



ANÁLISE DE ESTRATÉGIAS DE EMBARQUE NUM AIRBUS A320 RECORRENDO A MODELOS DE SIMULAÇÃO

HÉLIO GABRIEL DA ROCHA MOREIRA

julho de 2021

ANÁLISE DE ESTRATÉGIAS DE EMBARQUE NUM AIRBUS A320 RECORRENDO A MODELOS DE SIMULAÇÃO

Hélio Gabriel Rocha Moreira



Mestrado em Engenharia Electrotécnica e de Computadores

Área de Especialização de Automação e Sistemas

Departamento de Engenharia Eletrotécnica

Instituto Superior de Engenharia do Porto

2021

Este relatório satisfaz, parcialmente, os requisitos que constam da Ficha de Disciplina de Tese/Dissertação, do 2º ano, do Mestrado em Engenharia Eletrotécnica e de Computadores

Candidato: Hélio Gabriel Rocha Moreira, N° 1161131, 1161131@isep.ipp.pt

Orientação científica: Doutor Luís Carlos Ramos Nunes Pinto Ferreira, lpf@isep.ipp.pt

Coorientação científica: Doutor Paulo António da Silva Ávila, psa@isep.ipp.pt



Mestrado em Engenharia Eletrotécnica e de Computadores

Área de Especialização de Automação e Sistemas

Departamento de Engenharia Eletrotécnica

Instituto Superior de Engenharia do Porto

2 de julho de 2021

Agradecimentos

Os meus agradecimentos vão para todos aqueles que, de diferentes modos, apoiaram-me na realização desta dissertação. Gostaria, no entanto, de destacar aqueles que mais diretamente colaboraram comigo.

Em especial, ao Professor Doutor Luís Pinto Ferreira por ter aceitado ser meu orientador, pelo seu saber e incrível disponibilidade, pelo seu encorajamento, preocupação e ajuda que demonstrou na realização desta dissertação.

Ao Professor Doutor Paulo António da Silva Ávila por ter aceitado ser meu coorientador, pelo seu contributo e opinião assertiva.

A todas as pessoas (professores/colegas) que partilharam o seu conhecimento ao longo destes anos que de alguma forma me ajudaram a chegar até aqui, contribuindo de forma essencial para a minha formação, tanto a nível académico, como a nível humano.

Aos meus pais e irmã, pela paciência, apoio incondicional e a força que me deram para realizar esta dissertação.

E, por fim, aos meus amigos mais próximos por me terem acompanhado nos bons e nos maus momentos.

O meu mais sincero obrigado a todos.

Resumo

O tempo de embarque é um dos principais fatores que interferem na gestão e fluidez dos aeroportos, bem-estar de pessoas e a exigência atual do mundo empresarial. Portanto, para reduzir o tempo de embarque, é imperativo repensar os sistemas tradicionais que suportam o embarque de passageiros, tornando-os mais eficientes. Esta redução, para além de beneficiar as companhias aéreas, aeroportos e passageiros, beneficia também em termos económicos, operacionais e de satisfação do cliente.

Neste sentido, o trabalho apresentado nesta dissertação procura desenvolver um modelo de simulação, com recurso ao *software* Arena, que permite analisar o impacto de diferentes estratégias de embarque, verifique a possibilidade de redução dos tempos de embarque e explore situações excecionais como se verifica no atual contexto de pandemia COVID-19. Nas diversas simulações, selecionou-se sete estratégias de embarque incluindo *random*, *back-to-front*, *outside-in*, *reverse pyramid*, *blocks*, *Steffen* e *modified optimal*. Neste estudo são considerados vários cenários experimentais, analisando o seu impacto nas diferentes estratégias.

De uma forma geral, as simulações revelam que as estratégias *outside-in* e *reverse pyramid* são as mais eficazes, obtendo uma melhoria de até 20% no tempo de embarque. Além disso, a estratégia *back-to-front*, que é usada por várias companhias aéreas, é ineficiente. Estas companhias poderão beneficiar de uma melhoria no tempo de embarque até 40%, caso adotem uma das estratégias eficazes descritas nesta dissertação. As simulações realizadas em contexto de pandemia revelam que existe uma margem de progressão de 15% no tempo de embarque, caso seja utilizada a estratégia *reverse pyramid*, porque permite diminuir o número de interferências entre passageiros. Este é ainda um fator importante para reduzir riscos de infeção entre os passageiros.

Palavras-Chave

Estratégias de Embarque, Gestão de Aeroportos, Simulação, Pandemia.

Abstract

Boarding time is one of the major factors that interferes with the management and fluidity of airports, people welfare and the demands in the business world. For that, to efficiently establish passenger boarding, it is imperative to rethink the traditional systems and reduce the boarding time. This, besides benefiting the airlines, airports and passengers, benefit in turn economical, operational and customer satisfaction terms.

Taking this into consideration, the work presented in this dissertation seeks to development of a simulation model, using the Arena software, which allows analysis of the impact of different boarding strategies, verifies the different possibilities in reducing boarding times and explores exceptional situations, such as the current pandemic context COVID-19. In the various simulations, seven boarding strategies were selected including random, back-to-front, outside-in, reverse pyramid, blocks, Steffen, and modified optimal. In the present study several experimental scenarios are considered and their impact on different strategies analysed.

Overall, the simulations revealed that the outside-in and reverse pyramid strategies provided the best efficient, obtaining an improvement of up to 20% in boarding time. Furthermore, the back-to-front strategy, that is used by many airlines, is inefficient. These companies can benefit from an improvement in boarding time of up to 40% if they adopt one of the efficient strategies described in this thesis. Moreover, simulations carried out in a pandemic context revealed that there is a 15% progression margin in boarding time if the reverse pyramid strategy is used due to the reduction in the number of interferences between passengers. This is an important factor in reducing risks of infection among passengers.

