.value)
    var_temptextbox53 = Val(UserForm1.TextBox53.value)
    If Var_TempTextBox54 <> 0 And var_temptextbox53 <> 0 Then
        Var_TempConstructGR = "Pois" & "(" & var_temptextbox53 & ")"
        s.ResourceCapacity(s.SymbolNumber("RscGeneralRepairs")) = Var_TempTextBox54
        s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_GeneralRepairs")) =
Var_TempConstructGR
    Else
        MsgBox "Please insert values in all text boxes"
    End If
Case 3
    Var_TempTextBox54 = Val(UserForm1.TextBox54.value)
    var_temptextbox53 = Val(UserForm1.TextBox53.value)
    var_temptextbox52 = Val(UserForm1.TextBox52.value)
    If Var_TempTextBox54 <> 0 And var_temptextbox53 <> 0 And var_temptextbox52 <> 0 Then
        Var_TempConstructGR = "Norm" & "(" & var_temptextbox53 & "," & var_temptextbox52
& ")"
        s.ResourceCapacity(s.SymbolNumber("RscGeneralRepairs")) = Var_TempTextBox54
        s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_GeneralRepairs")) =
Var_TempConstructGR
    Else
        MsgBox "Please insert values in all text boxes"
    End If
Case 4
    Var_TempTextBox54 = Val(UserForm1.TextBox54.value)
    var_temptextbox53 = Val(UserForm1.TextBox53.value)
    var_temptextbox52 = Val(UserForm1.TextBox52.value)
    If Var_TempTextBox54 <> 0 And var_temptextbox53 <> 0 And var_temptextbox52 <> 0 Then
        Var_TempConstructGR = "Logn" & "(" & var_temptextbox53 & "," & var_temptextbox52
& ")"
        s.ResourceCapacity(s.SymbolNumber("RscGeneralRepairs")) = Var_TempTextBox54
        s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_GeneralRepairs")) =
Var_TempConstructGR
    Else
        MsgBox "Please insert values in all text boxes"
    End If
Case 5
    Var_TempTextBox54 = Val(UserForm1.TextBox54.value)
    var_temptextbox53 = Val(UserForm1.TextBox53.value)
    var_temptextbox52 = Val(UserForm1.TextBox52.value)
    If Var_TempTextBox54 <> 0 And var_temptextbox53 <> 0 And var_temptextbox52 <> 0 Then
        If (var_temptextbox53 < var_temptextbox52) Then
            Var_TempConstructGR = "Unif" & "(" & var_temptextbox53 & "," &
var_temptextbox52 & ")"
            s.ResourceCapacity(s.SymbolNumber("RscGeneralRepairs")) = Var_TempTextBox54
            s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_GeneralRepairs"))
= Var_TempConstructGR
        Else
            MsgBox "Minimum value should be inferior than maximum"
        End If
    Else
        MsgBox "Please insert values in all text boxes"
    End If
Case 6
    Var_TempTextBox54 = Val(UserForm1.TextBox54.value)
    var_temptextbox53 = Val(UserForm1.TextBox53.value)
    var_temptextbox52 = Val(UserForm1.TextBox52.value)
    var_temptextbox51 = Val(UserForm1.TextBox51.value)
    If Var_TempTextBox54 <> 0 And var_temptextbox53 <> 0 And var_temptextbox52 <> 0 And
var_temptextbox51 <> 0 Then
        If (var_temptextbox53 < var_temptextbox52) And (var_temptextbox52 <
var_temptextbox51) Then

```

```

        Var_TempConstructGR = "Tria" & "(" & var_temptextbox53 & "," &
var_temptextbox52 & "," & var_temptextbox51 & ")"
        s.ResourceCapacity(s.SymbolNumber("RscGeneralRepairs")) = Var_TempTextBox54
        s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_GeneralRepairs"))
= Var_TempConstructGR
    Else
        MsgBox "Please review Minimum, Mode and Maximum values"
    End If
    Else
        MsgBox "Please insert values in all text boxes"
    End If
    Case Else
        MsgBox "Please confirm distribution selection"
    End Select
End Sub

Private Sub CommandButton11_Click()

Dim m As Model
Dim s As SIMAN

Set m = ThisDocument.Model
Set s = m.SIMAN

Var_Combobox4Index = UserForm1.Combobox4.ListIndex

Select Case Var_Combobox4Index
    Case 1
        Var_TempTextBox47 = Val(UserForm1.TextBox47.value)
        Var_TempTextBox48 = Val(UserForm1.TextBox48.value)
        If Var_TempTextBox47 <> 0 And Var_TempTextBox48 <> 0 Then
            Var_TempConstructRecep = "Expo" & "(" & Var_TempTextBox48 & ")"
            s.ResourceCapacity(s.SymbolNumber("ResRecepcion")) = Var_TempTextBox47
            s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Recepcion")) =
Var_TempConstructRecep
        Else
            MsgBox "Please insert values in all text boxes"
        End If
    Case 2
        Var_TempTextBox47 = Val(UserForm1.TextBox47.value)
        Var_TempTextBox48 = Val(UserForm1.TextBox48.value)
        If Var_TempTextBox47 <> 0 And Var_TempTextBox48 <> 0 Then
            Var_TempConstructRecep = "Pois" & "(" & Var_TempTextBox48 & ")"
            s.ResourceCapacity(s.SymbolNumber("ResRecepcion")) = Var_TempTextBox47
            s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Recepcion")) =
Var_TempConstructRecep
        Else
            MsgBox "Please insert values in all text boxes"
        End If
    Case 3
        Var_TempTextBox47 = Val(UserForm1.TextBox47.value)
        Var_TempTextBox48 = Val(UserForm1.TextBox48.value)
        Var_TempTextBox49 = Val(UserForm1.TextBox49.value)
        If Var_TempTextBox47 <> 0 And Var_TempTextBox48 <> 0 And Var_TempTextBox49 <> 0 Then
            Var_TempConstructRecep = "Norm" & "(" & Var_TempTextBox48 & "," &
Var_TempTextBox49 & ")"
            s.ResourceCapacity(s.SymbolNumber("ResRecepcion")) = Var_TempTextBox47
            s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Recepcion")) =
Var_TempConstructRecep
        Else
            MsgBox "Please insert values in all text boxes"
        End If
    Case 4
        Var_TempTextBox47 = Val(UserForm1.TextBox47.value)

```

```

    Var_TempTextBox48 = Val (UserForm1.TextBox48.value)
    Var_TempTextBox49 = Val (UserForm1.TextBox49.value)
    If Var_TempTextBox47 <> 0 And Var_TempTextBox48 <> 0 And Var_TempTextBox49 <> 0 Then
        Var_TempConstructRecep = "Logn" & "(" & Var_TempTextBox48 & "," &
Var_TempTextBox49 & ")"
        s.ResourceCapacity(s.SymbolNumber("ResRecepcion")) = Var_TempTextBox47
        s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Recepcion")) =
Var_TempConstructRecep
    Else
        MsgBox "Please insert values in all text boxes"
    End If
    Case 5
        Var_TempTextBox47 = Val (UserForm1.TextBox47.value)
        Var_TempTextBox48 = Val (UserForm1.TextBox48.value)
        Var_TempTextBox49 = Val (UserForm1.TextBox49.value)
        If Var_TempTextBox47 <> 0 And Var_TempTextBox48 <> 0 And Var_TempTextBox49 <> 0 Then
            If (Var_TempTextBox48 < Var_TempTextBox49) Then
                Var_TempConstructRecep = "Unif" & "(" & Var_TempTextBox48 & "," &
Var_TempTextBox49 & ")"
                s.ResourceCapacity(s.SymbolNumber("ResRecepcion")) = Var_TempTextBox47
                s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Recepcion")) =
Var_TempConstructRecep
            Else
                MsgBox "Minimum value should be inferior than maximum"
            End If
        Else
            MsgBox "Please insert values in all text boxes"
        End If
    Case 6
        Var_TempTextBox47 = Val (UserForm1.TextBox47.value)
        Var_TempTextBox48 = Val (UserForm1.TextBox48.value)
        Var_TempTextBox49 = Val (UserForm1.TextBox49.value)
        Var_TempTextBox50 = Val (UserForm1.TextBox50.value)
        If Var_TempTextBox47 <> 0 And Var_TempTextBox48 <> 0 And Var_TempTextBox49 <> 0 And
Var_TempTextBox50 <> 0 Then
            If (Var_TempTextBox48 < Var_TempTextBox49) And (Var_TempTextBox49 <
Var_TempTextBox50) Then
                Var_TempConstructRecep = "Tria" & "(" & Var_TempTextBox48 & "," &
Var_TempTextBox49 & "," & Var_TempTextBox50 & ")"
                s.ResourceCapacity(s.SymbolNumber("ResRecepcion")) = Var_TempTextBox47
                s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Recepcion")) =
Var_TempConstructRecep
            Else
                MsgBox "Please review Minimum, Mode and Maximum values"
            End If
        Else
            MsgBox "Please insert values in all text boxes"
        End If
    Case Else
        MsgBox "Please confirm distribution selection"
    End Select
End Sub

Private Sub CommandButton12_Click()

Dim m As Model
Dim s As SIMAN

Set m = ThisDocument.Model
Set s = m.SIMAN

Var_Combobox8Index = UserForm1.Combobox8.ListIndex

Select Case Var_Combobox8Index
    Case 1

```

```

Var_TempTextBox86 = Val (UserForm1.TextBox86.value)
If Var_TempTextBox86 <> 0 Then
    Var_TempRouteRecIntPark = "Expo" & "(" & Var_TempTextBox86 & ")"
    s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_RouteIntParking")) =
Var_TempRouteRecIntPark
Else
    MsgBox "Please insert values in all text boxes"
End If
Case 2
Var_TempTextBox86 = Val (UserForm1.TextBox86.value)
If Var_TempTextBox86 <> 0 Then
    Var_TempRouteRecIntPark = "Pois" & "(" & Var_TempTextBox86 & ")"
    s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_RouteIntParking")) =
Var_TempRouteRecIntPark
Else
    MsgBox "Please insert values in all text boxes"
End If
Case 3
Var_TempTextBox86 = Val (UserForm1.TextBox86.value)
Var_TempTextBox87 = Val (UserForm1.TextBox87.value)
If Var_TempTextBox86 <> 0 And Var_TempTextBox87 <> 0 Then
    Var_TempRouteRecIntPark = "Norm" & "(" & Var_TempTextBox86 & "," &
Var_TempTextBox87 & ")"
    s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_RouteIntParking")) =
Var_TempRouteRecIntPark
Else
    MsgBox "Please insert values in all text boxes"
End If
Case 4
Var_TempTextBox86 = Val (UserForm1.TextBox86.value)
Var_TempTextBox87 = Val (UserForm1.TextBox87.value)
If Var_TempTextBox86 <> 0 And Var_TempTextBox87 <> 0 Then
    Var_TempRouteRecIntPark = "Logn" & "(" & Var_TempTextBox86 & "," &
Var_TempTextBox87 & ")"
    s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_RouteIntParking")) =
Var_TempRouteRecIntPark
Else
    MsgBox "Please insert values in all text boxes"
End If
Case 5
Var_TempTextBox86 = Val (UserForm1.TextBox86.value)
Var_TempTextBox87 = Val (UserForm1.TextBox87.value)
If Var_TempTextBox86 <> 0 And Var_TempTextBox87 <> 0 Then
    If (Var_TempTextBox86 < Var_TempTextBox87) Then
        Var_TempRouteRecIntPark = "Unif" & "(" & Var_TempTextBox86 & "," &
Var_TempTextBox87 & ")"
        s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_RouteIntParking"))
= Var_TempRouteRecIntPark
    Else
        MsgBox "Minimum value should be inferior than maximum"
    End If
Else
    MsgBox "Please insert values in all text boxes"
End If
Case 6
Var_TempTextBox86 = Val (UserForm1.TextBox86.value)
Var_TempTextBox87 = Val (UserForm1.TextBox87.value)
Var_TempTextBox88 = Val (UserForm1.TextBox88.value)
If Var_TempTextBox86 <> 0 And Var_TempTextBox87 <> 0 And Var_TempTextBox88 <> 0 Then
    If (Var_TempTextBox86 < Var_TempTextBox87) And (Var_TempTextBox87 <
Var_TempTextBox88) Then
        Var_TempRouteRecIntPark = "Tria" & "(" & Var_TempTextBox86 & "," &
Var_TempTextBox87 & "," & Var_TempTextBox88 & ")"
        s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_RouteIntParking"))
= Var_TempRouteRecIntPark
    Else
        MsgBox "Please review Minimum, Mode and Maximum values"
    End If
End If

```

```

        End If
    Else
        MsgBox "Please insert values in all text boxes"
    End If
Case Else
    MsgBox "Please confirm distribution selection"
End Select

End Sub

Private Sub CommandButton13_Click()

Dim m As Model
Dim s As SIMAN

Set m = ThisDocument.Model
Set s = m.SIMAN

Var_Combobox9Index = UserForm1.Combobox9.ListIndex

Select Case Var_Combobox9Index
    Case 1
        Var_TempTextBox91 = Val(UserForm1.TextBox91.value)
        If Var_TempTextBox91 <> 0 Then
            Var_TempRouteIntParkGR = "Expo" & "(" & Var_TempTextBox91 & ")"
            s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_RouteGeneralRepairs"))
= Var_TempRouteIntParkGR
        Else
            MsgBox "Please insert values in all text boxes"
        End If
    Case 2
        Var_TempTextBox91 = Val(UserForm1.TextBox91.value)
        If Var_TempTextBox91 <> 0 Then
            Var_TempRouteIntParkGR = "Pois" & "(" & Var_TempTextBox91 & ")"
            s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_RouteGeneralRepairs"))
= Var_TempRouteIntParkGR
        Else
            MsgBox "Please insert values in all text boxes"
        End If
    Case 3
        Var_TempTextBox91 = Val(UserForm1.TextBox91.value)
        Var_TempTextBox90 = Val(UserForm1.TextBox90.value)
        If Var_TempTextBox91 <> 0 And Var_TempTextBox90 <> 0 Then
            Var_TempRouteIntParkGR = "Norm" & "(" & Var_TempTextBox91 & "," &
Var_TempTextBox90 & ")"
            s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_RouteGeneralRepairs"))
= Var_TempRouteIntParkGR
        Else
            MsgBox "Please insert values in all text boxes"
        End If
    Case 4
        Var_TempTextBox91 = Val(UserForm1.TextBox91.value)
        Var_TempTextBox90 = Val(UserForm1.TextBox90.value)
        If Var_TempTextBox91 <> 0 And Var_TempTextBox90 <> 0 Then
            Var_TempRouteIntParkGR = "Logn" & "(" & Var_TempTextBox91 & "," &
Var_TempTextBox90 & ")"
            s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_RouteGeneralRepairs"))
= Var_TempRouteIntParkGR
        Else
            MsgBox "Please insert values in all text boxes"
        End If
    Case 5
        Var_TempTextBox91 = Val(UserForm1.TextBox91.value)
        Var_TempTextBox90 = Val(UserForm1.TextBox90.value)
        If Var_TempTextBox91 <> 0 And Var_TempTextBox90 <> 0 Then
            If (Var_TempTextBox91 < Var_TempTextBox90) Then

```

```

        Var_TempRouteIntParkGR = "Unif" & "(" & Var_TempTextBox91 & "," &
Var_TempTextBox90 & ")"

s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_RouteGeneralRepairs")) =
Var_TempRouteIntParkGR
    Else
        MsgBox "Minimum value should be inferior than maximum"
    End If
    Else
        MsgBox "Please insert values in all text boxes"
    End If
    Case 6
        Var_TempTextBox91 = Val(UserForm1.TextBox91.value)
        Var_TempTextBox90 = Val(UserForm1.TextBox90.value)
        Var_TempTextBox89 = Val(UserForm1.TextBox89.value)
        If Var_TempTextBox91 <> 0 And Var_TempTextBox90 <> 0 And Var_TempTextBox89 <> 0 Then
            If (Var_TempTextBox91 < Var_TempTextBox90) And (Var_TempTextBox90 <
Var_TempTextBox89) Then
                Var_TempRouteIntParkGR = "Tria" & "(" & Var_TempTextBox91 & "," &
Var_TempTextBox90 & "," & Var_TempTextBox89 & ")"

s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_RouteGeneralRepairs")) =
Var_TempRouteIntParkGR
            Else
                MsgBox "Please review Minimum, Mode and Maximum values"
            End If
            Else
                MsgBox "Please insert values in all text boxes"
            End If
            Case Else
                MsgBox "Please confirm distribution selection"
            End Select
        End Sub

Private Sub CommandButton14_Click()

Dim m As Model
Dim s As SIMAN

Set m = ThisDocument.Model
Set s = m.SIMAN

Var_Combobox10Index = UserForm1.Combobox10.ListIndex

Select Case Var_Combobox10Index
    Case 1
        Var_TempTextBox95 = Val(UserForm1.TextBox95.value)
        If Var_TempTextBox95 <> 0 Then
            Var_TempRouteIntParkQS = "Expo" & "(" & Var_TempTextBox95 & ")"
            s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_RouteQuickRepairs")) =
Var_TempRouteIntParkQS
        Else
            MsgBox "Please insert values in all text boxes"
        End If
    Case 2
        Var_TempTextBox95 = Val(UserForm1.TextBox95.value)
        If Var_TempTextBox95 <> 0 Then
            Var_TempRouteIntParkQS = "Pois" & "(" & Var_TempTextBox95 & ")"
            s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_RouteQuickRepairs")) =
Var_TempRouteIntParkQS
        Else
            MsgBox "Please insert values in all text boxes"
        End If
    Case 3
        Var_TempTextBox95 = Val(UserForm1.TextBox95.value)
        Var_TempTextBox94 = Val(UserForm1.TextBox94.value)

```

```

        If Var_TempTextBox95 <> 0 And Var_TempTextBox94 <> 0 Then
            Var_TempRouteIntParkQS = "Norm" & "(" & Var_TempTextBox95 & "," &
Var_TempTextBox94 & ")"
            s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_RouteQuickRepairs")) =
Var_TempRouteIntParkQS
        Else
            MsgBox "Please insert values in all text boxes"
        End If
    Case 4
        Var_TempTextBox95 = Val(UserForm1.TextBox95.value)
        Var_TempTextBox94 = Val(UserForm1.TextBox94.value)
        If Var_TempTextBox95 <> 0 And Var_TempTextBox94 <> 0 Then
            Var_TempRouteIntParkQS = "Logn" & "(" & Var_TempTextBox95 & "," &
Var_TempTextBox94 & ")"
            s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_RouteQuickRepairs")) =
Var_TempRouteIntParkQS
        Else
            MsgBox "Please insert values in all text boxes"
        End If
    Case 5
        Var_TempTextBox95 = Val(UserForm1.TextBox95.value)
        Var_TempTextBox94 = Val(UserForm1.TextBox94.value)
        If Var_TempTextBox95 <> 0 And Var_TempTextBox94 <> 0 Then
            If (Var_TempTextBox95 < Var_TempTextBox94) Then
                Var_TempRouteIntParkQS = "Unif" & "(" & Var_TempTextBox95 & "," &
Var_TempTextBox94 & ")"
            s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_RouteQuickRepairs")) =
Var_TempRouteIntParkQS
            Else
                MsgBox "Minimum value should be inferior than maximum"
            End If
        Else
            MsgBox "Please insert values in all text boxes"
        End If
    Case 6
        Var_TempTextBox95 = Val(UserForm1.TextBox95.value)
        Var_TempTextBox94 = Val(UserForm1.TextBox94.value)
        Var_TempTextBox93 = Val(UserForm1.TextBox93.value)
        If Var_TempTextBox95 <> 0 And Var_TempTextBox94 <> 0 And Var_TempTextBox93 <> 0 Then
            If (Var_TempTextBox95 < Var_TempTextBox94) And (Var_TempTextBox94 <
Var_TempTextBox93) Then
                Var_TempRouteIntParkQS = "Tria" & "(" & Var_TempTextBox95 & "," &
Var_TempTextBox94 & "," & Var_TempTextBox93 & ")"
            s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_RouteQuickRepairs")) =
Var_TempRouteIntParkQS
            Else
                MsgBox "Please review Minimum, Mode and Maximum values"
            End If
        Else
            MsgBox "Please insert values in all text boxes"
        End If
    Case Else
        MsgBox "Please confirm distribution selection"
    End Select
End Sub

Private Sub CommandButton15_Click()

Dim m As Model
Dim s As SIMAN

Set m = ThisDocument.Model
Set s = m.SIMAN

```

```

Var_Combobox11Index = UserForm1.Combobox11.ListIndex

Select Case Var_Combobox11Index
    Case 1
        Var_TempTextBox98 = Val(UserForm1.TextBox98.value)
        If Var_TempTextBox98 <> 0 Then
            Var_TempRouteIntParkOC = "Expo" & "(" & Var_TempTextBox98 & ")"
            s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_RouteOilChange")) =
Var_TempRouteIntParkOC
        Else
            MsgBox "Please insert values in all text boxes"
        End If
    Case 2
        Var_TempTextBox98 = Val(UserForm1.TextBox98.value)
        If Var_TempTextBox98 <> 0 Then
            Var_TempRouteIntParkOC = "Pois" & "(" & Var_TempTextBox98 & ")"
            s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_RouteOilChange")) =
Var_TempRouteIntParkOC
        Else
            MsgBox "Please insert values in all text boxes"
        End If
    Case 3
        Var_TempTextBox98 = Val(UserForm1.TextBox98.value)
        Var_TempTextBox97 = Val(UserForm1.TextBox97.value)
        If Var_TempTextBox98 <> 0 And Var_TempTextBox97 <> 0 Then
            Var_TempRouteIntParkOC = "Norm" & "(" & Var_TempTextBox98 & "," &
Var_TempTextBox97 & ")"
            s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_RouteOilChange")) =
Var_TempRouteIntParkOC
        Else
            MsgBox "Please insert values in all text boxes"
        End If
    Case 4
        Var_TempTextBox98 = Val(UserForm1.TextBox98.value)
        Var_TempTextBox97 = Val(UserForm1.TextBox97.value)
        If Var_TempTextBox98 <> 0 And Var_TempTextBox97 <> 0 Then
            Var_TempRouteIntParkOC = "Logn" & "(" & Var_TempTextBox98 & "," &
Var_TempTextBox97 & ")"
            s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_RouteOilChange")) =
Var_TempRouteIntParkOC
        Else
            MsgBox "Please insert values in all text boxes"
        End If
    Case 5
        Var_TempTextBox98 = Val(UserForm1.TextBox98.value)
        Var_TempTextBox97 = Val(UserForm1.TextBox97.value)
        If Var_TempTextBox98 <> 0 And Var_TempTextBox97 <> 0 Then
            If (Var_TempTextBox98 < Var_TempTextBox97) Then
                Var_TempRouteIntParkOC = "Unif" & "(" & Var_TempTextBox98 & "," &
Var_TempTextBox97 & ")"
                s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_RouteOilChange"))
= Var_TempRouteIntParkOC
            Else
                MsgBox "Minimum value should be inferior than maximum"
            End If
        Else
            MsgBox "Please insert values in all text boxes"
        End If
    Case 6
        Var_TempTextBox98 = Val(UserForm1.TextBox98.value)
        Var_TempTextBox97 = Val(UserForm1.TextBox97.value)
        Var_TempTextBox84 = Val(UserForm1.TextBox84.value)
        If Var_TempTextBox98 <> 0 And Var_TempTextBox97 <> 0 And Var_TempTextBox84 <> 0 Then
            If (Var_TempTextBox98 < Var_TempTextBox97) And (Var_TempTextBox97 <
Var_TempTextBox84) Then
                Var_TempRouteIntParkOC = "Tria" & "(" & Var_TempTextBox98 & "," &
Var_TempTextBox97 & "," & Var_TempTextBox84 & ")"

```

```

        s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_RouteOilChange"))
= Var_TempRouteIntParkOC
        Else
            MsgBox "Please review Minimum, Mode and Maximum values"
        End If
    Else
        MsgBox "Please insert values in all text boxes"
    End If
Case Else
    MsgBox "Please confirm distribution selection"
End Select

End Sub

Private Sub CommandButton16_Click()

Dim m As Model
Dim s As SIMAN

Set m = ThisDocument.Model
Set s = m.SIMAN

Var_TimeRouteIntPark =
s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_RouteIntParking"))
Var_DistributionRouteIntPark = Left(Var_TimeRouteIntPark, 4)

Select Case Var_DistributionRouteIntPark
    Case "Expo", "EXPO"
        UserForm1.Label108.Visible = False
        UserForm1.Label109.Visible = False
        UserForm1.TextBox70.Visible = False
        UserForm1.TextBox71.Visible = False
        UserForm1.TextBox68.value = "Exponencial"
        UserForm1.TextBox69.value = Val(Mid(Var_TimeRouteIntPark, 6))
        UserForm1.Label107.Caption = "Mean"
        UserForm1.Label107.Visible = True
        UserForm1.TextBox68.Visible = True
        UserForm1.TextBox69.Visible = True
    Case "Pois", "POIS"
        UserForm1.Label108.Visible = False
        UserForm1.Label109.Visible = False
        UserForm1.TextBox70.Visible = False
        UserForm1.TextBox71.Visible = False
        UserForm1.TextBox68.value = "Poisson"
        UserForm1.TextBox69.value = Val(Mid(Var_TimeRouteIntPark, 6))
        UserForm1.Label107.Caption = "Mean"
        UserForm1.Label107.Visible = True
        UserForm1.TextBox68.Visible = True
        UserForm1.TextBox69.Visible = True
    Case "Norm", "NORM"
        UserForm1.Label109.Visible = False
        UserForm1.TextBox71.Visible = False
        UserForm1.TextBox68.value = "Normal"
        UserForm1.TextBox69.value = Val(Mid(Var_TimeRouteIntPark, 6))
        UserForm1.TextBox70.value = Val(Mid(Var_TimeRouteIntPark,
InStr(Var_TimeRouteIntPark, ",") + 1))
        UserForm1.Label107.Caption = "Mean"
        UserForm1.Label107.Visible = True
        UserForm1.Label108.Caption = "StdDev"
        UserForm1.Label108.Visible = True
        UserForm1.TextBox68.Visible = True
        UserForm1.TextBox69.Visible = True
        UserForm1.TextBox70.Visible = True
    Case "Logn", "LOGN"
        UserForm1.Label109.Visible = False
        UserForm1.TextBox71.Visible = False
        UserForm1.TextBox68.value = "LogNormal"

```

```

        UserForm1.TextBox69.value = Val(Mid(Var_TimeRouteIntPark, 6))
        UserForm1.TextBox70.value = Val(Mid(Var_TimeRouteIntPark,
InStr(Var_TimeRouteIntPark, ",") + 1))
        UserForm1.Label1107.Caption = "LogMean"
        UserForm1.Label1107.Visible = True
        UserForm1.Label1108.Caption = "LogStd"
        UserForm1.Label1108.Visible = True
        UserForm1.TextBox68.Visible = True
        UserForm1.TextBox69.Visible = True
        UserForm1.TextBox70.Visible = True
    Case "Unif", "UNIF"
        UserForm1.Label1109.Visible = False
        UserForm1.TextBox71.Visible = False
        UserForm1.TextBox68.value = "Uniform"
        UserForm1.TextBox69.value = Val(Mid(Var_TimeRouteIntPark, 6))
        UserForm1.TextBox70.value = Val(Mid(Var_TimeRouteIntPark,
InStr(Var_TimeRouteIntPark, ",") + 1))
        UserForm1.Label1107.Caption = "Minimum"
        UserForm1.Label1107.Visible = True
        UserForm1.Label1108.Caption = "Maximum"
        UserForm1.Label1108.Visible = True
        UserForm1.TextBox68.Visible = True
        UserForm1.TextBox69.Visible = True
        UserForm1.TextBox70.Visible = True
    Case "Tria", "TRIA"
        UserForm1.TextBox68.value = "Triangular"
        UserForm1.TextBox69.value = Val(Mid(Var_TimeRouteIntPark, 6))
        UserForm1.TextBox70.value = Val(Mid(Var_TimeRouteIntPark,
InStr(Var_TimeRouteIntPark, ",") + 1))
        UserForm1.TextBox71.value = Val(Mid(Var_TimeRouteIntPark,
InStr(InStr(Var_TimeRouteIntPark, ",") + 1, Var_TimeRouteIntPark, ",") + 1))
        UserForm1.Label1107.Caption = "Minimum"
        UserForm1.Label1107.Visible = True
        UserForm1.Label1108.Caption = "Mode"
        UserForm1.Label1108.Visible = True
        UserForm1.Label1109.Caption = "Maximum"
        UserForm1.Label1109.Visible = True
        UserForm1.TextBox68.Visible = True
        UserForm1.TextBox69.Visible = True
        UserForm1.TextBox70.Visible = True
        UserForm1.TextBox71.Visible = True
End Select

Var_TimeRouteGR =
s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_RouteGeneralRepairs"))
Var_DistributionRouteGR = Left(Var_TimeRouteGR, 4)

Select Case Var_DistributionRouteGR
    Case "Expo", "EXPO"
        UserForm1.Label1113.Visible = False
        UserForm1.Label1112.Visible = False
        UserForm1.TextBox73.Visible = False
        UserForm1.TextBox72.Visible = False
        UserForm1.TextBox75.value = "Exponencial"
        UserForm1.TextBox74.value = Val(Mid(Var_TimeRouteGR, 6))
        UserForm1.Label111.Caption = "Mean"
        UserForm1.Label111.Visible = True
        UserForm1.TextBox75.Visible = True
        UserForm1.TextBox74.Visible = True
    Case "Pois", "POIS"
        UserForm1.Label1113.Visible = False
        UserForm1.Label1112.Visible = False
        UserForm1.TextBox73.Visible = False
        UserForm1.TextBox72.Visible = False
        UserForm1.TextBox75.value = "Poisson"
        UserForm1.TextBox74.value = Val(Mid(Var_TimeRouteGR, 6))
        UserForm1.Label111.Caption = "Mean"

```

```

        UserForm1.Label111.Visible = True
        UserForm1.TextBox75.Visible = True
        UserForm1.TextBox74.Visible = True
    Case "Norm", "NORM"
        UserForm1.Label112.Visible = False
        UserForm1.TextBox72.Visible = False
        UserForm1.TextBox75.value = "Normal"
        UserForm1.TextBox74.value = Val(Mid(Var_TimeRouteGR, 6))
        UserForm1.TextBox73.value = Val(Mid(Var_TimeRouteGR, InStr(Var_TimeRouteGR, ",") +
1))
        UserForm1.Label111.Caption = "Mean"
        UserForm1.Label111.Visible = True
        UserForm1.Label113.Caption = "StdDev"
        UserForm1.Label113.Visible = True
        UserForm1.TextBox75.Visible = True
        UserForm1.TextBox74.Visible = True
        UserForm1.TextBox73.Visible = True
    Case "Logn", "LOGN"
        UserForm1.Label112.Visible = False
        UserForm1.TextBox72.Visible = False
        UserForm1.TextBox75.value = "LogNormal"
        UserForm1.TextBox74.value = Val(Mid(Var_TimeRouteGR, 6))
        UserForm1.TextBox73.value = Val(Mid(Var_TimeRouteGR, InStr(Var_TimeRouteGR, ",") +
1))
        UserForm1.Label111.Caption = "LogMean"
        UserForm1.Label111.Visible = True
        UserForm1.Label113.Caption = "LogStd"
        UserForm1.Label113.Visible = True
        UserForm1.TextBox75.Visible = True
        UserForm1.TextBox74.Visible = True
        UserForm1.TextBox73.Visible = True
    Case "Unif", "UNIF"
        UserForm1.Label112.Visible = False
        UserForm1.TextBox72.Visible = False
        UserForm1.TextBox75.value = "Uniform"
        UserForm1.TextBox74.value = Val(Mid(Var_TimeRouteGR, 6))
        UserForm1.TextBox73.value = Val(Mid(Var_TimeRouteGR, InStr(Var_TimeRouteGR, ",") +
1))
        UserForm1.Label111.Caption = "Minimum"
        UserForm1.Label111.Visible = True
        UserForm1.Label113.Caption = "Maximum"
        UserForm1.Label113.Visible = True
        UserForm1.TextBox75.Visible = True
        UserForm1.TextBox74.Visible = True
        UserForm1.TextBox73.Visible = True
    Case "Tria", "TRIA"
        UserForm1.TextBox75.value = "Triangular"
        UserForm1.TextBox74.value = Val(Mid(Var_TimeRouteGR, 6))
        UserForm1.TextBox73.value = Val(Mid(Var_TimeRouteGR, InStr(Var_TimeRouteGR, ",") +
1))
        UserForm1.TextBox72.value = Val(Mid(Var_TimeRouteGR, InStr(InStr(Var_TimeRouteGR,
",") + 1, Var_TimeRouteGR, ",") + 1))
        UserForm1.Label111.Caption = "Minimum"
        UserForm1.Label111.Visible = True
        UserForm1.Label113.Caption = "Mode"
        UserForm1.Label113.Visible = True
        UserForm1.Label112.Caption = "Maximum"
        UserForm1.Label112.Visible = True
        UserForm1.TextBox75.Visible = True
        UserForm1.TextBox74.Visible = True
        UserForm1.TextBox73.Visible = True
        UserForm1.TextBox72.Visible = True
End Select

Var_TimeRouteQS =
s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_RouteQuickRepairs"))
Var_DistributionRouteQS = Left(Var_TimeRouteQS, 4)

```

```

Select Case Var_DistributionRouteQS
  Case "Expo", "EXPO"
    UserForm1.Label119.Visible = False
    UserForm1.Label118.Visible = False
    UserForm1.TextBox77.Visible = False
    UserForm1.TextBox76.Visible = False
    UserForm1.TextBox79.value = "Exponencial"
    UserForm1.TextBox78.value = Val (Mid(Var_TimeRouteQS, 6))
    UserForm1.Label117.Caption = "Mean"
    UserForm1.Label117.Visible = True
    UserForm1.TextBox79.Visible = True
    UserForm1.TextBox78.Visible = True
  Case "Pois", "POIS"
    UserForm1.Label119.Visible = False
    UserForm1.Label118.Visible = False
    UserForm1.TextBox77.Visible = False
    UserForm1.TextBox76.Visible = False
    UserForm1.TextBox79.value = "Poisson"
    UserForm1.TextBox78.value = Val (Mid(Var_TimeRouteQS, 6))
    UserForm1.Label117.Caption = "Mean"
    UserForm1.Label117.Visible = True
    UserForm1.TextBox79.Visible = True
    UserForm1.TextBox78.Visible = True
  Case "Norm", "NORM"
    UserForm1.Label118.Visible = False
    UserForm1.TextBox76.Visible = False
    UserForm1.TextBox79.value = "Normal"
    UserForm1.TextBox78.value = Val (Mid(Var_TimeRouteQS, 6))
    UserForm1.TextBox77.value = Val (Mid(Var_TimeRouteQS, InStr (Var_TimeRouteQS, ",") +
1))
    UserForm1.Label117.Caption = "Mean"
    UserForm1.Label117.Visible = True
    UserForm1.Label119.Caption = "StdDev"
    UserForm1.Label119.Visible = True
    UserForm1.TextBox79.Visible = True
    UserForm1.TextBox78.Visible = True
    UserForm1.TextBox77.Visible = True
  Case "Logn", "LOGN"
    UserForm1.Label118.Visible = False
    UserForm1.TextBox76.Visible = False
    UserForm1.TextBox79.value = "LogNormal"
    UserForm1.TextBox78.value = Val (Mid(Var_TimeRouteQS, 6))
    UserForm1.TextBox77.value = Val (Mid(Var_TimeRouteQS, InStr (Var_TimeRouteQS, ",") +
1))
    UserForm1.Label117.Caption = "LogMean"
    UserForm1.Label117.Visible = True
    UserForm1.Label119.Caption = "LogStd"
    UserForm1.Label119.Visible = True
    UserForm1.TextBox79.Visible = True
    UserForm1.TextBox78.Visible = True
    UserForm1.TextBox77.Visible = True
  Case "Unif", "UNIF"
    UserForm1.Label118.Visible = False
    UserForm1.TextBox76.Visible = False
    UserForm1.TextBox79.value = "Uniform"
    UserForm1.TextBox78.value = Val (Mid(Var_TimeRouteQS, 6))
    UserForm1.TextBox77.value = Val (Mid(Var_TimeRouteQS, InStr (Var_TimeRouteQS, ",") +
1))
    UserForm1.Label117.Caption = "Minimum"
    UserForm1.Label117.Visible = True
    UserForm1.Label119.Caption = "Maximum"
    UserForm1.Label119.Visible = True
    UserForm1.TextBox79.Visible = True
    UserForm1.TextBox78.Visible = True
    UserForm1.TextBox77.Visible = True
  Case "Tria", "TRIA"

```