Keywords

Boarding Time, Airport Management, Simulation, Pandemic.

Índice

AGRADECIMENTOS	I
RESUMO	III
ABSTRACT	V
ÍNDICE	VII
ÍNDICE DE FIGURAS	IX
ÍNDICE DE TABELAS	XIII
ACRÓNIMOS	XV
GLOSSÁRIO	XVII
1. INTRODUÇÃO	1
1.1. CONTEXTUALIZAÇÃO	1
1.2. OBJETIVOS	3
1.3. METODOLOGIA DE INVESTIGAÇÃO	4
1.4. CONTEÚDO E ORGANIZAÇÃO DA DISSERTAÇÃO	5
2. FUNDAMENTAÇÃO TEÓRICA E REVISÃO DE LITERATURA	7
2.1. CASOS DE ESTUDO	7
2.2. ANÁLISE DO PROCESSO, INTERFERÊNCIAS E ESTRATÉGIAS DE EMBARQUE	12
2.3. RESTRIÇÕES DEVIDO AO COVID-19 NO EMBARQUE.....	22
2.4. MODELOS PARA O PROCESSO DE EMBARQUE	23
3. ESTUDO DE SIMULAÇÃO	28
3.1. <i>INTERFACE</i> GRÁFICA	28
3.2. ANIMAÇÃO	35
3.3. MODELO LÓGICO.....	37
3.4. CÓDIGO VBA DESENVOLVIDO	60
4. ANÁLISE DE RESULTADOS	73
4.1. ANIMAÇÃO DO MODELO DE SIMULAÇÃO.....	73
4.2. ANÁLISE DE RESULTADOS	86
4.3. VALIDAÇÃO DO SISTEMA DESENVOLVIDO.....	98
5. CONCLUSÕES E PROPOSTAS DE TRABALHO FUTURO	100
5.1. CONCLUSÕES	100
5.2. CARACTER INOVADOR DO TRABALHO.....	102
5.3. PROPOSTAS DE TRABALHO FUTURO	102

REFERÊNCIAS BIBLIOGRÁFICAS	104
APÊNDICE A – CÓDIGO VBA	111

Índice de Figuras

Figura 1: Planeamento do <i>turnaround time</i> num gráfico de Gantt (Muzik, 2017).....	12
Figura 2: Elementos e distúrbios do <i>turnaround time</i> (Van Landeghem & Beuselinck, 2002)	13
Figura 3: Interferências no embarque (Jaehn & Neumann, 2015)	15
Figura 4: Tipos de interferências de assento(Cotfas et al., 2020)	16
Figura 5: Classificação das estratégias de embarque (Jaehn & Neumann, 2015)	17
Figura 6: Estratégias de embarque: (a) <i>random</i> ; (b) <i>back-to-front</i> ; (c) <i>outside-in</i> ; (d) <i>reverse pyramid</i> ; (e) <i>blocks</i> ; (f) <i>steffen</i> ; (g) <i>modified optimal</i> (Steffen & Hotchkiss, 2012)	20
Figura 7: Exemplo de configuração do avião numa situação de pandemia (Cotfas et al., 2020).....	23
Figura 8: Fluxograma para a Interface Gráfica Desenvolvida	29
Figura 9: Interface Gráfica - Página Inicial.....	30
Figura 10: Interface Gráfica - Parametrização, <i>tab Passengers Arrival</i>	31
Figura 11: Interface Gráfica - Parametrização, <i>tab Strategies</i>	32
Figura 12: Interface Gráfica - Parametrização, <i>tab Boarding Parameters</i>	33
Figura 13: Interface Gráfica - Parametrização, <i>tab Simulation Parameters</i>	34
Figura 14: Interface Gráfica – Resultados.....	35
Figura 15: Exemplo da Animação do Modelo Desenvolvido - Fonte imagem base do Airbus A320: (Seatplans.com, 2021)	36
Figura 16: Menu de Animação Informativo.....	37
Figura 17: Modelo de Simulação Desenvolvido	38
Figura 18: Conjunto de Blocos que criam as Entidades Passageiros	39
Figura 19: Parâmetros do bloco <i>Create</i>	40
Figura 20: Parâmetros do bloco <i>Access</i>	40
Figura 21: Parâmetros do bloco <i>Convey</i>	41
Figura 22: Módulo <i>Conveyor</i>	41
Figura 23: Módulo <i>Segment</i>	41
Figura 24: Conjunto de Blocos que atribuem lugares às entidades.....	42
Figura 25: Parâmetros do bloco <i>Decide, First_Pass</i>	42
Figura 26: Parâmetros do bloco <i>Assign, Start_Time</i>	43
Figura 27: Parâmetros do bloco <i>Assign, Pass_Count</i>	43
Figura 28: Incremento da variável <i>passengerscount</i>	44
Figura 29: Parâmetros do bloco <i>Assign, Assign_Seat</i>	45