```

        UserForm1.TextBox79.value = "Triangular"
        UserForm1.TextBox78.value = Val(Mid(Var_TimeRouteQS, 6))
        UserForm1.TextBox77.value = Val(Mid(Var_TimeRouteQS, InStr(Var_TimeRouteQS, ",") +
1))
        UserForm1.TextBox76.value = Val(Mid(Var_TimeRouteQS, InStr(InStr(Var_TimeRouteQS,
",") + 1, Var_TimeRouteQS, ",") + 1))
        UserForm1.Label117.Caption = "Minimum"
        UserForm1.Label117.Visible = True
        UserForm1.Label119.Caption = "Mode"
        UserForm1.Label119.Visible = True
        UserForm1.Label118.Caption = "Maximum"
        UserForm1.Label118.Visible = True
        UserForm1.TextBox79.Visible = True
        UserForm1.TextBox78.Visible = True
        UserForm1.TextBox77.Visible = True
        UserForm1.TextBox76.Visible = True
End Select

Var_TimeRouteOC = s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_RouteOilChange"))
Var_DistributionTimeOC = Left(Var_TimeRouteOC, 4)

Select Case Var_DistributionTimeOC
    Case "Expo", "EXPO"
        UserForm1.Label125.Visible = False
        UserForm1.Label124.Visible = False
        UserForm1.TextBox81.Visible = False
        UserForm1.TextBox80.Visible = False
        UserForm1.TextBox83.value = "Exponencial"
        UserForm1.TextBox82.value = Val(Mid(Var_TimeRouteOC, 6))
        UserForm1.Label123.Caption = "Mean"
        UserForm1.Label123.Visible = True
        UserForm1.TextBox83.Visible = True
        UserForm1.TextBox82.Visible = True
    Case "Pois", "POIS"
        UserForm1.Label125.Visible = False
        UserForm1.Label124.Visible = False
        UserForm1.TextBox81.Visible = False
        UserForm1.TextBox80.Visible = False
        UserForm1.TextBox83.value = "Poisson"
        UserForm1.TextBox82.value = Val(Mid(Var_TimeRouteOC, 6))
        UserForm1.Label123.Caption = "Mean"
        UserForm1.Label123.Visible = True
        UserForm1.TextBox83.Visible = True
        UserForm1.TextBox82.Visible = True
    Case "Norm", "NORM"
        UserForm1.Label124.Visible = False
        UserForm1.TextBox80.Visible = False
        UserForm1.TextBox83.value = "Normal"
        UserForm1.TextBox82.value = Val(Mid(Var_TimeRouteOC, 6))
        UserForm1.TextBox81.value = Val(Mid(Var_TimeRouteOC, InStr(Var_TimeRouteOC, ",") +
1))
        UserForm1.Label123.Caption = "Mean"
        UserForm1.Label123.Visible = True
        UserForm1.Label125.Caption = "StdDev"
        UserForm1.Label125.Visible = True
        UserForm1.TextBox83.Visible = True
        UserForm1.TextBox82.Visible = True
        UserForm1.TextBox81.Visible = True
    Case "Logn", "LOGN"
        UserForm1.Label124.Visible = False
        UserForm1.TextBox80.Visible = False
        UserForm1.TextBox83.value = "LogNormal"
        UserForm1.TextBox82.value = Val(Mid(Var_TimeRouteOC, 6))
        UserForm1.TextBox81.value = Val(Mid(Var_TimeRouteOC, InStr(Var_TimeRouteOC, ",") +
1))
        UserForm1.Label123.Caption = "LogMean"
        UserForm1.Label123.Visible = True

```

```

    UserForm1.Label125.Caption = "LogStd"
    UserForm1.Label125.Visible = True
    UserForm1.TextBox83.Visible = True
    UserForm1.TextBox82.Visible = True
    UserForm1.TextBox81.Visible = True
    Case "Unif", "UNIF"
        UserForm1.Label124.Visible = False
        UserForm1.TextBox80.Visible = False
        UserForm1.TextBox83.value = "Uniform"
        UserForm1.TextBox82.value = Val(Mid(Var_TimeRouteOC, 6))
        UserForm1.TextBox81.value = Val(Mid(Var_TimeRouteOC, InStr(Var_TimeRouteOC, ",") +
1))
        UserForm1.Label123.Caption = "Minimum"
        UserForm1.Label123.Visible = True
        UserForm1.Label125.Caption = "Maximum"
        UserForm1.Label125.Visible = True
        UserForm1.TextBox83.Visible = True
        UserForm1.TextBox82.Visible = True
        UserForm1.TextBox81.Visible = True
    Case "Tria", "TRIA"
        UserForm1.TextBox83.value = "Triangular"
        UserForm1.TextBox82.value = Val(Mid(Var_TimeRouteOC, 6))
        UserForm1.TextBox81.value = Val(Mid(Var_TimeRouteOC, InStr(Var_TimeRouteOC, ",") +
1))
        UserForm1.TextBox80.value = Val(Mid(Var_TimeRouteOC, InStr(InStr(Var_TimeRouteOC,
",") + 1, Var_TimeRouteOC, ",") + 1))
        UserForm1.Label123.Caption = "Minimum"
        UserForm1.Label123.Visible = True
        UserForm1.Label125.Caption = "Mode"
        UserForm1.Label125.Visible = True
        UserForm1.Label124.Caption = "Maximum"
        UserForm1.Label124.Visible = True
        UserForm1.TextBox83.Visible = True
        UserForm1.TextBox82.Visible = True
        UserForm1.TextBox81.Visible = True
        UserForm1.TextBox80.Visible = True
End Select

UserForm1.ComboBox8.Visible = True
UserForm1.ComboBox9.Visible = True
UserForm1.ComboBox10.Visible = True
UserForm1.ComboBox11.Visible = True

End Sub

Private Sub CommandButton17_Click()

Dim m As Model
Dim s As SIMAN

Set m = ThisDocument.Model
Set s = m.SIMAN

UserForm1.TextBox99.value = s.InitializeSystemBetweenReplications
UserForm1.TextBox100.value = s.InitializeStatisticsBetweenReplications
UserForm1.TextBox102.value = m.NumberOfReplications
UserForm1.TextBox101.value = m.HoursPerDay
UserForm1.TextBox103.value = m.ReplicationLength
UserForm1.TextBox105.value = m.WarmUpPeriod

UserForm1.TextBox99.Visible = True
UserForm1.TextBox100.Visible = True
UserForm1.TextBox101.Visible = True
UserForm1.TextBox102.Visible = True
UserForm1.TextBox103.Visible = True
UserForm1.TextBox105.Visible = True

```

```
UserForm1.CommandButton18.Visible = True

Var_RepLenghtTimeUnits = m.ReplicationLengthTimeUnits
Select Case Var_RepLenghtTimeUnits
    Case 0
        UserForm1.TextBox104.value = "Days"
        UserForm1.TextBox104.Visible = True
    Case 1
        UserForm1.TextBox104.value = "Hours"
        UserForm1.TextBox104.Visible = True
    Case 2
        UserForm1.TextBox104.value = "Minutes"
        UserForm1.TextBox104.Visible = True
    Case 3
        UserForm1.TextBox104.value = "Seconds"
        UserForm1.TextBox104.Visible = True
End Select

Var_WarmUpTimeUnits = m.WarmUpPeriodTimeUnits
Select Case Var_WarmUpTimeUnits
    Case 0
        UserForm1.TextBox106.value = "Days"
        UserForm1.TextBox106.Visible = True
    Case 1
        UserForm1.TextBox106.value = "Hours"
        UserForm1.TextBox106.Visible = True
    Case 2
        UserForm1.TextBox106.value = "Minutes"
        UserForm1.TextBox106.Visible = True
    Case 3
        UserForm1.TextBox106.value = "Seconds"
        UserForm1.TextBox106.Visible = True
End Select

If (UserForm1.TextBox99.value = True) Then
    UserForm1.ComboBox12.ListIndex = 0
Else
    UserForm1.ComboBox12.ListIndex = 1
End If
UserForm1.ComboBox12.Visible = True

If (UserForm1.TextBox100.value = True) Then
    UserForm1.ComboBox13.ListIndex = 0
Else
    UserForm1.ComboBox13.ListIndex = 1
End If
UserForm1.ComboBox13.Visible = True

UserForm1.TextBox107.value = UserForm1.TextBox102.value
UserForm1.TextBox107.Visible = True
UserForm1.TextBox108.value = UserForm1.TextBox101.value
UserForm1.TextBox108.Visible = True
UserForm1.TextBox109.value = UserForm1.TextBox103.value
UserForm1.TextBox109.Visible = True

Select Case UserForm1.TextBox104.value
    Case "Days"
        UserForm1.ComboBox14.ListIndex = 0
    Case "Hours"
        UserForm1.ComboBox14.ListIndex = 1
    Case "Minutes"
        UserForm1.ComboBox14.ListIndex = 2
    Case "Seconds"
        UserForm1.ComboBox14.ListIndex = 3
End Select
UserForm1.ComboBox14.Visible = True
```

```

UserForm1.TextBox110.value = UserForm1.TextBox105.value
UserForm1.TextBox110.Visible = True

Select Case UserForm1.TextBox106.value
    Case "Days"
        UserForm1.ComboBox15.ListIndex = 0
    Case "Hours"
        UserForm1.ComboBox15.ListIndex = 1
    Case "Minutes"
        UserForm1.ComboBox15.ListIndex = 2
    Case "Seconds"
        UserForm1.ComboBox15.ListIndex = 3
End Select
UserForm1.ComboBox15.Visible = True

End Sub

Private Sub CommandButton18_Click()

    Dim m As Model
    Dim s As SIMAN

    Set m = ThisDocument.Model
    Set s = m.SIMAN

    Var_ComboBox12Index = UserForm1.ComboBox12.ListIndex
    Select Case Var_ComboBox12Index
        Case 0
            s.InitializeSystemBetweenReplications = True
        Case 1
            s.InitializeSystemBetweenReplications = False
    End Select

    Var_ComboBox13Index = UserForm1.ComboBox13.ListIndex
    Select Case Var_ComboBox13Index
        Case 0
            s.InitializeStatisticsBetweenReplications = True
        Case 1
            s.InitializeStatisticsBetweenReplications = False
    End Select

    Dim Var_TempTextBox107 As String

    If UserForm1.TextBox107.value <> "" Then
        Var_TempTextBox107 = CStr(UserForm1.TextBox107.value)
        m.NumberOfReplications = Var_TempTextBox107
    Else
        MsgBox "Please insert values in all text boxes"
    End If

End Sub

Private Sub CommandButton19_Click()
    Dim m As Model
    Dim s As SIMAN

    Set m = ThisDocument.Model
    Set s = m.SIMAN

    s.VariableArrayValue(s.SymbolNumber("Alter_Quantity_Entity1")) = "2"
    s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity1")) = "Norm(1,0.1)"
    s.VariableArrayValue(s.SymbolNumber("Alter_Quantity_Entity2")) = "2"
    s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity2")) = "Norm(1,0.1)"
    s.VariableArrayValue(s.SymbolNumber("Alter_Quantity_Entity3")) = "2"
    s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity3")) = "Norm(1,0.1)"

    s.ResourceCapacity(s.SymbolNumber("ResReception")) = "2"

```

```

s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Recepcion")) = "Norm(10,5) "

s.ResourceCapacity(s.SymbolNumber("RscGeneralRepairs")) = "15"
s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_GeneralRepairs")) = "Norm(7.5,2) "
s.ResourceCapacity(s.SymbolNumber("RscQuickServices")) = "2"
s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_QuickServices")) =
"Norm(0.75,0.25) "
s.ResourceCapacity(s.SymbolNumber("RscOilChange")) = "2"
s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_OilChange")) = "Norm(0.75,0.25) "

End Sub

Private Sub CommandButton2_Click()

Dim m As Model
Dim s As SIMAN

Set m = ThisDocument.Model
Set s = m.SIMAN

UserForm1.Hide

End Sub

Private Sub CommandButton20_Click()
Dim m As Model
Dim s As SIMAN

Set m = ThisDocument.Model
Set s = m.SIMAN

s.VariableArrayValue(s.SymbolNumber("Alter_Quantity_Entity1")) = "2"
s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity1")) = "Norm(1,0.1) "
s.VariableArrayValue(s.SymbolNumber("Alter_Quantity_Entity2")) = "2"
s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity2")) = "Norm(1,0.1) "
s.VariableArrayValue(s.SymbolNumber("Alter_Quantity_Entity3")) = "2"
s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity3")) = "Norm(1,0.1) "

s.ResourceCapacity(s.SymbolNumber("ResRecepcion")) = "1"
s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Recepcion")) = "Norm(10,5) "

s.ResourceCapacity(s.SymbolNumber("RscGeneralRepairs")) = "8"
s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_GeneralRepairs")) = "Norm(7.5,2) "
s.ResourceCapacity(s.SymbolNumber("RscQuickServices")) = "1"
s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_QuickServices")) =
"Norm(0.75,0.25) "
s.ResourceCapacity(s.SymbolNumber("RscOilChange")) = "1"
s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_OilChange")) = "Norm(0.75,0.25) "

End Sub

Private Sub CommandButton21_Click()

Dim m As Model
Dim s As SIMAN

Set m = ThisDocument.Model
Set s = m.SIMAN

s.VariableArrayValue(s.SymbolNumber("Alter_Quantity_Entity1")) = "2"
s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity1")) = "Norm(1,0.1) "
s.VariableArrayValue(s.SymbolNumber("Alter_Quantity_Entity2")) = "2"
s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity2")) = "Norm(1,0.1) "
s.VariableArrayValue(s.SymbolNumber("Alter_Quantity_Entity3")) = "2"
s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity3")) = "Norm(1,0.1) "

s.ResourceCapacity(s.SymbolNumber("ResRecepcion")) = "3"

```

```

s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Recepcion")) = "Norm(10,5)"

s.ResourceCapacity(s.SymbolNumber("RscGeneralRepairs")) = "20"
s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_GeneralRepairs")) = "Norm(7.5,2)"
s.ResourceCapacity(s.SymbolNumber("RscQuickServices")) = "3"
s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_QuickServices")) =
"Norm(0.75,0.25)"
s.ResourceCapacity(s.SymbolNumber("RscOilChange")) = "3"
s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_OilChange")) = "Norm(0.75,0.25)"

End Sub

Private Sub CommandButton3_Click()

Dim m As Model
Dim s As SIMAN

Set m = ThisDocument.Model
Set s = m.SIMAN

UserForm1.TextBox1.value = s.VariableArrayValue(s.SymbolNumber("Alter_Quantity_Entity1"))
UserForm1.TextBox1.Visible = True

Var_ExpressionGR = s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity1"))
Var_DistributionGR = Left(Var_ExpressionGR, 4)

Select Case Var_DistributionGR
Case "Expo", "EXPO"
    UserForm1.Label8.Visible = False
    UserForm1.Label9.Visible = False
    UserForm1.TextBox4.Visible = False
    UserForm1.TextBox5.Visible = False
    UserForm1.TextBox2.value = "Exponencial"
    UserForm1.TextBox3.value = Val(Mid(Var_ExpressionGR, 6))
    UserForm1.Label7.Caption = "Mean"
    UserForm1.Label7.Visible = True
    UserForm1.TextBox2.Visible = True
    UserForm1.TextBox3.Visible = True
Case "Pois", "POIS"
    UserForm1.Label8.Visible = False
    UserForm1.Label9.Visible = False
    UserForm1.TextBox4.Visible = False
    UserForm1.TextBox5.Visible = False
    UserForm1.TextBox2.value = "Poisson"
    UserForm1.TextBox3.value = Val(Mid(Var_ExpressionGR, 6))
    UserForm1.Label7.Caption = "Mean"
    UserForm1.Label7.Visible = True
    UserForm1.TextBox2.Visible = True
    UserForm1.TextBox3.Visible = True
Case "Norm", "NORM"
    UserForm1.Label9.Visible = False
    UserForm1.TextBox5.Visible = False
    UserForm1.TextBox2.value = "Normal"
    UserForm1.TextBox3.value = Val(Mid(Var_ExpressionGR, 6))
    UserForm1.TextBox4.value = Val(Mid(Var_ExpressionGR, InStr(Var_ExpressionGR, ",") +
1))
    UserForm1.Label7.Caption = "Mean"
    UserForm1.Label7.Visible = True
    UserForm1.Label8.Caption = "StdDev"
    UserForm1.Label8.Visible = True
    UserForm1.TextBox2.Visible = True
    UserForm1.TextBox3.Visible = True
    UserForm1.TextBox4.Visible = True
Case "Logn", "LOGN"
    UserForm1.Label9.Visible = False
    UserForm1.TextBox5.Visible = False
    UserForm1.TextBox2.value = "LogNormal"

```

```

        UserForm1.TextBox3.value = Val(Mid(Var_ExpressionGR, 6))
        UserForm1.TextBox4.value = Val(Mid(Var_ExpressionGR, InStr(Var_ExpressionGR, ",") +
1))
        UserForm1.Label7.Caption = "LogMean"
        UserForm1.Label7.Visible = True
        UserForm1.Label8.Caption = "LogStd"
        UserForm1.Label8.Visible = True
        UserForm1.TextBox2.Visible = True
        UserForm1.TextBox3.Visible = True
        UserForm1.TextBox4.Visible = True
    Case "Unif", "UNIF"
        UserForm1.Label9.Visible = False
        UserForm1.TextBox5.Visible = False
        UserForm1.TextBox2.value = "Uniform"
        UserForm1.TextBox3.value = Val(Mid(Var_ExpressionGR, 6))
        UserForm1.TextBox4.value = Val(Mid(Var_ExpressionGR, InStr(Var_ExpressionGR, ",") +
1))
        UserForm1.Label7.Caption = "Minimum"
        UserForm1.Label7.Visible = True
        UserForm1.Label8.Caption = "Maximum"
        UserForm1.Label8.Visible = True
        UserForm1.TextBox2.Visible = True
        UserForm1.TextBox3.Visible = True
        UserForm1.TextBox4.Visible = True
    Case "Tria", "TRIA"
        UserForm1.TextBox2.value = "Triangular"
        UserForm1.TextBox3.value = Val(Mid(Var_ExpressionGR, 6))
        UserForm1.TextBox4.value = Val(Mid(Var_ExpressionGR, InStr(Var_ExpressionGR, ",") +
1))
        UserForm1.TextBox5.value = Val(Mid(Var_ExpressionGR, InStr(InStr(Var_ExpressionGR,
",") + 1, Var_ExpressionGR, ",") + 1))
        UserForm1.Label7.Caption = "Minimum"
        UserForm1.Label7.Visible = True
        UserForm1.Label8.Caption = "Mode"
        UserForm1.Label8.Visible = True
        UserForm1.Label9.Caption = "Maximum"
        UserForm1.Label9.Visible = True
        UserForm1.TextBox2.Visible = True
        UserForm1.TextBox3.Visible = True
        UserForm1.TextBox4.Visible = True
        UserForm1.TextBox5.Visible = True
End Select