Figura 30: Atribuição do <i>Line_Attribute</i>	45
Figura 31: Atribuição do <i>Seat_Attribute</i>	45
Figura 32: Parâmetros do bloco <i>Convey, Rd Convey to Aisle Station</i>	46
Figura 33: Conjunto de blocos que transporta os passageiros até ao lugar de destino	46
Figura 34: Parâmetros do bloco <i>Station, Set_Current_Aisle</i>	47
Figura 35: Parâmetros do bloco <i>Decide, ReachLine</i>	47
Figura 36: Parâmetros do bloco <i>Convey, Convey to Next Aisle</i>	48
Figura 37: Parâmetros do bloco <i>Seize, Seat_Reach</i>	48
Figura 38: Alteração do estado do recurso	49
Figura 39: Parâmetros do bloco <i>Decide, Luggage</i>	49
Figura 40: Parâmetros do bloco <i>Process, Put Away</i>	50
Figura 41: Conjunto de blocos que simulam as interferências de assento	51
Figura 42: Preenchimento da matriz lugares no bloco <i>Assign</i>	51
Figura 43: Parâmetros do bloco <i>Decide, FirstClass?</i>	52
Figura 44: Parâmetros do bloco <i>Decide, SeatInterference</i>	53
Figura 45: Exemplo de bloco <i>Decide, MiddleInterference</i>	53
Figura 46: Exemplo de bloco <i>Decide, WindowInterference</i>	54
Figura 47: Interferência do tipo 1	55
Figura 48: Interferência do tipo 3 e tipo 4.....	55
Figura 49: Interferência do tipo 2.....	55
Figura 50: Parâmetros do bloco <i>Assign, PassengerSeated</i>	56
Figura 51: Tarefa para alterar o estado do recurso	56
Figura 52: Conjunto de blocos que regista os tempos de embarque	57
Figura 53: Parâmetros do bloco <i>Exit</i>	57
Figura 54: Registo da cadência de saída.....	57
Figura 55: Registo do tempo individual de embarque.....	58
Figura 56: Parâmetros do bloco <i>Decide, Last_Pass_Seated</i>	58
Figura 57: Parâmetros do bloco <i>Assign, Record Final Time</i>	59
Figura 58: Cálculo do tempo de embarque.....	59
Figura 59: Parâmetros do bloco <i>Dispose</i>	59
Figura 60: Parâmetros para a animação do modelo.....	74
Figura 61: Cenários do embarque para os passageiros da primeira classe	75
Figura 62: Estratégia <i>back-to-front</i> - Fonte imagem base do Airbus A320: (Seatplans.com, 2021) 76	
Figura 63: Estratégia <i>random</i> - Fonte imagem base do Airbus A320: (Seatplans.com, 2021)	77
Figura 64: Estratégia <i>outside-in</i> - Fonte imagem base do Airbus A320: (Seatplans.com, 2021).....	78

Figura 65: Estratégia <i>reverse pyramid</i> - Fonte imagem base do Airbus A320: (Seatplans.com, 2021)	79
Figura 66: Estratégia <i>blocks</i> - - Fonte imagem base do Airbus A320: (Seatplans.com, 2021).....	80
Figura 67: Estratégia <i>Steffen</i> - Fonte imagem base do Airbus A320: (Seatplans.com, 2021).....	81
Figura 68: Estratégia <i>modified optimal</i> - Fonte imagem base do Airbus A320: (Seatplans.com, 2021)	82
Figura 69: Possíveis cenários de embarque para os passageiros de classe prioritária Fonte imagem base do Airbus A320: (Seatplans.com, 2021)	84
Figura 70: Possíveis cenários de embarque, modo de doenças infecciosas Fonte imagem base do Airbus A320: (Seatplans.com, 2021)	85
Figura 71: Tempo total de embarque	87
Figura 72: Tempo de embarque dependente do número de passageiros com bagagem de mão	89
Figura 73: Tempo de embarque dependente do número de passageiros prioritários	92
Figura 74: Tempo total de embarque, modo doenças infecciosas.....	93
Figura 75: Tempo de embarque dependente do número de passageiros com bagagem de mão, modo doenças infecciosas	95
Figura 76: Tempo de embarque dependente do número de passageiros prioritários, modo doenças infecciosas.....	97
Figura 77: Tempo individual de embarque, dependendo do número de famílias	98

Índice de Tabelas

Tabela 1 - Artigos relacionados com o projeto	8
Tabela 2: Estratégias de embarque utilizadas por companhias aéreas	18
Tabela 3: Vantagens e desvantagens das estratégias de embarque (Jaehn & Neumann, 2015)	21
Tabela 4: Divisão de classes de embarque	61
Tabela 5: Atribuição de assentos na estratégia <i>Random</i>	63
Tabela 6: Atribuição de assentos na estratégia <i>back-to-front</i>	63
Tabela 7: Divisão da Classe Económica na estratégia <i>back-to-front</i>	64
Tabela 8: Atribuição de assentos na estratégia <i>outside-in</i>	64
Tabela 9: Divisão da Classe Económica na estratégia <i>outsider-in</i>	65
Tabela 10: Atribuição de assentos na estratégia <i>reverse pyramid</i>	65
Tabela 11: Divisão da Classe Económica na estratégia <i>reverse pyramid</i>	66
Tabela 12: Atribuição de assentos na estratégia <i>blocks</i>	66
Tabela 13: Divisão da Classe Económica na estratégia <i>blocks</i>	67
Tabela 14: Atribuição dos 20 primeiros assentos na estratégia <i>Steffen</i>	67
Tabela 15: Atribuição de assentos na estratégia <i>modified optimal</i>	68
Tabela 16: Divisão da Classe Económica na estratégia <i>modified optimal</i>	68
Tabela 17: Divisão de classes de embarque com classe prioritária.....	69
Tabela 18: Estado dos Recursos.....	74
Tabela 19: Valores dos vários atrasos	86
Tabela 20: Tempo de embarque obtido de acordo com as diferentes estratégias.....	88
Tabela 21: Tempo individual de embarque obtido de acordo com as diferentes estratégias	88
Tabela 22: Comparação do tempo de embarque com e sem bagagem de mão	90
Tabela 23: Comparação do tempo individual de embarque com e sem bagagem de mão	91
Tabela 24: Tempo total de embarque obtido, modo doenças infecciosas	93
Tabela 25: Tempo individual de embarque obtido, modo doenças infecciosas.....	94
Tabela 26: Comparação do tempo de embarque com e sem bagagem de mão, modo doenças infecciosas.....	95
Tabela 27: Comparação do tempo individual de embarque com e sem bagagem de mão, modo doenças infecciosas	96
Tabela 28: Comparação de resultados existentes na literatura.....	98

Acrónimos

- USD – United States Dollar
- DOB – Dynamically Optimized Boarding
- DEVS – Discrete Event System Specification
- WMA – Window Middle Aisle
- IATA – International Air Transport Association
- VBA – Visual Basic for Application
- TNOW – Current Simulation Time

Glossário

Turnaround Time – Tempo necessário para descarregar um avião após a sua chegada à porta de embarque e prepará-lo novamente para a decolagem.

1. INTRODUÇÃO

Este estudo emerge da demanda por sistemas eficientes no tempo de embarque aeroportuário. O primeiro capítulo da dissertação é inteiramente dedicado à introdução do tema “Análise de Estratégias de embarque num Airbus A320 recorrendo a modelos de simulação”. Deste modo, começa-se por enquadrar o tema proposto e abordar a pertinência do seu estudo, depois são apresentados os objetivos da investigação, seguidamente aborda-se qual a metodologia seguida ao longo deste estudo e, por fim, dá-se indicação sobre qual a estrutura que o trabalho seguirá.

1.1. CONTEXTUALIZAÇÃO

O avião tornou-se um transporte integrante do nosso quotidiano (Bidanda et al., 2017), com um crescimento acentuado, nas últimas décadas (Kalic et al., 2013). Porém, esta procura pelas viagens aéreas provocou uma tremenda tensão no setor da aviação, procurando esta ir, simultaneamente, ao encontro da eficiência, da satisfação do cliente e do lucro (Nyquist & McFadden, 2008; Soolaki et al., 2012). Para evitar congestionamentos nos aeroportos e aumentar a eficiência operacional, a gestão dos aeroportos tem aumentado o número de portas de embarque e as companhias aéreas têm sido ‘obrigadas’ a procurar novas soluções de forma a reduzir o tempo de resposta dos aviões (*turnaround time*) (Van Landeghem & Beuselinck, 2002). Com o aumento da utilização do avião, minimiza-se o *turnaround time*,

que, por sua vez, aumenta a eficiência econômica do seu uso (Kalic et al., 2013). Isto acontece porque as companhias aéreas apenas começam a gerar receitas quando os aviões se encontram a voar, sendo considerado não produtivo o tempo que estes se encontram no solo (Jafer & Mi, 2017; Soolaki et al., 2012). O *turnaround time* é principalmente constituído pelo desembarque, limpeza e embarque, mas o seu ponto mais crítico encontra-se no embarque. Estima-se que a redução de 1 minuto no embarque economiza \$30 *United States Dollar* (USD) em cada *turnaround* (Nyquist & McFadden, 2008). Todos os dias, milhares de pessoas passam pela experiência de embarcar num avião. Todavia, este processo – colocar os passageiros num avião – nem sempre é organizado da melhor maneira (Jaehn & Neumann, 2015). As companhias aéreas lidam com algumas dificuldades relativamente ao tempo de embarque porque têm controlo limitado sobre os passageiros (Jafer & Mi, 2017). Para reduzir o tempo de embarque é necessário minimizar o número de interferências entre os passageiros. Concomitantemente, é necessário desenvolver uma política de embarque para controlar a ordem em que os passageiros entram no avião (Jafer & Mi, 2017; Soolaki et al., 2012). As companhias aéreas já possuem uma variedade de estratégias de embarque (Jafer & Mi, 2017; Soolaki et al., 2012), porém, quando executadas incorretamente, proporcionam atrasos, que podem acarretar custos elevados às companhias aéreas (Bidanda et al., 2017). No caso de estratégias bem aplicadas, as três partes principais – companhias aéreas, operadores aeroportuários e passageiros –, acabam por beneficiar da redução do tempo de embarque (Jaehn & Neumann, 2015). Para além disso, os passageiros apreciam soluções que minimizem o tempo de espera na porta de embarque (Zeineddine, 2017).

Assim sendo, o setor da aviação reconhece que, apesar do processo de embarque ter um custo significativo, ainda existe pouca literatura que aborde estratégias para melhorar o processo (Jafer & Mi, 2017; Nyquist & McFadden, 2008; Soolaki et al., 2012). Muitos investigadores propuseram diferentes estratégias de embarque com o objetivo de reduzir o seu tempo, mesmo que seja apenas de alguns minutos (2–8) (Jafer & Mi, 2017). Embora as companhias aéreas desejem melhorar o processo de embarque, estas são cautelosas a mudanças operacionais (Jafer & Mi, 2017; Soolaki et al., 2012). Portanto, qualquer proposta é sempre vista com cautela, mas este estudo pode ajudar a diminuir o tempo de embarque em vários cenários.

Apesar desta dissertação se focar na redução do tempo de embarque, não causando stress nem desconforto nos passageiros, devido à pandemia COVID-19, questões de saúde e ambientes de voo seguro preocupam as companhias aéreas. A maior razão pela qual o

coronavírus se tornou tão mortal, deve-se ao facto de pessoas contaminadas sem sintomas, poderem ter altas cargas virais e espalhar a doença (Regnier et al., 2020). Portanto, devido às regras de distanciamento social impostas pelas autoridades e instituições na área da gestão do transporte aéreo, o processo de embarque dos aviões mereceu alteração. Neste novo contexto, as companhias aéreas têm lutado para aumentar a confiança dos seus passageiros no transporte aéreo durante este surto, adotando, entre outras medidas, uma série de políticas relacionadas com o processo de embarque (Milne et al., 2020). Comparando as expectativas dos passageiros com as medidas adotadas pelas companhias aéreas, observa-se que existe um terreno comum no que diz respeito às regras de distanciamento físico (Vos, 2020). Este distanciamento tem sido uma das medidas mais discutidas durante o surto provocado pelo coronavírus, porque apesar de restringir a contaminação entre humanos, esta pode afetar a saúde mental e o bem-estar das pessoas, tornando-se numa medida controversa (Donovan, 2020). No entanto, para se adaptarem à nova realidade de distanciamento físico, as companhias aéreas deverão repensar em mecanismos de melhoria e adaptação de estratégias de embarque, para aumentarem a segurança dos seus passageiros durante a viagem em tempos de pandemia (Cotfas et al., 2020).