UserForm1.TextBox10.value = s.VariableArrayValue(s.SymbolNumber("Alter_Quantity_Entity2"))
UserForm1.TextBox10.Visible = True

Var_ExpressionQS = s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity2"))
Var_DistributionQS = Left(Var_ExpressionQS, 4)

Select Case Var_DistributionQS
    Case "Expo", "EXPO"
        UserForm1.Label11.Visible = False
        UserForm1.Label10.Visible = False
        UserForm1.TextBox7.Visible = False
        UserForm1.TextBox6.Visible = False
        UserForm1.TextBox9.value = "Exponencial"
        UserForm1.TextBox8.value = Val(Mid(Var_ExpressionQS, 6))
        UserForm1.Label12.Caption = "Mean"
        UserForm1.Label12.Visible = True
        UserForm1.TextBox9.Visible = True
        UserForm1.TextBox8.Visible = True
    Case "Pois", "POIS"
        UserForm1.Label11.Visible = False
        UserForm1.Label10.Visible = False
        UserForm1.TextBox7.Visible = False
        UserForm1.TextBox6.Visible = False
        UserForm1.TextBox9.value = "Poisson"

```

```

    UserForm1.TextBox8.value = Val(Mid(Var_ExpressionQS, 6))
    UserForm1.Label12.Caption = "Mean"
    UserForm1.Label12.Visible = True
    UserForm1.TextBox9.Visible = True
    UserForm1.TextBox8.Visible = True
Case "Norm", "NORM"
    UserForm1.Label10.Visible = False
    UserForm1.TextBox6.Visible = False
    UserForm1.TextBox9.value = "Normal"
    UserForm1.TextBox8.value = Val(Mid(Var_ExpressionQS, 6))
    UserForm1.TextBox7.value = Val(Mid(Var_ExpressionQS, InStr(Var_ExpressionQS, ",") +
1))
    UserForm1.Label12.Caption = "Mean"
    UserForm1.Label12.Visible = True
    UserForm1.Label11.Caption = "StdDev"
    UserForm1.Label11.Visible = True
    UserForm1.TextBox9.Visible = True
    UserForm1.TextBox8.Visible = True
    UserForm1.TextBox7.Visible = True
Case "Logn", "LOGN"
    UserForm1.Label10.Visible = False
    UserForm1.TextBox6.Visible = False
    UserForm1.TextBox9.value = "LogNormal"
    UserForm1.TextBox8.value = Val(Mid(Var_ExpressionQS, 6))
    UserForm1.TextBox7.value = Val(Mid(Var_ExpressionQS, InStr(Var_ExpressionQS, ",") +
1))
    UserForm1.Label12.Caption = "LogMean"
    UserForm1.Label12.Visible = True
    UserForm1.Label11.Caption = "LogStd"
    UserForm1.Label11.Visible = True
    UserForm1.TextBox9.Visible = True
    UserForm1.TextBox8.Visible = True
    UserForm1.TextBox7.Visible = True
Case "Unif", "UNIF"
    UserForm1.Label10.Visible = False
    UserForm1.TextBox6.Visible = False
    UserForm1.TextBox9.value = "Uniform"
    UserForm1.TextBox8.value = Val(Mid(Var_ExpressionQS, 6))
    UserForm1.TextBox7.value = Val(Mid(Var_ExpressionQS, InStr(Var_ExpressionQS, ",") +
1))
    UserForm1.Label12.Caption = "Minimum"
    UserForm1.Label12.Visible = True
    UserForm1.Label11.Caption = "Maximum"
    UserForm1.Label11.Visible = True
    UserForm1.TextBox9.Visible = True
    UserForm1.TextBox8.Visible = True
    UserForm1.TextBox7.Visible = True
Case "Tria", "TRIA"
    UserForm1.TextBox9.value = "Triangular"
    UserForm1.TextBox8.value = Val(Mid(Var_ExpressionQS, 6))
    UserForm1.TextBox7.value = Val(Mid(Var_ExpressionQS, InStr(Var_ExpressionQS, ",") +
1))
    UserForm1.TextBox6.value = Val(Mid(Var_ExpressionQS, InStr(InStr(Var_ExpressionQS,
",") + 1, Var_ExpressionQS, ",") + 1))
    UserForm1.Label12.Caption = "Minimum"
    UserForm1.Label12.Visible = True
    UserForm1.Label11.Caption = "Mode"
    UserForm1.Label11.Visible = True
    UserForm1.Label10.Caption = "Maximum"
    UserForm1.Label10.Visible = True
    UserForm1.TextBox9.Visible = True
    UserForm1.TextBox8.Visible = True
    UserForm1.TextBox7.Visible = True
    UserForm1.TextBox6.Visible = True
End Select

UserForm1.TextBox15.value = s.VariableArrayValue(s.SymbolNumber("Alter_Quantity_Entity3"))

```

```

UserForm1.TextBox15.Visible = True

Var_ExpressionOC = s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity3"))
Var_DistributionOC = Left(Var_ExpressionOC, 4)

Select Case Var_DistributionOC
  Case "Expo", "EXPO"
    UserForm1.Label16.Visible = False
    UserForm1.Label15.Visible = False
    UserForm1.TextBox12.Visible = False
    UserForm1.TextBox11.Visible = False
    UserForm1.TextBox14.value = "Exponencial"
    UserForm1.TextBox13.value = Val(Mid(Var_ExpressionOC, 6))
    UserForm1.Label17.Caption = "Mean"
    UserForm1.Label17.Visible = True
    UserForm1.TextBox14.Visible = True
    UserForm1.TextBox13.Visible = True
  Case "Pois", "POIS"
    UserForm1.Label16.Visible = False
    UserForm1.Label15.Visible = False
    UserForm1.TextBox12.Visible = False
    UserForm1.TextBox11.Visible = False
    UserForm1.TextBox14.value = "Poisson"
    UserForm1.TextBox13.value = Val(Mid(Var_ExpressionOC, 6))
    UserForm1.Label17.Caption = "Mean"
    UserForm1.Label17.Visible = True
    UserForm1.TextBox14.Visible = True
    UserForm1.TextBox13.Visible = True
  Case "Norm", "NORM"
    UserForm1.Label15.Visible = False
    UserForm1.TextBox11.Visible = False
    UserForm1.TextBox14.value = "Normal"
    UserForm1.TextBox13.value = Val(Mid(Var_ExpressionOC, 6))
    UserForm1.TextBox12.value = Val(Mid(Var_ExpressionOC, InStr(Var_ExpressionOC, ",") +
1))
    UserForm1.Label17.Caption = "Mean"
    UserForm1.Label17.Visible = True
    UserForm1.Label16.Caption = "StdDev"
    UserForm1.Label16.Visible = True
    UserForm1.TextBox14.Visible = True
    UserForm1.TextBox13.Visible = True
    UserForm1.TextBox12.Visible = True
  Case "Logn", "LOGN"
    UserForm1.Label15.Visible = False
    UserForm1.TextBox11.Visible = False
    UserForm1.TextBox14.value = "LogNormal"
    UserForm1.TextBox13.value = Val(Mid(Var_ExpressionOC, 6))
    UserForm1.TextBox12.value = Val(Mid(Var_ExpressionOC, InStr(Var_ExpressionOC, ",") +
1))
    UserForm1.Label17.Caption = "LogMean"
    UserForm1.Label17.Visible = True
    UserForm1.Label16.Caption = "LogStd"
    UserForm1.Label16.Visible = True
    UserForm1.TextBox14.Visible = True
    UserForm1.TextBox13.Visible = True
    UserForm1.TextBox12.Visible = True
  Case "Unif", "UNIF"
    UserForm1.Label15.Visible = False
    UserForm1.TextBox11.Visible = False
    UserForm1.TextBox14.value = "Uniform"
    UserForm1.TextBox13.value = Val(Mid(Var_ExpressionOC, 6))
    UserForm1.TextBox12.value = Val(Mid(Var_ExpressionOC, InStr(Var_ExpressionOC, ",") +
1))
    UserForm1.Label17.Caption = "Minimum"
    UserForm1.Label17.Visible = True
    UserForm1.Label16.Caption = "Maximum"
    UserForm1.Label16.Visible = True

```

```

        UserForm1.TextBox14.Visible = True
        UserForm1.TextBox13.Visible = True
        UserForm1.TextBox12.Visible = True
    Case "Tria", "TRIA"
        UserForm1.TextBox14.value = "Triangular"
        UserForm1.TextBox13.value = Val(Mid(Var_ExpressionOC, 6))
        UserForm1.TextBox12.value = Val(Mid(Var_ExpressionOC, InStr(Var_ExpressionOC, ",") +
1))
        UserForm1.TextBox11.value = Val(Mid(Var_ExpressionOC, InStr(InStr(Var_ExpressionOC,
",") + 1, Var_ExpressionOC, ",") + 1))
        UserForm1.Label17.Caption = "Minimum"
        UserForm1.Label17.Visible = True
        UserForm1.Label16.Caption = "Mode"
        UserForm1.Label16.Visible = True
        UserForm1.Label15.Caption = "Maximum"
        UserForm1.Label15.Visible = True
        UserForm1.TextBox14.Visible = True
        UserForm1.TextBox13.Visible = True
        UserForm1.TextBox12.Visible = True
        UserForm1.TextBox11.Visible = True
    End Select

UserForm1.TextBox17.Visible = True
UserForm1.TextBox26.Visible = True
UserForm1.TextBox30.Visible = True
UserForm1.ComboBox1.Visible = True
UserForm1.ComboBox2.Visible = True
UserForm1.ComboBox3.Visible = True

End Sub

Private Sub CommandButton4_Click()

Dim m As Model
Dim s As SIMAN

Set m = ThisDocument.Model
Set s = m.SIMAN

Var_ComboBox1Index = UserForm1.ComboBox1.ListIndex

Select Case Var_ComboBox1Index
    Case 1
        Var_TempTextBox17 = Val(UserForm1.TextBox17.value)
        Var_TempTextBox19 = Val(UserForm1.TextBox19.value)
        If Var_TempTextBox17 <> 0 And Var_TempTextBox19 <> 0 Then
            Var_TempConstructGR = "Expo" & "(" & Var_TempTextBox19 & ")"
            s.VariableArrayValue(s.SymbolNumber("Alter_Quantity_Entity1")) =
Var_TempTextBox17
            s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity1")) =
Var_TempConstructGR
        Else
            MsgBox "Please insert values in all text boxes"
        End If
    Case 2
        Var_TempTextBox17 = Val(UserForm1.TextBox17.value)
        Var_TempTextBox19 = Val(UserForm1.TextBox19.value)
        If Var_TempTextBox17 <> 0 And Var_TempTextBox19 <> 0 Then
            Var_TempConstructGR = "Pois" & "(" & Var_TempTextBox19 & ")"
            s.VariableArrayValue(s.SymbolNumber("Alter_Quantity_Entity1")) =
Var_TempTextBox17
            s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity1")) =
Var_TempConstructGR
        Else
            MsgBox "Please insert values in all text boxes"
        End If
    Case 3

```

```

    Var_TempTextBox17 = Val(UserForm1.TextBox17.value)
    Var_TempTextBox19 = Val(UserForm1.TextBox19.value)
    Var_TempTextBox20 = Val(UserForm1.TextBox20.value)
    If Var_TempTextBox17 <> 0 And Var_TempTextBox19 <> 0 And Var_TempTextBox20 <> 0 Then
        Var_TempConstructGR = "Norm" & "(" & Var_TempTextBox19 & "," & Var_TempTextBox20
& ")"
        s.VariableArrayValue(s.SymbolNumber("Alter_Quantity_Entity1")) =
Var_TempTextBox17
        s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity1")) =
Var_TempConstructGR
    Else
        MsgBox "Please insert values in all text boxes"
    End If
Case 4
    Var_TempTextBox17 = Val(UserForm1.TextBox17.value)
    Var_TempTextBox19 = Val(UserForm1.TextBox19.value)
    Var_TempTextBox20 = Val(UserForm1.TextBox20.value)
    If Var_TempTextBox17 <> 0 And Var_TempTextBox19 <> 0 And Var_TempTextBox20 <> 0 Then
        Var_TempConstructGR = "Logn" & "(" & Var_TempTextBox19 & "," & Var_TempTextBox20
& ")"
        s.VariableArrayValue(s.SymbolNumber("Alter_Quantity_Entity1")) =
Var_TempTextBox17
        s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity1")) =
Var_TempConstructGR
    Else
        MsgBox "Please insert values in all text boxes"
    End If
Case 5
    Var_TempTextBox17 = Val(UserForm1.TextBox17.value)
    Var_TempTextBox19 = Val(UserForm1.TextBox19.value)
    Var_TempTextBox20 = Val(UserForm1.TextBox20.value)
    If Var_TempTextBox17 <> 0 And Var_TempTextBox19 <> 0 And Var_TempTextBox20 <> 0 Then
        If (Var_TempTextBox19 < Var_TempTextBox20) Then
            Var_TempConstructGR = "Unif" & "(" & Var_TempTextBox19 & "," &
Var_TempTextBox20 & ")"
            s.VariableArrayValue(s.SymbolNumber("Alter_Quantity_Entity1")) =
Var_TempTextBox17
            s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity1")) =
Var_TempConstructGR
        Else
            MsgBox "Minimum value should be inferior than maximum"
        End If
    Else
        MsgBox "Please insert values in all text boxes"
    End If
Case 6
    Var_TempTextBox17 = Val(UserForm1.TextBox17.value)
    Var_TempTextBox19 = Val(UserForm1.TextBox19.value)
    Var_TempTextBox20 = Val(UserForm1.TextBox20.value)
    Var_TempTextBox21 = Val(UserForm1.TextBox21.value)
    If Var_TempTextBox17 <> 0 And Var_TempTextBox19 <> 0 And Var_TempTextBox20 <> 0 And
Var_TempTextBox21 <> 0 Then
        If (Var_TempTextBox19 < Var_TempTextBox20) And (Var_TempTextBox20 <
Var_TempTextBox21) Then
            Var_TempConstructGR = "Tria" & "(" & Var_TempTextBox19 & "," &
Var_TempTextBox20 & "," & Var_TempTextBox21 & ")"
            s.VariableArrayValue(s.SymbolNumber("Alter_Quantity_Entity1")) =
Var_TempTextBox17
            s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity1")) =
Var_TempConstructGR
        Else
            MsgBox "Please review Minimum, Mode and Maximum values"
        End If
    Else
        MsgBox "Please insert values in all text boxes"
    End If
Case Else

```

```

        MsgBox "Please confirm distribution selection"
    End Select

End Sub

Private Sub CommandButton5_Click()
    Dim m As Model
    Dim s As SIMAN

    Set m = ThisDocument.Model
    Set s = m.SIMAN

    Var_Combobox2Index = UserForm1.Combobox2.ListIndex

    Select Case Var_Combobox2Index
        Case 1
            Var_TempTextBox26 = Val(UserForm1.TextBox26.value)
            Var_TempTextBox24 = Val(UserForm1.TextBox24.value)
            If Var_TempTextBox26 <> 0 And Var_TempTextBox24 <> 0 Then
                Var_TempConstructQR = "Expo" & "(" & Var_TempTextBox24 & ")"
                s.VariableArrayValue(s.SymbolNumber("Alter_Quantity_Entity2")) =
Var_TempTextBox26
                s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity2")) =
Var_TempConstructQR
            Else
                MsgBox "Please insert values in all text boxes"
            End If
        Case 2
            Var_TempTextBox26 = Val(UserForm1.TextBox26.value)
            Var_TempTextBox24 = Val(UserForm1.TextBox24.value)
            If Var_TempTextBox26 <> 0 And Var_TempTextBox24 <> 0 Then
                Var_TempConstructQR = "Pois" & "(" & Var_TempTextBox24 & ")"
                s.VariableArrayValue(s.SymbolNumber("Alter_Quantity_Entity2")) =
Var_TempTextBox26
                s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity2")) =
Var_TempConstructQR
            Else
                MsgBox "Please insert values in all text boxes"
            End If
        Case 3
            Var_TempTextBox26 = Val(UserForm1.TextBox26.value)
            Var_TempTextBox24 = Val(UserForm1.TextBox24.value)
            Var_TempTextBox23 = Val(UserForm1.TextBox23.value)
            If Var_TempTextBox26 <> 0 And Var_TempTextBox24 <> 0 And Var_TempTextBox23 <> 0 Then
                Var_TempConstructQR = "Norm" & "(" & Var_TempTextBox24 & "," & Var_TempTextBox23
& ")"
                s.VariableArrayValue(s.SymbolNumber("Alter_Quantity_Entity2")) =
Var_TempTextBox26
                s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity2")) =
Var_TempConstructQR
            Else
                MsgBox "Please insert values in all text boxes"
            End If
        Case 4
            Var_TempTextBox26 = Val(UserForm1.TextBox26.value)
            Var_TempTextBox24 = Val(UserForm1.TextBox24.value)
            Var_TempTextBox23 = Val(UserForm1.TextBox23.value)
            If Var_TempTextBox26 <> 0 And Var_TempTextBox24 <> 0 And Var_TempTextBox23 <> 0 Then
                Var_TempConstructQR = "Logn" & "(" & Var_TempTextBox24 & "," & Var_TempTextBox23
& ")"
                s.VariableArrayValue(s.SymbolNumber("Alter_Quantity_Entity2")) =
Var_TempTextBox26
                s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity2")) =
Var_TempConstructQR
            Else
                MsgBox "Please insert values in all text boxes"
            End If
    End Select

```

```

Case 5
    Var_TempTextBox26 = Val (UserForm1.TextBox26.value)
    Var_TempTextBox24 = Val (UserForm1.TextBox24.value)
    Var_TempTextBox23 = Val (UserForm1.TextBox23.value)
    If Var_TempTextBox26 <> 0 And Var_TempTextBox24 <> 0 And Var_TempTextBox23 <> 0 Then
        If (Var_TempTextBox24 < Var_TempTextBox23) Then
            Var_TempConstructQR = "Unif" & "(" & Var_TempTextBox24 & "," &
Var_TempTextBox23 & ")"
                s.VariableArrayValue(s.SymbolNumber("Alter_Quantity_Entity2")) =
Var_TempTextBox26
                s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity2")) =
Var_TempConstructQR
            Else
                MsgBox "Minimum value should be inferior than maximum"
            End If
        Else
            MsgBox "Please insert values in all text boxes"
        End If
    Case 6
        Var_TempTextBox26 = Val (UserForm1.TextBox26.value)
        Var_TempTextBox24 = Val (UserForm1.TextBox24.value)
        Var_TempTextBox23 = Val (UserForm1.TextBox23.value)
        Var_TempTextBox22 = Val (UserForm1.TextBox22.value)
        If Var_TempTextBox26 <> 0 And Var_TempTextBox24 <> 0 And Var_TempTextBox23 <> 0 And
Var_TempTextBox22 <> 0 Then
            If (Var_TempTextBox24 < Var_TempTextBox23) And (Var_TempTextBox23 <
Var_TempTextBox22) Then
                Var_TempConstructQR = "Tria" & "(" & Var_TempTextBox24 & "," &
Var_TempTextBox23 & "," & Var_TempTextBox22 & ")"
                s.VariableArrayValue(s.SymbolNumber("Alter_Quantity_Entity2")) =
Var_TempTextBox26
                s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity2")) =
Var_TempConstructQR
            Else
                MsgBox "Please review Minimum, Mode and Maximum values"
            End If
        Else
            MsgBox "Please insert values in all text boxes"
        End If
    Case Else
        MsgBox "Please confirm distribution selection"
    End Select

End Sub

Private Sub CommandButton6_Click()

Dim m As Model
Dim s As SIMAN

Set m = ThisDocument.Model
Set s = m.SIMAN

Var_Combobox3Index = UserForm1.Combobox3.ListIndex

Select Case Var_Combobox3Index
    Case 1
        Var_TempTextBox30 = Val (UserForm1.TextBox30.value)
        Var_TempTextBox28 = Val (UserForm1.TextBox28.value)
        If Var_TempTextBox30 <> 0 And Var_TempTextBox28 <> 0 Then
            Var_TempConstructOC = "Expo" & "(" & Var_TempTextBox28 & ")"
                s.VariableArrayValue(s.SymbolNumber("Alter_Quantity_Entity3")) =
Var_TempTextBox30
                s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity3")) =
Var_TempConstructOC
            Else

```

```

        MsgBox "Please insert values in all text boxes"
    End If
Case 2
    Var_TempTextBox30 = Val(UserForm1.TextBox30.value)
    Var_TempTextBox28 = Val(UserForm1.TextBox28.value)
    If Var_TempTextBox30 <> 0 And Var_TempTextBox28 <> 0 Then
        Var_TempConstructOC = "Pois" & "(" & Var_TempTextBox28 & ")"
        s.VariableArrayValue(s.SymbolNumber("Alter_Quantity_Entity3")) =
Var_TempTextBox30
        s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity3")) =
Var_TempConstructOC
    Else
        MsgBox "Please insert values in all text boxes"
    End If
Case 3
    Var_TempTextBox30 = Val(UserForm1.TextBox30.value)
    Var_TempTextBox28 = Val(UserForm1.TextBox28.value)
    Var_TempTextBox27 = Val(UserForm1.TextBox27.value)
    If Var_TempTextBox30 <> 0 And Var_TempTextBox28 <> 0 And Var_TempTextBox27 <> 0 Then
        Var_TempConstructOC = "Norm" & "(" & Var_TempTextBox28 & "," & Var_TempTextBox27
& ")"
        s.VariableArrayValue(s.SymbolNumber("Alter_Quantity_Entity3")) =
Var_TempTextBox30
        s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity3")) =
Var_TempConstructOC
    Else
        MsgBox "Please insert values in all text boxes"
    End If
Case 4
    Var_TempTextBox30 = Val(UserForm1.TextBox30.value)
    Var_TempTextBox28 = Val(UserForm1.TextBox28.value)
    Var_TempTextBox27 = Val(UserForm1.TextBox27.value)
    If Var_TempTextBox30 <> 0 And Var_TempTextBox28 <> 0 And Var_TempTextBox27 <> 0 Then
        Var_TempConstructOC = "Logn" & "(" & Var_TempTextBox28 & "," & Var_TempTextBox27
& ")"
        s.VariableArrayValue(s.SymbolNumber("Alter_Quantity_Entity3")) =
Var_TempTextBox30
        s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity3")) =
Var_TempConstructOC
    Else
        MsgBox "Please insert values in all text boxes"
    End If
Case 5
    Var_TempTextBox30 = Val(UserForm1.TextBox30.value)
    Var_TempTextBox28 = Val(UserForm1.TextBox28.value)
    Var_TempTextBox27 = Val(UserForm1.TextBox27.value)
    If Var_TempTextBox30 <> 0 And Var_TempTextBox28 <> 0 And Var_TempTextBox27 <> 0 Then
        If (Var_TempTextBox28 < Var_TempTextBox27) Then
            Var_TempConstructOC = "Unif" & "(" & Var_TempTextBox28 & "," &
Var_TempTextBox27 & ")"
            s.VariableArrayValue(s.SymbolNumber("Alter_Quantity_Entity3")) =
Var_TempTextBox30
            s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity3")) =
Var_TempConstructOC
        Else
            MsgBox "Minimum value should be inferior than maximum"
        End If
    Else
        MsgBox "Please insert values in all text boxes"
    End If
Case 6
    Var_TempTextBox30 = Val(UserForm1.TextBox30.value)
    Var_TempTextBox28 = Val(UserForm1.TextBox28.value)
    Var_TempTextBox27 = Val(UserForm1.TextBox27.value)
    Var_TempTextBox16 = Val(UserForm1.TextBox16.value)
    If Var_TempTextBox30 <> 0 And Var_TempTextBox28 <> 0 And Var_TempTextBox27 <> 0 And
Var_TempTextBox16 <> 0 Then

```

```

        If (Var_TempTextBox28 < Var_TempTextBox27) And (Var_TempTextBox27 <
Var_TempTextBox16) Then
            Var_TempConstructOC = "Tria" & "(" & Var_TempTextBox28 & "," &
Var_TempTextBox27 & "," & Var_TempTextBox16 & ")"
            s.VariableArrayValue(s.SymbolNumber("Alter_Quantity_Entity3")) =
Var_TempTextBox30
            s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity3")) =
Var_TempConstructOC
        Else
            MsgBox "Please review Minimum, Mode and Maximum values"
        End If
    Else
        MsgBox "Please insert values in all text boxes"
    End If
Case Else
    MsgBox "Please confirm distribution selection"
End Select

End Sub

Private Sub CommandButton7_Click()

Dim m As Model
Dim s As SIMAN

Set m = ThisDocument.Model
Set s = m.SIMAN

UserForm1.TextBox31.value = s.ResourceCapacity(s.SymbolNumber("ResRecepcion"))
UserForm1.TextBox31.Visible = True

Var_ExpressionRecepcion =
s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Recepcion"))
Var_DistributionRecepcion = Left(Var_ExpressionRecepcion, 4)

Select Case Var_DistributionRecepcion
    Case "Expo", "EXPO"
        UserForm1.Label52.Visible = False
        UserForm1.Label53.Visible = False
        UserForm1.TextBox34.Visible = False
        UserForm1.TextBox35.Visible = False
        UserForm1.TextBox32.value = "Exponencial"
        UserForm1.TextBox33.value = Val(Mid(Var_ExpressionRecepcion, 6))
        UserForm1.Label51.Caption = "Mean"
        UserForm1.Label51.Visible = True
        UserForm1.TextBox32.Visible = True
        UserForm1.TextBox33.Visible = True
    Case "Pois", "POIS"
        UserForm1.Label52.Visible = False
        UserForm1.Label53.Visible = False
        UserForm1.TextBox34.Visible = False
        UserForm1.TextBox35.Visible = False
        UserForm1.TextBox32.value = "Poisson"
        UserForm1.TextBox33.value = Val(Mid(Var_ExpressionRecepcion, 6))
        UserForm1.Label51.Caption = "Mean"
        UserForm1.Label51.Visible = True
        UserForm1.TextBox32.Visible = True
        UserForm1.TextBox33.Visible = True
    Case "Norm", "NORM"
        UserForm1.Label53.Visible = False
        UserForm1.TextBox35.Visible = False
        UserForm1.TextBox32.value = "Normal"
        UserForm1.TextBox33.value = Val(Mid(Var_ExpressionRecepcion, 6))
        UserForm1.TextBox34.value = Val(Mid(Var_ExpressionRecepcion,
Instr(Var_ExpressionRecepcion, ",") + 1))
        UserForm1.Label51.Caption = "Mean"

```

```

    UserForm1.Label51.Visible = True
    UserForm1.Label52.Caption = "StdDev"
    UserForm1.Label52.Visible = True
    UserForm1.TextBox32.Visible = True
    UserForm1.TextBox33.Visible = True
    UserForm1.TextBox34.Visible = True
    Case "Logn", "LOGN"
        UserForm1.Label53.Visible = False
        UserForm1.TextBox35.Visible = False
        UserForm1.TextBox32.value = "LogNormal"
        UserForm1.TextBox33.value = Val(Mid(Var_ExpressionRecepcion, 6))
        UserForm1.TextBox34.value = Val(Mid(Var_ExpressionRecepcion,
InStr(Var_ExpressionRecepcion, ",") + 1))
        UserForm1.Label51.Caption = "LogMean"
        UserForm1.Label51.Visible = True
        UserForm1.Label52.Caption = "LogStd"
        UserForm1.Label52.Visible = True
        UserForm1.TextBox32.Visible = True
        UserForm1.TextBox33.Visible = True
        UserForm1.TextBox34.Visible = True
    Case "Unif", "UNIF"
        UserForm1.Label53.Visible = False
        UserForm1.TextBox35.Visible = False
        UserForm1.TextBox32.value = "Uniform"
        UserForm1.TextBox33.value = Val(Mid(Var_ExpressionRecepcion, 6))
        UserForm1.TextBox34.value = Val(Mid(Var_ExpressionRecepcion,
InStr(Var_ExpressionRecepcion, ",") + 1))
        UserForm1.Label51.Caption = "Minimum"
        UserForm1.Label51.Visible = True
        UserForm1.Label52.Caption = "Maximum"
        UserForm1.Label52.Visible = True
        UserForm1.TextBox32.Visible = True
        UserForm1.TextBox33.Visible = True
        UserForm1.TextBox34.Visible = True
    Case "Tria", "TRIA"
        UserForm1.TextBox32.value = "Triangular"
        UserForm1.TextBox33.value = Val(Mid(Var_ExpressionRecepcion, 6))
        UserForm1.TextBox34.value = Val(Mid(Var_ExpressionRecepcion,
InStr(Var_ExpressionRecepcion, ",") + 1))
        UserForm1.TextBox35.value = Val(Mid(Var_ExpressionRecepcion,
InStr(InStr(Var_ExpressionRecepcion, ",") + 1, Var_ExpressionRecepcion, ",") + 1))
        UserForm1.Label51.Caption = "Minimum"
        UserForm1.Label51.Visible = True
        UserForm1.Label52.Caption = "Mode"
        UserForm1.Label52.Visible = True
        UserForm1.Label53.Caption = "Maximum"
        UserForm1.Label53.Visible = True
        UserForm1.TextBox32.Visible = True
        UserForm1.TextBox33.Visible = True
        UserForm1.TextBox34.Visible = True
        UserForm1.TextBox35.Visible = True
End Select

UserForm1.TextBox40.value = s.ResourceCapacity(s.SymbolNumber("RscGeneralRepairs"))
UserForm1.TextBox40.Visible = True

Var_ExpressionTimeGR =
s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_GeneralRepairs"))
Var_DistributionTimeGR = Left(Var_ExpressionTimeGR, 4)

Select Case Var_DistributionTimeGR
    Case "Expo", "EXPO"
        UserForm1.Label55.Visible = False
        UserForm1.Label54.Visible = False
        UserForm1.TextBox37.Visible = False
        UserForm1.TextBox36.Visible = False
        UserForm1.TextBox39.value = "Exponencial"

```

```

    UserForm1.TextBox38.value = Val(Mid(Var_ExpressionTimeGR, 6))
    UserForm1.Label56.Caption = "Mean"
    UserForm1.Label56.Visible = True
    UserForm1.TextBox39.Visible = True
    UserForm1.TextBox38.Visible = True
Case "Pois", "POIS"
    UserForm1.Label55.Visible = False
    UserForm1.Label54.Visible = False
    UserForm1.TextBox37.Visible = False
    UserForm1.TextBox36.Visible = False
    UserForm1.TextBox39.value = "Poisson"
    UserForm1.TextBox38.value = Val(Mid(Var_ExpressionTimeGR, 6))
    UserForm1.Label56.Caption = "Mean"
    UserForm1.Label56.Visible = True
    UserForm1.TextBox39.Visible = True
    UserForm1.TextBox38.Visible = True
Case "Norm", "NORM"
    UserForm1.Label54.Visible = False
    UserForm1.TextBox36.Visible = False
    UserForm1.TextBox39.value = "Normal"
    UserForm1.TextBox38.value = Val(Mid(Var_ExpressionTimeGR, 6))
    UserForm1.TextBox37.value = Val(Mid(Var_ExpressionTimeGR,
Instr(Var_ExpressionTimeGR, ",") + 1))
    UserForm1.Label56.Caption = "Mean"
    UserForm1.Label56.Visible = True
    UserForm1.Label55.Caption = "StdDev"
    UserForm1.Label55.Visible = True
    UserForm1.TextBox39.Visible = True
    UserForm1.TextBox38.Visible = True
    UserForm1.TextBox37.Visible = True
Case "Logn", "LOGN"
    UserForm1.Label54.Visible = False
    UserForm1.TextBox36.Visible = False
    UserForm1.TextBox39.value = "LogNormal"
    UserForm1.TextBox38.value = Val(Mid(Var_ExpressionTimeGR, 6))
    UserForm1.TextBox37.value = Val(Mid(Var_ExpressionTimeGR,
Instr(Var_ExpressionTimeGR, ",") + 1))
    UserForm1.Label56.Caption = "LogMean"
    UserForm1.Label56.Visible = True
    UserForm1.Label55.Caption = "LogStd"
    UserForm1.Label55.Visible = True
    UserForm1.TextBox39.Visible = True
    UserForm1.TextBox38.Visible = True
    UserForm1.TextBox37.Visible = True
Case "Unif", "UNIF"
    UserForm1.Label54.Visible = False
    UserForm1.TextBox36.Visible = False
    UserForm1.TextBox39.value = "Uniform"
    UserForm1.TextBox38.value = Val(Mid(Var_ExpressionTimeGR, 6))
    UserForm1.TextBox37.value = Val(Mid(Var_ExpressionTimeGR,
Instr(Var_ExpressionTimeGR, ",") + 1))
    UserForm1.Label56.Caption = "Minimum"
    UserForm1.Label56.Visible = True
    UserForm1.Label55.Caption = "Maximum"
    UserForm1.Label55.Visible = True
    UserForm1.TextBox39.Visible = True
    UserForm1.TextBox38.Visible = True
    UserForm1.TextBox37.Visible = True
Case "Tria", "TRIA"
    UserForm1.TextBox39.value = "Triangular"
    UserForm1.TextBox38.value = Val(Mid(Var_ExpressionTimeGR, 6))
    UserForm1.TextBox37.value = Val(Mid(Var_ExpressionTimeGR,
Instr(Var_ExpressionTimeGR, ",") + 1))
    UserForm1.TextBox36.value = Val(Mid(Var_ExpressionTimeGR,
Instr(Instr(Var_ExpressionTimeGR, ",") + 1, Var_ExpressionTimeGR, ",") + 1))
    UserForm1.Label56.Caption = "Minimum"
    UserForm1.Label56.Visible = True

```

```

    UserForm1.Label155.Caption = "Mode"
    UserForm1.Label155.Visible = True
    UserForm1.Label154.Caption = "Maximum"
    UserForm1.Label154.Visible = True
    UserForm1.TextBox39.Visible = True
    UserForm1.TextBox38.Visible = True
    UserForm1.TextBox37.Visible = True
    UserForm1.TextBox36.Visible = True
End Select

UserForm1.TextBox45.value = s.ResourceCapacity(s.SymbolNumber("RscQuickServices"))
UserForm1.TextBox45.Visible = True

Var_ExpressionTimeQS =
s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_QuickServices"))
Var_DistributionTimeQS = Left(Var_ExpressionTimeQS, 4)

Select Case Var_DistributionTimeQS
    Case "Expo", "EXPO"
        UserForm1.Label160.Visible = False
        UserForm1.Label159.Visible = False
        UserForm1.TextBox42.Visible = False
        UserForm1.TextBox41.Visible = False
        UserForm1.TextBox44.value = "Exponencial"
        UserForm1.TextBox43.value = Val(Mid(Var_ExpressionTimeQS, 6))
        UserForm1.Label161.Caption = "Mean"
        UserForm1.Label161.Visible = True
        UserForm1.TextBox44.Visible = True
        UserForm1.TextBox43.Visible = True
    Case "Pois", "POIS"
        UserForm1.Label160.Visible = False
        UserForm1.Label159.Visible = False
        UserForm1.TextBox42.Visible = False
        UserForm1.TextBox41.Visible = False
        UserForm1.TextBox44.value = "Poisson"
        UserForm1.TextBox43.value = Val(Mid(Var_ExpressionTimeQS, 6))
        UserForm1.Label161.Caption = "Mean"
        UserForm1.Label161.Visible = True
        UserForm1.TextBox44.Visible = True
        UserForm1.TextBox43.Visible = True
    Case "Norm", "NORM"
        UserForm1.Label159.Visible = False
        UserForm1.TextBox41.Visible = False
        UserForm1.TextBox44.value = "Normal"
        UserForm1.TextBox43.value = Val(Mid(Var_ExpressionTimeQS, 6))
        UserForm1.TextBox42.value = Val(Mid(Var_ExpressionTimeQS,
InStr(Var_ExpressionTimeQS, ",") + 1))
        UserForm1.Label161.Caption = "Mean"
        UserForm1.Label161.Visible = True
        UserForm1.Label160.Caption = "StdDev"
        UserForm1.Label160.Visible = True
        UserForm1.TextBox44.Visible = True
        UserForm1.TextBox43.Visible = True
        UserForm1.TextBox42.Visible = True
    Case "Logn", "LOGN"
        UserForm1.Label159.Visible = False
        UserForm1.TextBox41.Visible = False
        UserForm1.TextBox44.value = "LogNormal"
        UserForm1.TextBox43.value = Val(Mid(Var_ExpressionTimeQS, 6))
        UserForm1.TextBox42.value = Val(Mid(Var_ExpressionTimeQS,
InStr(Var_ExpressionTimeQS, ",") + 1))
        UserForm1.Label161.Caption = "LogMean"
        UserForm1.Label161.Visible = True
        UserForm1.Label160.Caption = "LogStd"
        UserForm1.Label160.Visible = True
        UserForm1.TextBox44.Visible = True
        UserForm1.TextBox43.Visible = True

```