O trabalho aqui apresentado enquadra-se no âmbito da unidade curricular de Tese e Dissertação (TEDI) do Curso de Mestrado em Engenharia Eletrotécnica e de Computadores, no ramo de Automação e Sistemas.

1.2. OBJETIVOS

Este trabalho tem como objetivo a melhoria do processo de embarque num avião. Para tal, foi desenvolvida uma ferramenta baseada em modelos de simulação, utilizando o *software* de simulação ARENA, que permita ao utilizador testar e analisar o impacto de diferentes estratégias de embarque, tendo como suporte os seguintes pontos:

- Identificar e estudar as diferentes estratégias de embarque, motivos e causas que levam ao seu estudo;
- Desenvolvimento da ferramenta que permita testar as diferentes estratégias de embarque em diferentes condições;
- Identificação das melhores estratégias de embarque e fatores que contribuem para a sua eficácia;

- Criar um modo de funcionamento que permita testar as diferentes estratégias numa situação de pandemia;

1.3. METODOLOGIA DE INVESTIGAÇÃO

Para atingir os objetivos propostos neste trabalho, optou-se por seguir uma metodologia com uma abordagem de carácter quantitativo. A pesquisa quantitativa, segundo Bhandari, (2020), é uma classificação do método científico que utiliza diferentes técnicas estatísticas para quantificar opiniões e informações de um determinado estudo. Neste tipo de pesquisa, os dados são geralmente adquiridos através de tabelas e gráficos. Este processo é particularmente útil quando se pretende responder estatisticamente às questões “Como?” e “Porquê?” sobre um conjunto de eventos (Meyer, 2001).

O presente estudo aborda um modelo de simulação recorrendo ao *software* Arena, para identificar as melhores estratégias de embarque. Também se pretende perceber qual a reação destas estratégias perante um contexto de pandemia.

A metodologia de investigação delineada para realizar o cumprimento dos objetivos propostos é composta por quatro fases, sendo elas:

1. Revisão bibliográfica na área proposta – Estudo aprofundado sobre o problema do embarque de passageiros em aviões na literatura existente. A literatura é essencial para qualquer disciplina ou projeto de pesquisa. Através de estudos anteriores é possível mapear e avaliar a área em estudo, delinear objetivos e justificar a pesquisa em questão (Snyder, 2019).
2. Domínio da linguagem de simulação – Aquisição de conhecimentos sobre a linguagem de simulação necessária para suporte do modelo a construir.
3. Desenvolvimento de ferramenta de simulação e implementação das estratégias delineadas – Esta fase compreende o desenvolvimento do modelo que descreve o embarque de passageiros num avião, assim como a implementação das estratégias que procuram melhorar o tempo de embarque.
4. Análise dos resultados obtidos – Para finalizar, nesta fase é realizada uma análise dos resultados obtidos como forma de avaliar a possibilidade de implementação das estratégias desenvolvidas.

No desenvolvimento da ferramenta de simulação foi dada especial atenção à sua utilização, para que a mesma seja apreendida de forma intuitiva, facilitando o seu uso por qualquer utilizador, mesmo aqueles que não possuem grandes conhecimentos técnicos na área, permitindo selecionar facilmente diferentes estratégias e adotar diversos cenários para o embarque de passageiros.

1.4. CONTEÚDO E ORGANIZAÇÃO DA DISSERTAÇÃO

A presente dissertação é composta, essencialmente, por cinco capítulos, nomeadamente: Introdução, Fundamentação Teórica e Revisão de Literatura, Estudo de Simulação, Análise de Resultados e, por fim, Conclusões e Propostas de Trabalho Futuro. Os parágrafos seguintes descrevem respetivamente o conteúdo de cada um destes capítulos.

O Capítulo 1 procura, de uma forma introdutória, dar a conhecer a base para o desenvolvimento do trabalho, focando-se nos objetivos a atingir, quais as opções metodológicas selecionadas e a estrutura que a dissertação irá seguir.

No Capítulo 2, intitulado *Fundamentação Teórica e Revisão de Literatura*, é apresentada uma revisão da literatura, onde são abordados os conceitos considerados mais importantes para sustentar o trabalho desenvolvido nos capítulos seguintes.

O Capítulo 3, denominado *Estudo de Simulação*, descreve o estudo de simulação realizado, envolvendo a construção e desenvolvimento dos modelos de simulação com base no *software* ARENA.

No Capítulo 4, designado *Análise de Resultados*, são apresentados e discutidos os resultados obtidos.

O Capítulo 5, cujo título é *Conclusões e Propostas de Trabalho Futuro*, sintetiza o trabalho desenvolvido, analisando todo o trabalho realizado nos capítulos anteriores. São ainda identificadas as principais dificuldades encontradas e apresentadas as limitações do modelo e propostas para futuros trabalhos.

No final são apresentadas as respetivas referências utilizadas e apêndices.

2. FUNDAMENTAÇÃO TEÓRICA E REVISÃO DE LITERATURA

Para fundamentar a elaboração desta dissertação, realiza-se neste capítulo uma revisão bibliográfica das estratégias aplicadas no embarque de aviões. Os conceitos teóricos necessários ao desenvolvimento do projeto são também abordados e analisados, assim como o processo de embarque, as interferências causadas e as diversas estratégias de embarque existentes. O capítulo tem por base a consulta de diversos artigos científicos relacionados com o tema em questão.

2.1. CASOS DE ESTUDO

Na literatura, é possível encontrar diversos trabalhos sobre o embarque aeroportuário, seus diferentes métodos e estratégias para tornar mais seguro e rápido o embarque dos passageiros nos aviões. A finalidade desta tabela é proporcionar ao leitor um resumo do trabalho previamente desenvolvidos sobre esta temática.