```

    UserForm1.TextBox42.Visible = True
Case "Unif", "UNIF"
    UserForm1.Label59.Visible = False
    UserForm1.TextBox41.Visible = False
    UserForm1.TextBox44.value = "Uniform"
    UserForm1.TextBox43.value = Val(Mid(Var_ExpressionTimeQS, 6))
    UserForm1.TextBox42.value = Val(Mid(Var_ExpressionTimeQS,
InStr(Var_ExpressionTimeQS, ",") + 1))
    UserForm1.Label61.Caption = "Minimum"
    UserForm1.Label61.Visible = True
    UserForm1.Label60.Caption = "Maximum"
    UserForm1.Label60.Visible = True
    UserForm1.TextBox44.Visible = True
    UserForm1.TextBox43.Visible = True
    UserForm1.TextBox42.Visible = True
Case "Tria", "TRIA"
    UserForm1.TextBox44.value = "Triangular"
    UserForm1.TextBox43.value = Val(Mid(Var_ExpressionTimeQS, 6))
    UserForm1.TextBox42.value = Val(Mid(Var_ExpressionTimeQS,
InStr(Var_ExpressionTimeQS, ",") + 1))
    UserForm1.TextBox41.value = Val(Mid(Var_ExpressionTimeQS,
InStr(InStr(Var_ExpressionTimeQS, ",") + 1, Var_ExpressionTimeQS, ",") + 1))
    UserForm1.Label61.Caption = "Minimum"
    UserForm1.Label61.Visible = True
    UserForm1.Label60.Caption = "Mode"
    UserForm1.Label60.Visible = True
    UserForm1.Label59.Caption = "Maximum"
    UserForm1.Label59.Visible = True
    UserForm1.TextBox44.Visible = True
    UserForm1.TextBox43.Visible = True
    UserForm1.TextBox42.Visible = True
    UserForm1.TextBox41.Visible = True
End Select

UserForm1.TextBox62.value = s.ResourceCapacity(s.SymbolNumber("RscOilChange"))
UserForm1.TextBox62.Visible = True

Var_ExpressionTimeOC = s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_OilChange"))
Var_DistributionTimeOC = Left(Var_ExpressionTimeOC, 4)

Select Case Var_DistributionTimeOC
Case "Expo", "EXPO"
    UserForm1.Label91.Visible = False
    UserForm1.Label90.Visible = False
    UserForm1.TextBox59.Visible = False
    UserForm1.TextBox58.Visible = False
    UserForm1.TextBox61.value = "Exponencial"
    UserForm1.TextBox60.value = Val(Mid(Var_ExpressionTimeOC, 6))
    UserForm1.Label92.Caption = "Mean"
    UserForm1.Label92.Visible = True
    UserForm1.TextBox61.Visible = True
    UserForm1.TextBox60.Visible = True
Case "Pois", "POIS"
    UserForm1.Label91.Visible = False
    UserForm1.Label90.Visible = False
    UserForm1.TextBox59.Visible = False
    UserForm1.TextBox58.Visible = False
    UserForm1.TextBox61.value = "Poisson"
    UserForm1.TextBox60.value = Val(Mid(Var_ExpressionTimeOC, 6))
    UserForm1.Label92.Caption = "Mean"
    UserForm1.Label92.Visible = True
    UserForm1.TextBox61.Visible = True
    UserForm1.TextBox60.Visible = True
Case "Norm", "NORM"
    UserForm1.Label90.Visible = False
    UserForm1.TextBox58.Visible = False
    UserForm1.TextBox61.value = "Normal"

```

```

        UserForm1.TextBox60.value = Val(Mid(Var_ExpressionTimeOC, 6))
        UserForm1.TextBox59.value = Val(Mid(Var_ExpressionTimeOC,
InStr(Var_ExpressionTimeOC, ",") + 1))
        UserForm1.Label92.Caption = "Mean"
        UserForm1.Label92.Visible = True
        UserForm1.Label91.Caption = "StdDev"
        UserForm1.Label91.Visible = True
        UserForm1.TextBox61.Visible = True
        UserForm1.TextBox60.Visible = True
        UserForm1.TextBox59.Visible = True
    Case "Logn", "LOGN"
        UserForm1.Label90.Visible = False
        UserForm1.TextBox58.Visible = False
        UserForm1.TextBox61.value = "LogNormal"
        UserForm1.TextBox60.value = Val(Mid(Var_ExpressionTimeOC, 6))
        UserForm1.TextBox59.value = Val(Mid(Var_ExpressionTimeOC,
InStr(Var_ExpressionTimeOC, ",") + 1))
        UserForm1.Label92.Caption = "LogMean"
        UserForm1.Label92.Visible = True
        UserForm1.Label91.Caption = "LogStd"
        UserForm1.Label91.Visible = True
        UserForm1.TextBox61.Visible = True
        UserForm1.TextBox60.Visible = True
        UserForm1.TextBox59.Visible = True
    Case "Unif", "UNIF"
        UserForm1.Label90.Visible = False
        UserForm1.TextBox58.Visible = False
        UserForm1.TextBox61.value = "Uniform"
        UserForm1.TextBox60.value = Val(Mid(Var_ExpressionTimeOC, 6))
        UserForm1.TextBox59.value = Val(Mid(Var_ExpressionTimeOC,
InStr(Var_ExpressionTimeOC, ",") + 1))
        UserForm1.Label92.Caption = "Minimum"
        UserForm1.Label92.Visible = True
        UserForm1.Label91.Caption = "Maximum"
        UserForm1.Label91.Visible = True
        UserForm1.TextBox61.Visible = True
        UserForm1.TextBox60.Visible = True
        UserForm1.TextBox59.Visible = True
    Case "Tria", "TRIA"
        UserForm1.TextBox61.value = "Triangular"
        UserForm1.TextBox60.value = Val(Mid(Var_ExpressionTimeOC, 6))
        UserForm1.TextBox59.value = Val(Mid(Var_ExpressionTimeOC,
InStr(Var_ExpressionTimeOC, ",") + 1))
        UserForm1.TextBox58.value = Val(Mid(Var_ExpressionTimeOC,
InStr(InStr(Var_ExpressionTimeOC, ",") + 1, Var_ExpressionTimeOC, ",") + 1))
        UserForm1.Label92.Caption = "Minimum"
        UserForm1.Label92.Visible = True
        UserForm1.Label91.Caption = "Mode"
        UserForm1.Label91.Visible = True
        UserForm1.Label90.Caption = "Maximum"
        UserForm1.Label90.Visible = True
        UserForm1.TextBox61.Visible = True
        UserForm1.TextBox60.Visible = True
        UserForm1.TextBox59.Visible = True
        UserForm1.TextBox58.Visible = True
End Select

UserForm1.TextBox47.Visible = True
UserForm1.TextBox54.Visible = True
UserForm1.TextBox57.Visible = True
UserForm1.TextBox66.Visible = True
UserForm1.ComboBox4.Visible = True
UserForm1.ComboBox5.Visible = True
UserForm1.ComboBox6.Visible = True
UserForm1.ComboBox7.Visible = True

```

```

End Sub

Private Sub CommandButton8_Click()

Dim m As Model
Dim s As SIMAN

Set m = ThisDocument.Model
Set s = m.SIMAN

Var_Combobox7Index = UserForm1.ComboBox7.ListIndex

Select Case Var_Combobox7Index
    Case 1
        Var_TempTextBox66 = Val(UserForm1.TextBox66.value)
        var_temptextbox65 = Val(UserForm1.TextBox65.value)
        If Var_TempTextBox66 <> 0 And var_temptextbox65 <> 0 Then
            Var_TempConstructOC = "Expo" & "(" & var_temptextbox65 & ")"
            s.ResourceCapacity(s.SymbolNumber("RscOilChange")) = Var_TempTextBox66
            s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_OilChange")) =
Var_TempConstructOC
        Else
            MsgBox "Please insert values in all text boxes"
        End If
    Case 2
        Var_TempTextBox66 = Val(UserForm1.TextBox66.value)
        var_temptextbox65 = Val(UserForm1.TextBox65.value)
        If Var_TempTextBox66 <> 0 And var_temptextbox65 <> 0 Then
            Var_TempConstructOC = "Pois" & "(" & var_temptextbox65 & ")"
            s.ResourceCapacity(s.SymbolNumber("RscOilChange")) = Var_TempTextBox66
            s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_OilChange")) =
Var_TempConstructOC
        Else
            MsgBox "Please insert values in all text boxes"
        End If
    Case 3
        Var_TempTextBox66 = Val(UserForm1.TextBox66.value)
        var_temptextbox65 = Val(UserForm1.TextBox65.value)
        var_temptextbox64 = Val(UserForm1.TextBox64.value)
        If Var_TempTextBox66 <> 0 And var_temptextbox65 <> 0 And var_temptextbox64 <> 0 Then
            Var_TempConstructOC = "Norm" & "(" & var_temptextbox65 & "," & var_temptextbox64
& ")"
            s.ResourceCapacity(s.SymbolNumber("RscOilChange")) = Var_TempTextBox66
            s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_OilChange")) =
Var_TempConstructOC
        Else
            MsgBox "Please insert values in all text boxes"
        End If
    Case 4
        Var_TempTextBox66 = Val(UserForm1.TextBox66.value)
        var_temptextbox65 = Val(UserForm1.TextBox65.value)
        var_temptextbox64 = Val(UserForm1.TextBox64.value)
        If Var_TempTextBox66 <> 0 And var_temptextbox65 <> 0 And var_temptextbox64 <> 0 Then
            Var_TempConstructOC = "Logn" & "(" & var_temptextbox65 & "," & var_temptextbox64
& ")"
            s.ResourceCapacity(s.SymbolNumber("RscOilChange")) = Var_TempTextBox66
            s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_OilChange")) =
Var_TempConstructOC
        Else
            MsgBox "Please insert values in all text boxes"
        End If
    Case 5
        Var_TempTextBox66 = Val(UserForm1.TextBox66.value)
        var_temptextbox65 = Val(UserForm1.TextBox65.value)
        var_temptextbox64 = Val(UserForm1.TextBox64.value)
        If Var_TempTextBox66 <> 0 And var_temptextbox65 <> 0 And var_temptextbox64 <> 0 Then
            If (var_temptextbox65 < var_temptextbox64) Then

```

```

        Var_TempConstructOC = "Unif" & "(" & var temptextbox65 & "," &
var temptextbox64 & ")"
        s.ResourceCapacity(s.SymbolNumber("RscOilChange")) = Var_TempTextBox66
        s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_OilChange")) =
Var_TempConstructOC
    Else
        MsgBox "Minimum value should be inferior than maximum"
    End If
    Else
        MsgBox "Please insert values in all text boxes"
    End If
    Case 6
        Var_TempTextBox66 = Val(UserForm1.TextBox66.value)
        var temptextbox65 = Val(UserForm1.TextBox65.value)
        var temptextbox64 = Val(UserForm1.TextBox64.value)
        var temptextbox63 = Val(UserForm1.TextBox63.value)
        If Var_TempTextBox66 <> 0 And var temptextbox65 <> 0 And var temptextbox64 <> 0 And
var temptextbox63 <> 0 Then
            If (var temptextbox65 < var temptextbox64) And (var temptextbox64 <
var temptextbox63) Then
                Var_TempConstructOC = "Tria" & "(" & var temptextbox65 & "," &
var temptextbox64 & "," & var temptextbox63 & ")"
                s.ResourceCapacity(s.SymbolNumber("RscOilChange")) = Var_TempTextBox66
                s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_OilChange")) =
Var_TempConstructOC
            Else
                MsgBox "Please review Minimum, Mode and Maximum values"
            End If
        Else
            MsgBox "Please insert values in all text boxes"
        End If
    Case Else
        MsgBox "Please confirm distribution selection"
    End Select
End Sub

Private Sub CommandButton9_Click()

Dim m As Model
Dim s As SIMAN

Set m = ThisDocument.Model
Set s = m.SIMAN

Var_Combobox6Index = UserForm1.Combobox6.ListIndex

Select Case Var_Combobox6Index
    Case 1

        Var_TempTextBox57 = Val(UserForm1.TextBox57.value)
        var temptextbox56 = Val(UserForm1.TextBox56.value)
        If Var_TempTextBox57 <> 0 And var temptextbox56 <> 0 Then
            Var_TempConstructQS = "Expo" & "(" & var temptextbox56 & ")"
            s.ResourceCapacity(s.SymbolNumber("RscQuickServices")) = Var_TempTextBox57
            s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_QuickServices")) =
Var_TempConstructQS
        Else
            MsgBox "Please insert values in all text boxes"
        End If
    Case 2
        Var_TempTextBox57 = Val(UserForm1.TextBox57.value)
        var temptextbox56 = Val(UserForm1.TextBox56.value)
        If Var_TempTextBox57 <> 0 And var temptextbox56 <> 0 Then
            Var_TempConstructQS = "Pois" & "(" & var temptextbox56 & ")"

```

```

        s.ResourceCapacity(s.SymbolNumber("RscQuickServices")) = Var_TempTextBox57
        s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_QuickServices")) =
Var_TempConstructQS
    Else
        MsgBox "Please insert values in all text boxes"
    End If
Case 3
    Var_TempTextBox57 = Val(UserForm1.TextBox57.value)
    var_temptextbox56 = Val(UserForm1.TextBox56.value)
    var_temptextbox55 = Val(UserForm1.TextBox55.value)
    If Var_TempTextBox57 <> 0 And var_temptextbox56 <> 0 And var_temptextbox55 <> 0 Then
        Var_TempConstructQS = "Norm" & "(" & var_temptextbox56 & "," & var_temptextbox55
& ")"
        s.ResourceCapacity(s.SymbolNumber("RscQuickServices")) = Var_TempTextBox57
        s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_QuickServices")) =
Var_TempConstructQS
    Else
        MsgBox "Please insert values in all text boxes"
    End If
Case 4
    Var_TempTextBox57 = Val(UserForm1.TextBox57.value)
    var_temptextbox56 = Val(UserForm1.TextBox56.value)
    var_temptextbox55 = Val(UserForm1.TextBox55.value)
    If Var_TempTextBox57 <> 0 And var_temptextbox56 <> 0 And var_temptextbox55 <> 0 Then
        Var_TempConstructQS = "Logn" & "(" & var_temptextbox56 & "," & var_temptextbox55
& ")"
        s.ResourceCapacity(s.SymbolNumber("RscQuickServices")) = Var_TempTextBox57
        s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_QuickServices")) =
Var_TempConstructQS
    Else
        MsgBox "Please insert values in all text boxes"
    End If
Case 5
    Var_TempTextBox57 = Val(UserForm1.TextBox57.value)
    var_temptextbox56 = Val(UserForm1.TextBox56.value)
    var_temptextbox55 = Val(UserForm1.TextBox55.value)
    If Var_TempTextBox57 <> 0 And var_temptextbox56 <> 0 And var_temptextbox55 <> 0 Then
        If (var_temptextbox56 < var_temptextbox55) Then
            Var_TempConstructQS = "Unif" & "(" & var_temptextbox56 & "," &
var_temptextbox55 & ")"
            s.ResourceCapacity(s.SymbolNumber("RscQuickServices")) = Var_TempTextBox57
            s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_QuickServices")) =
Var_TempConstructQS
        Else
            MsgBox "Minimum value should be inferior than maximum"
        End If
    Else
        MsgBox "Please insert values in all text boxes"
    End If
Case 6
    Var_TempTextBox57 = Val(UserForm1.TextBox57.value)
    var_temptextbox56 = Val(UserForm1.TextBox56.value)
    var_temptextbox55 = Val(UserForm1.TextBox55.value)
    var_temptextbox46 = Val(UserForm1.TextBox46.value)
    If Var_TempTextBox57 <> 0 And var_temptextbox56 <> 0 And var_temptextbox55 <> 0 And
var_temptextbox46 <> 0 Then
        If (var_temptextbox56 < var_temptextbox55) And (var_temptextbox55 <
var_temptextbox46) Then
            Var_TempConstructQS = "Tria" & "(" & var_temptextbox56 & "," &
var_temptextbox55 & "," & var_temptextbox46 & ")"
            s.ResourceCapacity(s.SymbolNumber("RscQuickServices")) = Var_TempTextBox57
            s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_QuickServices")) =
Var_TempConstructQS
        Else
            MsgBox "Please review Minimum, Mode and Maximum values"
        End If
    Else

```

```
        MsgBox "Please insert values in all text boxes"
    End If
    Case Else
        MsgBox "Please confirm distribution selection"
    End Select
```

```
End Sub
```

```
Private Sub MultiPage1_Change()
```

```
UserForm1.Label7.Visible = False
UserForm1.Label8.Visible = False
UserForm1.Label9.Visible = False
UserForm1.Label10.Visible = False
UserForm1.Label11.Visible = False
UserForm1.Label15.Visible = False
UserForm1.Label16.Visible = False
UserForm1.Label17.Visible = False
UserForm1.Label27.Visible = False
UserForm1.Label12.Visible = False
UserForm1.Label28.Visible = False
UserForm1.Label29.Visible = False
UserForm1.Label30.Visible = False
UserForm1.Label31.Visible = False
UserForm1.Label32.Visible = False
UserForm1.Label35.Visible = False
UserForm1.Label36.Visible = False
UserForm1.Label37.Visible = False
```

```
UserForm1.TextBox1.Visible = False
UserForm1.TextBox2.Visible = False
UserForm1.TextBox3.Visible = False
UserForm1.TextBox4.Visible = False
UserForm1.TextBox5.Visible = False
UserForm1.TextBox6.Visible = False
UserForm1.TextBox7.Visible = False
UserForm1.TextBox8.Visible = False
UserForm1.TextBox9.Visible = False
UserForm1.TextBox10.Visible = False
UserForm1.TextBox11.Visible = False
UserForm1.TextBox12.Visible = False
UserForm1.TextBox13.Visible = False
UserForm1.TextBox14.Visible = False
UserForm1.TextBox15.Visible = False
UserForm1.TextBox16.Visible = False
UserForm1.TextBox17.Visible = False
UserForm1.TextBox19.Visible = False
UserForm1.TextBox20.Visible = False
UserForm1.TextBox21.Visible = False
UserForm1.TextBox22.Visible = False
UserForm1.TextBox23.Visible = False
UserForm1.TextBox24.Visible = False
UserForm1.TextBox26.Visible = False
UserForm1.TextBox27.Visible = False
UserForm1.TextBox28.Visible = False
UserForm1.TextBox30.Visible = False
UserForm1.ComboBox1.Visible = False
UserForm1.ComboBox2.Visible = False
UserForm1.ComboBox3.Visible = False
```

```
UserForm1.CommandButton4.Visible = False
UserForm1.CommandButton5.Visible = False
UserForm1.CommandButton6.Visible = False
```