Para a escolha dos artigos definiu-se alguns critérios, nomeadamente:

- Similaridade com o tema deste projeto;

- Trabalhos recentes;

Na Tabela 1, apresentada a seguir, compreende os principais trabalhos científicos no âmbito do embarque de aviões e que são relevantes para este estudo.

Tabela 1 - Artigos relacionados com o projeto

Referências	Descrição	Método
(Marelli, Mattocks, & Merry, 1998)	O primeiro documento que examina detalhadamente o problema do embarque num avião. Deste estudo conduzido pela Boeing nasce o conceito <i>turnaround time</i> e donde é discutido formas de reduzir o mesmo. Para tal, os autores desenvolveram uma simulação por computador, aplicada ao embarque de um Boeing 757 com 240 assentos. Os resultados obtidos mostram que o modelo <i>outside-in</i> pode reduzir o tempo de embarque em 46% e o uso de 2 portas pode reduzir o tempo até 20%.	Simulação
(Van Landeghem & Beuselinck, 2002)	Este trabalho teve como objetivo a redução do tempo de embarque de passageiros nos aviões. Através do software de simulação Arena foram simuladas várias estratégias de embarque num avião com 132 lugares e 23 filas, sendo os resultados obtidos expressos em Excel. Olhando para os mesmos, é possível concluir que as melhores estratégias são aquelas onde existe separação consecutiva dos passageiros, pelo menos numa fila, a fim de reduzirem a interferência. Os autores observaram que a estratégia mais rápida foi a <i>By_halfrow_alt_2</i> , que apresentou um tempo de embarque de 15,8 minutos, enquanto estratégias que separam os passageiros por blocos demoraram pelo menos 27,7 minutos.	Simulação
(Nyquist & McFadden, 2008)	Este trabalho foi desenvolvido com o objetivo de determinar qual a forma mais económica de embarcar passageiros nos aviões, tendo em conta a qualidade e a satisfação do cliente. Através da aquisição de dados e modelos matemáticos, concluiu-se que a melhor estratégia para reduzir o tempo médio de embarque é usar um método não tradicional, como a estratégia <i>reverse pyramid</i> , que reduz o tempo de embarque em 30%, e poupa, desta maneira, 35% no custo anual da companhia aérea comparativamente ao método tradicional. Os resultados também mostram que a limitação da bagagem	Análítico

	de mão a uma peça reduz o tempo de embarque em 15% e a sua eliminação 40%.	
(Soolaki et al., 2012)	O objetivo deste trabalho foi incrementar o tempo que um determinado avião se encontra em voo, reduzindo o seu <i>turnaround time</i> . Ao analisar os passos deste ciclo foi decidido que a melhor solução estaria na diminuição do tempo de embarque. Através do <i>software Matlab</i> , usando um algoritmo genético, foi desenvolvida um modelo de programação linear mista, aplicado a um Airbus-A320. Para além dos resultados mostrarem um menor número de interferências, com o método <i>reverse pyramid</i> , apresentam também uma diminuição no tempo de processamento em comparação com o modelo matemático dos <i>softwares LINGO e CPLEX</i> .	Analítico
(Tang, Wu, Huang, & Caccetta, 2012)	A análise dos autores utiliza o tráfego pedestre como teoria para modelar os movimentos dos passageiros durante o processo de embarque. Para além disso, é apresentada uma estratégia de embarque que considera as propriedades individuais dos passageiros. Esta e outras estratégias são analisadas através do <i>Matlab</i> . Os resultados indicam que as estratégias onde o embarque é realizado por assento, levam ao cruzamento entre passageiros, gerando interferências. Também se concluiu que a estratégia que considera as propriedades individuais dos passageiros, elimina a maioria das interferências.	Analítico
(Steffen & Hotchkiss, 2012)	Este trabalho foi desenvolvido para testar o método de <i>Steffen</i> com o intuito de reduzir o tempo de embarque dos passageiros nos aviões. Para tal, foi utilizado um método experimental num Boeing 757 onde são utilizadas 12 filas com 6 assentos cada. Cinco métodos foram testados com 72 passageiros de várias idades. Os resultados mostraram uma melhoria de 31,48% com a metodologia <i>Steffen</i> comparativamente ao <i>random</i> , e o método <i>outside-in</i> foi 10% mais rápido que o método <i>random</i> .	Experimental
(Kalic et al., 2013)	Este estudo foi feito para detetar o problema no embarque na linha aérea. Para tal foram analisadas várias estratégias de <i>boarding</i> , recorrendo a modelos de simulação com a utilização do software GPSSH, a fim de detetar a estratégia mais eficiente. Os resultados mostraram uma melhoria de 8,62% com a metodologia <i>outside-in</i> comparativamente à metodologia aleatória.	Simulação

(Qiang, Jia, Xie, & Gao, 2014)	Este trabalho tem como principal objetivo o aperfeiçoamento do tempo de embarque através da aplicação de uma estratégia, onde os passageiros com mais bagagem embarcam primeiro. Para tal foi utilizado um modelo de simulação baseado no autômato celular, onde cada célula representa um assento num avião, formado este por 25 filas com 6 assentos cada. Os resultados mostram que o modelo aprimorado apresenta uma eficácia e robustez excelente, acelerando o processo de embarque em 24% em relação ao embarque aleatório.	Simulação
(Jaehn & Neumann, 2015)	Este estudo visou a redução do <i>turnaround time</i> de um avião através da utilização de métodos de embarque eficientes. Este artigo examina, em primeiro lugar, as diversas estratégias de embarque e, de seguida, apresenta uma revisão da literatura, descrevendo os 12 artigos mais relevantes até à data. Por fim, faz um resumo do estado de arte descrevendo os resultados principais. Estes referem que a estratégia <i>back-to-front</i> requer mais tempo do que outras estratégias mais fáceis de implementar, como o embarque aleatório.	Análise
(Zeineddine, 2017)	Este estudo teve como objetivo a otimização do embarque de passageiros em aviões através do método <i>Dynamically Optimized Boarding</i> (DOB). A simulação de eventos discretos, aplicados num avião com 20 filas e 6 colunas, e a comparação deste modelo com outros considerados eficazes. Os resultados apresentam um desempenho próximo do método de <i>Steffen</i> . Em comparação ao modelo aleatório, o DOB reduziu o tempo de embarque em 34,9%.	Simulação
(Jafer & Mi, 2017)	Este estudo fez uma comparação entre estratégias de embarque num Airbus A320, onde foram utilizadas 26 filas e 150 assentos disponíveis, através da modelação e simulação. Para tal, foi usado o formalismo hierárquico <i>Discrete Event System Specification</i> e a sua extensão celular (Cell-DEVS), sendo implementadas 8 metodologias de embarque. Pelos resultados, é possível concluir que a melhor estratégia simulada foi a <i>Window Middle Aisle</i> (WMA), apresentando apenas 26,16 minutos, enquanto a pior estratégia foi a <i>back-to-front</i> , apresentando aproximadamente 37 minutos para o embarque.	Simulação

(Bidanda et al., 2017)	Este projeto foi desenvolvido para rever e otimizar os modelos de embarque num avião comercial. Para tal, foi comparada a abordagem analítica e a abordagem de simulação ponderando as suas limitações. Este artigo apresenta que todas as variáveis devem ser classificadas em 3 categorias: de aeroporto, de avião e externas. Tendo isto em consideração é então proposto um modelo de natureza dinâmica onde seja viável a sua modificação e precisão em termos de resultados. Os autores exemplificam com uma simulação do tipo <i>Web based</i> .	Análise
(Schultz, 2017)	Para reduzir o tempo de embarque, o trabalho em questão oferece uma abordagem inovadora, utilizando um assento deslizante (<i>Side-Slip Seat</i>) no lugar junto ao corredor, permitindo, assim, que os passageiros se cruzem durante o processo. Para extrair os benefícios desta nova abordagem, o autor analisa diferentes estratégias de embarque através de um algoritmo evolutivo. Os resultados mostram que a utilização do assento deslizante reduz o tempo de embarque entre 10-20%, mas que tal é dependente da estratégia utilizada. O algoritmo aplicado também mostra que as estratégias que alternam entre o lado direito e esquerdo do corredor são as que tiram maior proveito desta abordagem.	Analítico
(Ren & Xu, 2018)	Este trabalho analisou o embarque de aviões baseado nas interferências ocorridas. Foi conduzido um teste experimental numa cabine de embarque com quarenta e oito assentos, avaliando 6 estratégias de embarque. Também foi explorado o impacto que a música tem no tempo perceptível pelos passageiros. Os resultados confirmam que as estratégias <i>reverse pyramid</i> e <i>outside-in</i> exibem boa eficiência. A ausência de música no método aleatório reduz o tempo de embarque em 13%.	Experimental
(Kisiel, 2020)	Este trabalho tem como objetivo aumentar a eficiência do embarque nos aeroportos tendo em conta o embarque prioritário dos passageiros. O artigo analisa a sensibilidade das estratégias mais comuns face ao número de passageiros prioritários. Para tal, é desenvolvido um modelo de simulação num Airbus A320, com 30 filas e com 6 assentos em cada fila. Os resultados mostram que a utilização do embarque prioritário pode reduzir a eficácia do método de <i>Steffen</i> e <i>outsider-in</i> em 45% e 25% respetivamente. Por outro lado, o método <i>by row</i> e <i>half-</i>	Simulação

	row beneficiaram em 10% no tempo de embarque utilizando 110 passageiros prioritários.	
--	---	--

Como é possível verificar na Tabela 1, são analisados os principais artigos existentes referentes ao embarque de passageiros num avião. Os métodos analíticos e a simulação foram os mais utilizados para o estudo do problema.

2.2. ANÁLISE DO PROCESSO, INTERFERÊNCIAS E ESTRATÉGIAS DE EMBARQUE

Neste subcapítulo será analisado todo o processo de embarque, o percurso que os passageiros realizam, as causas dos atrasos durante o processo e, por fim, as estratégias de embarque existentes na literatura.

2.2.1. TURNAROUND TIME

Marelli et al., (1998) define o *turnaround time* ou *turn time* de um avião como o tempo necessário para descarregar um avião após a sua chegada à porta e prepará-lo para a partida novamente. Este tempo consiste em ações como: desembarque, descarregamento e carregamento da bagagem e/ou carga, abastecimento, limpeza da cabine, manutenção da cozinha e embarque (Soolaki et al., 2012; Kalic et al., 2013; Jafer and Mi, 2017). Estas ações e respetivos tempos podem ser visualizados na Figura 1.

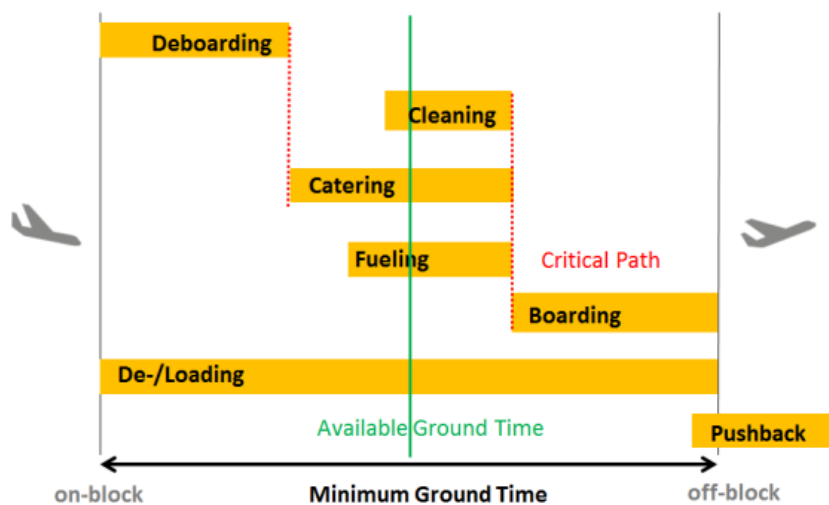


Figura 1: Planeamento do *turnaround time* num gráfico de Gantt (Muzik, 2017)