```
UserForm1.TextBox1.value = ""
UserForm1.TextBox2.value = ""
```

```
UserForm1.TextBox3.value = ""
UserForm1.TextBox4.value = ""
UserForm1.TextBox5.value = ""
UserForm1.TextBox6.value = ""
UserForm1.TextBox7.value = ""
UserForm1.TextBox8.value = ""
UserForm1.TextBox9.value = ""
UserForm1.TextBox10.value = ""
UserForm1.TextBox11.value = ""
UserForm1.TextBox12.value = ""
UserForm1.TextBox13.value = ""
UserForm1.TextBox14.value = ""
UserForm1.TextBox15.value = ""
UserForm1.TextBox16.value = ""
UserForm1.TextBox17.value = ""
UserForm1.TextBox19.value = ""
UserForm1.TextBox20.value = ""
UserForm1.TextBox21.value = ""
UserForm1.TextBox22.value = ""
UserForm1.TextBox23.value = ""
UserForm1.TextBox24.value = ""
UserForm1.TextBox26.value = ""
UserForm1.TextBox27.value = ""
UserForm1.TextBox28.value = ""
UserForm1.TextBox30.value = ""
```

```
Var_Combobox1 = UserForm1.Combobox1.ListIndex
```

```
Select Case Var_Combobox1
  Case -1
    Var_Combobox1Existe = 0
  Case 0
    Var_Combobox1Existe = 1
  Case 1
    Var_Combobox1Existe = 1
  Case 2
    Var_Combobox1Existe = 1
  Case 3
    Var_Combobox1Existe = 1
  Case 4
    Var_Combobox1Existe = 1
  Case 5
    Var_Combobox1Existe = 1
  Case 6
    Var_Combobox1Existe = 1
  Case Else
    Var_Combobox1Existe = 0
End Select
```

```
Var_Combobox2 = UserForm1.Combobox2.ListIndex
```

```
Select Case Var_Combobox2
  Case -1
    Var_Combobox2Existe = 0
  Case 0
    Var_Combobox2Existe = 1
  Case 1
    Var_Combobox2Existe = 1
  Case 2
    Var_Combobox2Existe = 1
  Case 3
    Var_Combobox2Existe = 1
  Case 4
    Var_Combobox2Existe = 1
  Case 5
    Var_Combobox2Existe = 1
  Case 6
    Var_Combobox2Existe = 1
```

```

        Case Else
            Var_Combobox2Existe = 0
        End Select

Var_Combobox3 = UserForm1.Combobox3.ListIndex
Select Case Var_Combobox3
    Case -1
        Var_Combobox3Existe = 0
    Case 0
        Var_Combobox3Existe = 1
    Case 1
        Var_Combobox3Existe = 1
    Case 2
        Var_Combobox3Existe = 1
    Case 3
        Var_Combobox3Existe = 1
    Case 4
        Var_Combobox3Existe = 1
    Case 5
        Var_Combobox3Existe = 1
    Case 6
        Var_Combobox3Existe = 1
    Case Else
        Var_Combobox3Existe = 0
End Select

If Var_Combobox1Existe = 0 Then
    UserForm1.Combobox1.AddItem ""
    UserForm1.Combobox1.AddItem "Exponencial"
    UserForm1.Combobox1.AddItem "Poisson"
    UserForm1.Combobox1.AddItem "Normal"
    UserForm1.Combobox1.AddItem "LogNormal"
    UserForm1.Combobox1.AddItem "Uniform"
    UserForm1.Combobox1.AddItem "Triangular"
    UserForm1.Combobox1.ListIndex = 0
    Var_Combobox1Existe = 1
End If

If Var_Combobox2Existe = 0 Then
    UserForm1.Combobox2.AddItem ""
    UserForm1.Combobox2.AddItem "Exponencial"
    UserForm1.Combobox2.AddItem "Poisson"
    UserForm1.Combobox2.AddItem "Normal"
    UserForm1.Combobox2.AddItem "LogNormal"
    UserForm1.Combobox2.AddItem "Uniform"
    UserForm1.Combobox2.AddItem "Triangular"
    stIndex = 6
    UserForm1.Combobox2.ListIndex = 0
    Var_Combobox2Existe = 1
End If

If Var_Combobox3Existe = 0 Then
    UserForm1.Combobox3.AddItem ""
    UserForm1.Combobox3.AddItem "Exponencial"
    UserForm1.Combobox3.AddItem "Poisson"
    UserForm1.Combobox3.AddItem "Normal"
    UserForm1.Combobox3.AddItem "LogNormal"
    UserForm1.Combobox3.AddItem "Uniform"
    UserForm1.Combobox3.AddItem "Triangular"
    UserForm1.Combobox3.ListIndex = 0
    Var_Combobox3Existe = 1
End If

UserForm1.Label151.Visible = False
UserForm1.Label152.Visible = False
UserForm1.Label153.Visible = False

```

```
UserForm1.Label54.Visible = False
UserForm1.Label55.Visible = False
UserForm1.Label56.Visible = False
UserForm1.Label59.Visible = False
UserForm1.Label60.Visible = False
UserForm1.Label61.Visible = False
UserForm1.Label71.Visible = False
UserForm1.Label72.Visible = False
UserForm1.Label73.Visible = False
UserForm1.Label74.Visible = False
UserForm1.Label75.Visible = False
UserForm1.Label76.Visible = False
UserForm1.Label79.Visible = False
UserForm1.Label80.Visible = False
UserForm1.Label81.Visible = False
UserForm1.Label90.Visible = False
UserForm1.Label91.Visible = False
UserForm1.Label92.Visible = False
UserForm1.Label96.Visible = False
UserForm1.Label97.Visible = False
UserForm1.Label98.Visible = False
```

```
UserForm1.TextBox31.Visible = False
UserForm1.TextBox32.Visible = False
UserForm1.TextBox33.Visible = False
UserForm1.TextBox34.Visible = False
UserForm1.TextBox35.Visible = False
UserForm1.TextBox36.Visible = False
UserForm1.TextBox37.Visible = False
UserForm1.TextBox38.Visible = False
UserForm1.TextBox39.Visible = False
UserForm1.TextBox40.Visible = False
UserForm1.TextBox41.Visible = False
UserForm1.TextBox42.Visible = False
UserForm1.TextBox43.Visible = False
UserForm1.TextBox44.Visible = False
UserForm1.TextBox45.Visible = False
UserForm1.TextBox46.Visible = False
UserForm1.TextBox47.Visible = False
UserForm1.TextBox48.Visible = False
UserForm1.TextBox49.Visible = False
UserForm1.TextBox50.Visible = False
UserForm1.TextBox51.Visible = False
UserForm1.TextBox52.Visible = False
UserForm1.TextBox53.Visible = False
UserForm1.TextBox54.Visible = False
UserForm1.TextBox55.Visible = False
UserForm1.TextBox56.Visible = False
UserForm1.TextBox57.Visible = False
UserForm1.TextBox58.Visible = False
UserForm1.TextBox59.Visible = False
UserForm1.TextBox60.Visible = False
UserForm1.TextBox61.Visible = False
UserForm1.TextBox62.Visible = False
UserForm1.TextBox63.Visible = False
UserForm1.TextBox64.Visible = False
UserForm1.TextBox65.Visible = False
UserForm1.TextBox66.Visible = False
UserForm1.ComboBox4.Visible = False
UserForm1.ComboBox5.Visible = False
UserForm1.ComboBox6.Visible = False
UserForm1.ComboBox7.Visible = False
```

```
UserForm1.CommandButton8.Visible = False
UserForm1.CommandButton9.Visible = False
UserForm1.CommandButton10.Visible = False
UserForm1.CommandButton11.Visible = False
```

```
UserForm1.TextBox31.value = ""
UserForm1.TextBox32.value = ""
UserForm1.TextBox33.value = ""
UserForm1.TextBox34.value = ""
UserForm1.TextBox35.value = ""
UserForm1.TextBox36.value = ""
UserForm1.TextBox37.value = ""
UserForm1.TextBox38.value = ""
UserForm1.TextBox39.value = ""
UserForm1.TextBox40.value = ""
UserForm1.TextBox41.value = ""
UserForm1.TextBox42.value = ""
UserForm1.TextBox43.value = ""
UserForm1.TextBox44.value = ""
UserForm1.TextBox45.value = ""
UserForm1.TextBox46.value = ""
UserForm1.TextBox47.value = ""
UserForm1.TextBox48.value = ""
UserForm1.TextBox49.value = ""
UserForm1.TextBox50.value = ""
UserForm1.TextBox51.value = ""
UserForm1.TextBox52.value = ""
UserForm1.TextBox53.value = ""
UserForm1.TextBox54.value = ""
UserForm1.TextBox55.value = ""
UserForm1.TextBox56.value = ""
UserForm1.TextBox57.value = ""
UserForm1.TextBox58.value = ""
UserForm1.TextBox59.value = ""
UserForm1.TextBox60.value = ""
UserForm1.TextBox61.value = ""
UserForm1.TextBox62.value = ""
UserForm1.TextBox63.value = ""
UserForm1.TextBox64.value = ""
UserForm1.TextBox65.value = ""
UserForm1.TextBox66.value = ""
```

```
Var_ComboBox4 = UserForm1.ComboBox4.ListIndex
Select Case Var_ComboBox4
    Case -1
        Var_ComboBox4Existe = 0
    Case 0
        Var_ComboBox4Existe = 1
    Case 1
        Var_ComboBox4Existe = 1
    Case 2
        Var_ComboBox4Existe = 1
    Case 3
        Var_ComboBox4Existe = 1
    Case 4
        Var_ComboBox4Existe = 1
    Case 5
        Var_ComboBox4Existe = 1
    Case 6
        Var_ComboBox4Existe = 1
    Case Else
        Var_ComboBox4Existe = 0
End Select
```

```
Var_ComboBox5 = UserForm1.ComboBox5.ListIndex
Select Case Var_ComboBox5
    Case -1
        Var_ComboBox5Existe = 0
    Case 0
        Var_ComboBox5Existe = 1
```

```
Case 1
    Var_Combobox5Existe = 1
Case 2
    Var_Combobox5Existe = 1
Case 3
    Var_Combobox5Existe = 1
Case 4
    Var_Combobox5Existe = 1
Case 5
    Var_Combobox5Existe = 1
Case 6
    Var_Combobox5Existe = 1
Case Else
    Var_Combobox5Existe = 0
End Select

Var_Combobox6 = UserForm1.Combobox6.ListIndex
Select Case Var_Combobox6
Case -1
    Var_Combobox6Existe = 0
Case 0
    Var_Combobox6Existe = 1
Case 1
    Var_Combobox6Existe = 1
Case 2
    Var_Combobox6Existe = 1
Case 3
    Var_Combobox6Existe = 1
Case 4
    Var_Combobox6Existe = 1
Case 5
    Var_Combobox6Existe = 1
Case 6
    Var_Combobox6Existe = 1
Case Else
    Var_Combobox6Existe = 0
End Select

Var_Combobox7 = UserForm1.Combobox7.ListIndex
Select Case Var_Combobox7
Case -1
    Var_Combobox7Existe = 0
Case 0
    Var_Combobox7Existe = 1
Case 1
    Var_Combobox7Existe = 1
Case 2
    Var_Combobox7Existe = 1
Case 3
    Var_Combobox7Existe = 1
Case 4
    Var_Combobox7Existe = 1
Case 5
    Var_Combobox7Existe = 1
Case 6
    Var_Combobox7Existe = 1
Case Else
    Var_Combobox7Existe = 0
End Select
)
If Var_Combobox4Existe = 0 Then
    UserForm1.Combobox4.AddItem ""
    UserForm1.Combobox4.AddItem "Exponencial"
    UserForm1.Combobox4.AddItem "Poisson"
    UserForm1.Combobox4.AddItem "Normal"
    UserForm1.Combobox4.AddItem "LogNormal"
    UserForm1.Combobox4.AddItem "Uniform"
```

```
UserForm1.ComboBox4.AddItem "Triangular"
UserForm1.ComboBox4.ListIndex = 0
Var_ComboBox4Existe = 1
End If

If Var_ComboBox5Existe = 0 Then
    UserForm1.ComboBox5.AddItem ""
    UserForm1.ComboBox5.AddItem "Exponencial"
    UserForm1.ComboBox5.AddItem "Poisson"
    UserForm1.ComboBox5.AddItem "Normal"
    UserForm1.ComboBox5.AddItem "LogNormal"
    UserForm1.ComboBox5.AddItem "Uniform"
    UserForm1.ComboBox5.AddItem "Triangular"
    UserForm1.ComboBox5.ListIndex = 0
    Var_ComboBox5Existe = 1
End If

If Var_ComboBox6Existe = 0 Then
    UserForm1.ComboBox6.AddItem ""
    UserForm1.ComboBox6.AddItem "Exponencial"
    UserForm1.ComboBox6.AddItem "Poisson"
    UserForm1.ComboBox6.AddItem "Normal"
    UserForm1.ComboBox6.AddItem "LogNormal"
    UserForm1.ComboBox6.AddItem "Uniform"
    UserForm1.ComboBox6.AddItem "Triangular"
    UserForm1.ComboBox6.ListIndex = 0
    Var_ComboBox6Existe = 1
End If

If Var_ComboBox7Existe = 0 Then
    UserForm1.ComboBox7.AddItem ""
    UserForm1.ComboBox7.AddItem "Exponencial"
    UserForm1.ComboBox7.AddItem "Poisson"
    UserForm1.ComboBox7.AddItem "Normal"
    UserForm1.ComboBox7.AddItem "LogNormal"
    UserForm1.ComboBox7.AddItem "Uniform"
    UserForm1.ComboBox7.AddItem "Triangular"
    UserForm1.ComboBox7.ListIndex = 0
    Var_ComboBox7Existe = 1
End If

UserForm1.Label107.Visible = False
UserForm1.Label108.Visible = False
UserForm1.Label109.Visible = False
UserForm1.Label111.Visible = False
UserForm1.Label112.Visible = False
UserForm1.Label113.Visible = False
UserForm1.Label117.Visible = False
UserForm1.Label118.Visible = False
UserForm1.Label119.Visible = False
UserForm1.Label123.Visible = False
UserForm1.Label124.Visible = False
UserForm1.Label125.Visible = False
UserForm1.Label132.Visible = False
UserForm1.Label133.Visible = False
UserForm1.Label134.Visible = False
UserForm1.Label136.Visible = False
UserForm1.Label137.Visible = False
UserForm1.Label138.Visible = False
UserForm1.Label142.Visible = False
UserForm1.Label143.Visible = False
UserForm1.Label144.Visible = False
UserForm1.Label148.Visible = False
UserForm1.Label149.Visible = False
UserForm1.Label150.Visible = False

UserForm1.TextBox68.Visible = False
```

```
UserForm1.TextBox69.Visible = False
UserForm1.TextBox70.Visible = False
UserForm1.TextBox71.Visible = False
UserForm1.TextBox72.Visible = False
UserForm1.TextBox73.Visible = False
UserForm1.TextBox74.Visible = False
UserForm1.TextBox75.Visible = False
UserForm1.TextBox76.Visible = False
UserForm1.TextBox77.Visible = False
UserForm1.TextBox78.Visible = False
UserForm1.TextBox79.Visible = False
UserForm1.TextBox80.Visible = False
UserForm1.TextBox81.Visible = False
UserForm1.TextBox82.Visible = False
UserForm1.TextBox83.Visible = False
UserForm1.TextBox84.Visible = False
UserForm1.TextBox86.Visible = False
UserForm1.TextBox87.Visible = False
UserForm1.TextBox88.Visible = False
UserForm1.TextBox89.Visible = False
UserForm1.TextBox90.Visible = False
UserForm1.TextBox91.Visible = False
UserForm1.TextBox93.Visible = False
UserForm1.TextBox94.Visible = False
UserForm1.TextBox97.Visible = False
UserForm1.TextBox98.Visible = False
UserForm1.ComboBox8.Visible = False
UserForm1.ComboBox9.Visible = False
UserForm1.ComboBox10.Visible = False
UserForm1.ComboBox11.Visible = False
```

```
UserForm1.CommandButton12.Visible = False
UserForm1.CommandButton13.Visible = False
UserForm1.CommandButton14.Visible = False
UserForm1.CommandButton15.Visible = False
```

```
UserForm1.TextBox68.value = ""
UserForm1.TextBox69.value = ""
UserForm1.TextBox70.value = ""
UserForm1.TextBox71.value = ""
UserForm1.TextBox72.value = ""
UserForm1.TextBox73.value = ""
UserForm1.TextBox74.value = ""
UserForm1.TextBox75.value = ""
UserForm1.TextBox76.value = ""
UserForm1.TextBox77.value = ""
UserForm1.TextBox78.value = ""
UserForm1.TextBox79.value = ""
UserForm1.TextBox80.value = ""
UserForm1.TextBox81.value = ""
UserForm1.TextBox82.value = ""
UserForm1.TextBox83.value = ""
UserForm1.TextBox84.value = ""
UserForm1.TextBox86.value = ""
UserForm1.TextBox87.value = ""
UserForm1.TextBox88.value = ""
UserForm1.TextBox89.value = ""
UserForm1.TextBox90.value = ""
UserForm1.TextBox91.value = ""
UserForm1.TextBox93.value = ""
UserForm1.TextBox94.value = ""
UserForm1.TextBox97.value = ""
UserForm1.TextBox98.value = ""
```

```
Var_ComboBox8 = UserForm1.ComboBox8.ListIndex
Select Case Var_ComboBox8
    Case -1
```

```
        Var_Combobox8Existe = 0
    Case 0
        Var_Combobox8Existe = 1
    Case 1
        Var_Combobox8Existe = 1
    Case 2
        Var_Combobox8Existe = 1
    Case 3
        Var_Combobox8Existe = 1
    Case 4
        Var_Combobox8Existe = 1
    Case 5
        Var_Combobox8Existe = 1
    Case 6
        Var_Combobox8Existe = 1
    Case Else
        Var_Combobox8Existe = 0
End Select

Var_Combobox9 = UserForm1.Combobox9.ListIndex
Select Case Var_Combobox9
    Case -1
        Var_Combobox9Existe = 0
    Case 0
        Var_Combobox9Existe = 1
    Case 1
        Var_Combobox9Existe = 1
    Case 2
        Var_Combobox9Existe = 1
    Case 3
        Var_Combobox9Existe = 1
    Case 4
        Var_Combobox9Existe = 1
    Case 5
        Var_Combobox9Existe = 1
    Case 6
        Var_Combobox9Existe = 1
    Case Else
        Var_Combobox9Existe = 0
End Select

Var_Combobox10 = UserForm1.Combobox10.ListIndex
Select Case Var_Combobox10
    Case -1
        Var_Combobox10Existe = 0
    Case 0
        Var_Combobox10Existe = 1
    Case 1
        Var_Combobox10Existe = 1
    Case 2
        Var_Combobox10Existe = 1
    Case 3
        Var_Combobox10Existe = 1
    Case 4
        Var_Combobox10Existe = 1
    Case 5
        Var_Combobox10Existe = 1
    Case 6
        Var_Combobox10Existe = 1
    Case Else
        Var_Combobox10Existe = 0
End Select

Var_Combobox11 = UserForm1.Combobox11.ListIndex
Select Case Var_Combobox11
    Case -1
        Var_Combobox11Existe = 0
```

```
Case 0
    Var_Combobox11Existe = 1
Case 1
    Var_Combobox11Existe = 1
Case 2
    Var_Combobox11Existe = 1
Case 3
    Var_Combobox11Existe = 1
Case 4
    Var_Combobox11Existe = 1
Case 5
    Var_Combobox11Existe = 1
Case 6
    Var_Combobox11Existe = 1
Case Else
    Var_Combobox11Existe = 0
End Select

If Var_Combobox8Existe = 0 Then
    UserForm1.Combobox8.AddItem ""
    UserForm1.Combobox8.AddItem "Exponencial"
    UserForm1.Combobox8.AddItem "Poisson"
    UserForm1.Combobox8.AddItem "Normal"
    UserForm1.Combobox8.AddItem "LogNormal"
    UserForm1.Combobox8.AddItem "Uniform"
    UserForm1.Combobox8.AddItem "Triangular"
    UserForm1.Combobox8.ListIndex = 0
    Var_Combobox8Existe = 1
End If

If Var_Combobox9Existe = 0 Then
    UserForm1.Combobox9.AddItem ""
    UserForm1.Combobox9.AddItem "Exponencial"
    UserForm1.Combobox9.AddItem "Poisson"
    UserForm1.Combobox9.AddItem "Normal"
    UserForm1.Combobox9.AddItem "LogNormal"
    UserForm1.Combobox9.AddItem "Uniform"
    UserForm1.Combobox9.AddItem "Triangular"
    UserForm1.Combobox9.ListIndex = 0
    Var_Combobox9Existe = 1
End If

If Var_Combobox10Existe = 0 Then
    UserForm1.Combobox10.AddItem ""
    UserForm1.Combobox10.AddItem "Exponencial"
    UserForm1.Combobox10.AddItem "Poisson"
    UserForm1.Combobox10.AddItem "Normal"
    UserForm1.Combobox10.AddItem "LogNormal"
    UserForm1.Combobox10.AddItem "Uniform"
    UserForm1.Combobox10.AddItem "Triangular"
    UserForm1.Combobox10.ListIndex = 0
    Var_Combobox10Existe = 1
End If

If Var_Combobox11Existe = 0 Then
    UserForm1.Combobox11.AddItem ""
    UserForm1.Combobox11.AddItem "Exponencial"
    UserForm1.Combobox11.AddItem "Poisson"
    UserForm1.Combobox11.AddItem "Normal"
    UserForm1.Combobox11.AddItem "LogNormal"
    UserForm1.Combobox11.AddItem "Uniform"
    UserForm1.Combobox11.AddItem "Triangular"
    UserForm1.Combobox11.ListIndex = 0
    Var_Combobox11Existe = 1
End If

UserForm1.TextBox99.Visible = False
```

```
UserForm1.TextBox100.Visible = False
UserForm1.TextBox101.Visible = False
UserForm1.TextBox102.Visible = False
UserForm1.TextBox103.Visible = False
UserForm1.TextBox104.Visible = False
UserForm1.TextBox105.Visible = False
UserForm1.TextBox106.Visible = False

UserForm1.TextBox107.Visible = False
UserForm1.TextBox108.Visible = False
UserForm1.TextBox109.Visible = False
UserForm1.TextBox110.Visible = False
UserForm1.ComboBox12.Visible = False
UserForm1.ComboBox13.Visible = False
UserForm1.ComboBox14.Visible = False
UserForm1.ComboBox15.Visible = False

UserForm1.CommandButton18.Visible = False

UserForm1.TextBox99.value = ""
UserForm1.TextBox100.value = ""
UserForm1.TextBox101.value = ""
UserForm1.TextBox102.value = ""
UserForm1.TextBox103.value = ""
UserForm1.TextBox104.value = ""
UserForm1.TextBox105.value = ""
UserForm1.TextBox106.value = ""

Var_Combobox12 = UserForm1.ComboBox12.ListIndex
Select Case Var_Combobox12
    Case -1
        Var_Combobox12Existe = 0
    Case 0
        Var_Combobox12Existe = 1
    Case 1
        Var_Combobox12Existe = 1
    Case Else
        Var_Combobox12Existe = 0
End Select

Var_Combobox13 = UserForm1.ComboBox13.ListIndex
Select Case Var_Combobox13
    Case -1
        Var_Combobox13Existe = 0
    Case 0
        Var_Combobox13Existe = 1
    Case 1
        Var_Combobox13Existe = 1
    Case Else
        Var_Combobox13Existe = 0
End Select

Var_Combobox14 = UserForm1.ComboBox14.ListIndex
Select Case Var_Combobox14
    Case -1
        Var_Combobox14Existe = 0
    Case 0
        Var_Combobox14Existe = 1
    Case 1
        Var_Combobox14Existe = 1
    Case 2
        Var_Combobox14Existe = 1
    Case 3
        Var_Combobox14Existe = 1
    Case Else
        Var_Combobox14Existe = 0
End Select
```

```
Var_Combobox15 = UserForm1.Combobox15.ListIndex
Select Case Var_Combobox15
    Case -1
        Var_Combobox15Existe = 0
    Case 0
        Var_Combobox15Existe = 1
    Case 1
        Var_Combobox15Existe = 1
    Case 2
        Var_Combobox15Existe = 1
    Case 3
        Var_Combobox15Existe = 1
    Case Else
        Var_Combobox15Existe = 0
End Select

If Var_Combobox12Existe = 0 Then
    UserForm1.Combobox12.AddItem "True"
    UserForm1.Combobox12.AddItem "False"
    UserForm1.Combobox12.ListIndex = 0
    Var_Combobox12Existe = 1
End If

If Var_Combobox13Existe = 0 Then
    UserForm1.Combobox13.AddItem "True"
    UserForm1.Combobox13.AddItem "False"
    UserForm1.Combobox13.ListIndex = 0
    Var_Combobox13Existe = 1
End If

If Var_Combobox14Existe = 0 Then
    UserForm1.Combobox14.AddItem "Days"
    UserForm1.Combobox14.AddItem "Hours"
    UserForm1.Combobox14.AddItem "Minutes"
    UserForm1.Combobox14.AddItem "Seconds"
    UserForm1.Combobox14.ListIndex = 0
    Var_Combobox14Existe = 1
End If

If Var_Combobox15Existe = 0 Then
    UserForm1.Combobox15.AddItem "Days"
    UserForm1.Combobox15.AddItem "Hours"
    UserForm1.Combobox15.AddItem "Minutes"
    UserForm1.Combobox15.AddItem "Seconds"
    UserForm1.Combobox15.ListIndex = 0
    Var_Combobox15Existe = 1
End If

End Sub

Private Sub TextBox17_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

Select Case KeyAscii
    Case vbKey0 To vbKey9
    Case vbKeyBack, vbKeyClear, vbKeyDelete
    Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
    Case Else
        KeyAscii = 0
        Beep
End Select

End Sub
```

```
Private Sub TextBox19_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

Select Case KeyAscii
    Case vbKey0 To vbKey9
        Case vbKeyBack, vbKeyClear, vbKeyDelete
        Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
        Case Else
            KeyAscii = 0
            Beep
End Select

End Sub

Private Sub TextBox20_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

Select Case KeyAscii
    Case vbKey0 To vbKey9
        Case vbKeyBack, vbKeyClear, vbKeyDelete
        Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
        Case Else
            KeyAscii = 0
            Beep
End Select

End Sub

Private Sub TextBox21_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

Select Case KeyAscii
    Case vbKey0 To vbKey9
        Case vbKeyBack, vbKeyClear, vbKeyDelete
        Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
        Case Else
            KeyAscii = 0
            Beep
End Select

End Sub

Private Sub TextBox22_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

Select Case KeyAscii
    Case vbKey0 To vbKey9
        Case vbKeyBack, vbKeyClear, vbKeyDelete
        Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
        Case Else
            KeyAscii = 0
            Beep
End Select

End Sub

Private Sub TextBox23_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

Select Case KeyAscii
    Case vbKey0 To vbKey9
        Case vbKeyBack, vbKeyClear, vbKeyDelete
        Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
        Case Else
            KeyAscii = 0
            Beep
End Select

End Sub

Private Sub TextBox24_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
```

```
Select Case KeyAscii
  Case vbKey0 To vbKey9
  Case vbKeyBack, vbKeyClear, vbKeyDelete
  Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
  Case Else
    KeyAscii = 0
    Beep
End Select

End Sub

Private Sub TextBox26_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

Select Case KeyAscii
  Case vbKey0 To vbKey9
  Case vbKeyBack, vbKeyClear, vbKeyDelete
  Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
  Case Else
    KeyAscii = 0
    Beep
End Select

End Sub

Private Sub TextBox16_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

Select Case KeyAscii
  Case vbKey0 To vbKey9
  Case vbKeyBack, vbKeyClear, vbKeyDelete
  Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
  Case Else
    KeyAscii = 0
    Beep
End Select

End Sub

Private Sub TextBox27_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

Select Case KeyAscii
  Case vbKey0 To vbKey9
  Case vbKeyBack, vbKeyClear, vbKeyDelete
  Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
  Case Else
    KeyAscii = 0
    Beep
End Select

End Sub

Private Sub TextBox28_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

Select Case KeyAscii
  Case vbKey0 To vbKey9
  Case vbKeyBack, vbKeyClear, vbKeyDelete
  Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
  Case Else
    KeyAscii = 0
    Beep
End Select

End Sub

Private Sub TextBox30_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

Select Case KeyAscii
  Case vbKey0 To vbKey9
```

```
Case vbKeyBack, vbKeyClear, vbKeyDelete
Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
Case Else
    KeyAscii = 0
    Beep
End Select

End Sub

Private Sub TextBox47_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

Select Case KeyAscii
    Case vbKey0 To vbKey9
    Case vbKeyBack, vbKeyClear, vbKeyDelete
    Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
    Case Else
        KeyAscii = 0
        Beep
End Select

End Sub

Private Sub TextBox48_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

Select Case KeyAscii
    Case vbKey0 To vbKey9
    Case vbKeyBack, vbKeyClear, vbKeyDelete
    Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
    Case Else
        KeyAscii = 0
        Beep
End Select

End Sub

Private Sub TextBox49_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

Select Case KeyAscii
    Case vbKey0 To vbKey9
    Case vbKeyBack, vbKeyClear, vbKeyDelete
    Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
    Case Else
        KeyAscii = 0
        Beep
End Select

End Sub

Private Sub TextBox50_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

Select Case KeyAscii
    Case vbKey0 To vbKey9
    Case vbKeyBack, vbKeyClear, vbKeyDelete
    Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
    Case Else
        KeyAscii = 0
        Beep
End Select

End Sub

Private Sub TextBox54_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

Select Case KeyAscii
    Case vbKey0 To vbKey9
```

```
Case vbKeyBack, vbKeyClear, vbKeyDelete
Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
Case Else
    KeyAscii = 0
    Beep
End Select

End Sub

Private Sub TextBox53_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

Select Case KeyAscii
    Case vbKey0 To vbKey9
    Case vbKeyBack, vbKeyClear, vbKeyDelete
    Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
    Case Else
        KeyAscii = 0
        Beep
End Select

End Sub

Private Sub TextBox52_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

Select Case KeyAscii
    Case vbKey0 To vbKey9
    Case vbKeyBack, vbKeyClear, vbKeyDelete
    Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
    Case Else
        KeyAscii = 0
        Beep
End Select

End Sub

Private Sub TextBox51_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

Select Case KeyAscii
    Case vbKey0 To vbKey9
    Case vbKeyBack, vbKeyClear, vbKeyDelete
    Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
    Case Else
        KeyAscii = 0
        Beep
End Select

End Sub

Private Sub TextBox57_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

Select Case KeyAscii
    Case vbKey0 To vbKey9
    Case vbKeyBack, vbKeyClear, vbKeyDelete
    Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
    Case Else
        KeyAscii = 0
        Beep
End Select

End Sub

Private Sub TextBox56_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

Select Case KeyAscii
    Case vbKey0 To vbKey9
```

```
Case vbKeyBack, vbKeyClear, vbKeyDelete
Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
Case Else
    KeyAscii = 0
    Beep
End Select

End Sub

Private Sub TextBox55_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

Select Case KeyAscii
    Case vbKey0 To vbKey9
    Case vbKeyBack, vbKeyClear, vbKeyDelete
    Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
    Case Else
        KeyAscii = 0
        Beep
End Select

End Sub

Private Sub TextBox46_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

Select Case KeyAscii
    Case vbKey0 To vbKey9
    Case vbKeyBack, vbKeyClear, vbKeyDelete
    Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
    Case Else
        KeyAscii = 0
        Beep
End Select

End Sub

Private Sub TextBox66_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

Select Case KeyAscii
    Case vbKey0 To vbKey9
    Case vbKeyBack, vbKeyClear, vbKeyDelete
    Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
    Case Else
        KeyAscii = 0
        Beep
End Select

End Sub

Private Sub TextBox65_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

Select Case KeyAscii
    Case vbKey0 To vbKey9
    Case vbKeyBack, vbKeyClear, vbKeyDelete
    Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
    Case Else
        KeyAscii = 0
        Beep
End Select

End Sub

Private Sub TextBox64_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

Select Case KeyAscii
```

```
Case vbKey0 To vbKey9
Case vbKeyBack, vbKeyClear, vbKeyDelete
Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
Case Else
    KeyAscii = 0
    Beep
End Select

End Sub

Private Sub TextBox63_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

Select Case KeyAscii
    Case vbKey0 To vbKey9
    Case vbKeyBack, vbKeyClear, vbKeyDelete
    Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
    Case Else
        KeyAscii = 0
        Beep
End Select

End Sub

Private Sub TextBox86_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

Select Case KeyAscii
    Case vbKey0 To vbKey9
    Case vbKeyBack, vbKeyClear, vbKeyDelete
    Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
    Case Else
        KeyAscii = 0
        Beep
End Select

End Sub

Private Sub TextBox87_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

Select Case KeyAscii
    Case vbKey0 To vbKey9
    Case vbKeyBack, vbKeyClear, vbKeyDelete
    Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
    Case Else
        KeyAscii = 0
        Beep
End Select

End Sub

Private Sub TextBox88_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

Select Case KeyAscii
    Case vbKey0 To vbKey9
    Case vbKeyBack, vbKeyClear, vbKeyDelete
    Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
    Case Else
        KeyAscii = 0
        Beep
End Select

End Sub

Private Sub TextBox89_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

Select Case KeyAscii
    Case vbKey0 To vbKey9
```

```
Case vbKeyBack, vbKeyClear, vbKeyDelete
Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
Case Else
    KeyAscii = 0
    Beep
End Select

End Sub

Private Sub TextBox90_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

Select Case KeyAscii
    Case vbKey0 To vbKey9
    Case vbKeyBack, vbKeyClear, vbKeyDelete
    Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
    Case Else
        KeyAscii = 0
        Beep
End Select

End Sub

Private Sub TextBox91_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

Select Case KeyAscii
    Case vbKey0 To vbKey9
    Case vbKeyBack, vbKeyClear, vbKeyDelete
    Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
    Case Else
        KeyAscii = 0
        Beep
End Select

End Sub

Private Sub TextBox93_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

Select Case KeyAscii
    Case vbKey0 To vbKey9
    Case vbKeyBack, vbKeyClear, vbKeyDelete
    Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
    Case Else
        KeyAscii = 0
        Beep
End Select

End Sub

Private Sub TextBox94_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)

Select Case KeyAscii
    Case vbKey0 To vbKey9
    Case vbKeyBack, vbKeyClear, vbKeyDelete
    Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
    Case Else
        KeyAscii = 0
        Beep
End Select

End Sub
```

```
Private Sub TextBox95_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
```

```
    Select Case KeyAscii
        Case vbKey0 To vbKey9
        Case vbKeyBack, vbKeyClear, vbKeyDelete
        Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
        Case Else
            KeyAscii = 0
            Beep
    End Select
```

```
End Sub
```

```
Private Sub TextBox98_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
```

```
    Select Case KeyAscii
        Case vbKey0 To vbKey9
        Case vbKeyBack, vbKeyClear, vbKeyDelete
        Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
        Case Else
            KeyAscii = 0
            Beep
    End Select
```

```
End Sub
```

```
Private Sub TextBox97_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
```

```
    Select Case KeyAscii
        Case vbKey0 To vbKey9
        Case vbKeyBack, vbKeyClear, vbKeyDelete
        Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
        Case Else
            KeyAscii = 0
            Beep
    End Select
```

```
End Sub
```

```
Private Sub TextBox84_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
```

```
    Select Case KeyAscii
        Case vbKey0 To vbKey9
        Case vbKeyBack, vbKeyClear, vbKeyDelete
        Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
        Case Else
            KeyAscii = 0
            Beep
    End Select
```

```
End Sub
```

```
Private Sub TextBox107_KeyPress(ByVal KeyAscii As MSForms.ReturnInteger)
```

```
    Select Case KeyAscii
        Case vbKey0 To vbKey9
        Case vbKeyBack, vbKeyClear, vbKeyDelete
        Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
        Case Else
            KeyAscii = 0
            Beep
    End Select
```

```
End Sub
```

```
Private Sub TextBox108_KeyPress (ByVal KeyAscii As MSForms.ReturnInteger)
```

```
    Select Case KeyAscii
        Case vbKey0 To vbKey9
        Case vbKeyBack, vbKeyClear, vbKeyDelete
        Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
        Case Else
            KeyAscii = 0
            Beep
    End Select

End Sub
```

```
Private Sub TextBox109_KeyPress (ByVal KeyAscii As MSForms.ReturnInteger)
```

```
    Select Case KeyAscii
        Case vbKey0 To vbKey9
        Case vbKeyBack, vbKeyClear, vbKeyDelete
        Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
        Case Else
            KeyAscii = 0
            Beep
    End Select

End Sub
```

```
Private Sub TextBox110_KeyPress (ByVal KeyAscii As MSForms.ReturnInteger)
```

```
    Select Case KeyAscii
        Case vbKey0 To vbKey9
        Case vbKeyBack, vbKeyClear, vbKeyDelete
        Case vbKeyLeft, vbKeyRight, vbKeyUp, vbKeyDown, vbKeyTab
        Case Else
            KeyAscii = 0
            Beep
    End Select

End Sub
```