No estudo desenvolvido por Van Landeghem & Beuselinck, (2002), através da realização de entrevistas pessoais, concluiu que o *turn time* pode ir desde os 30 minutos até aos 60 minutos. Teoricamente, sair do avião leva aproximadamente 10 a 15 minutos, a limpeza leva cerca de 15 a 20 minutos e o embarque leva apenas 10 minutos. Mas na prática, esse tempo quase sempre é ultrapassado, muitas vezes chegando aos 30 minutos (Van Landeghem & Beuselinck, 2002). Anteriormente, a velocidade média de embarque era de cerca de 20 passageiros por minuto. No entanto, nos últimos 30 anos, essa taxa diminuiu para cerca de nove passageiros por minuto (Marelli et al., 1998). Este declínio de 55% na eficiência do processo de embarque pode ser atribuído ao aumento do uso da bagagem de mão, maior conveniência por parte dos passageiros, dos dados demográficos dos passageiros, das estratégias aplicadas pela companhia aérea e pela distância de voo (Nyquist & McFadden, 2008). A diminuição da taxa de embarque levou ao aumento do tempo de embarque. Estas razões conduziram ao estudo e à importância de rever o processo de embarque em aviões de passageiros (Soolaki et al., 2012). Na Figura 2 é possível observar a repartição do *turn time* e respetivos distúrbios.

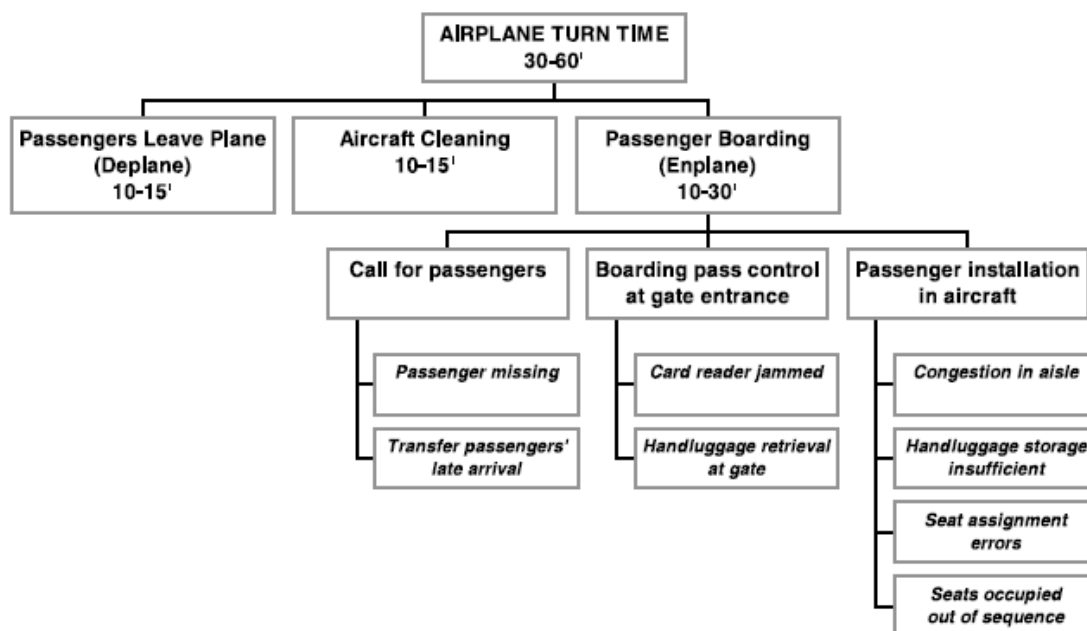


Figura 2: Elementos e distúrbios do *turnaround time* (Van Landeghem & Beuselinck, 2002)

2.2.2. PROCESSO DE EMBARQUE

Segundo Kalic et al., (2013) e Van Landeghem & Beuselinck, (2002) o processo de embarque nos aviões realiza-se, normalmente, da seguinte forma:

Etapa 1 - Os passageiros são chamados para o embarque pelo agente da porta de embarque da companhia aérea. Caso exista alguma estratégia de embarque, estes são chamados a embarcar na sequência proposta. Neste momento os passageiros começam a fazer fila na porta de embarque, enquanto o agente controla o cartão de embarque de cada passageiro e regista a sua entrada através de um leitor de bilhetes. Os fatores que podem atrasar este processo encontram-se em passageiros perdidos na área do aeroporto, que precisam de informação adicional, que chegam tarde à porta de embarque ou que apresentam excesso de bagagem de mão.

Etapa 2 - Os passageiros podem seguir em direção ao avião de várias formas. Através da ponte de acesso, a pé ou de autocarro. Caso a deslocação seja feita de autocarro, o atraso no embarque será agravado, pois além de várias viagens necessárias para trazer os passageiros da porta de embarque para o avião, os passageiros chegam ao avião em elevado número.

Etapa 3 - Ao entrar no avião, os passageiros formam novamente fila no corredor único do aparelho até chegarem ao lugar destinado. Neste processo, podem ocorrer várias interferências e atrasos provenientes do comportamento dos passageiros. Quando chegam ao seu lugar, os passageiros precisam de guardar a sua bagagem no compartimento adequado, bloqueando o corredor durante o processo. Após a conclusão de tal ação, a menos que não haja interferências no assento, o passageiro pode finalmente tomar o seu lugar. Caso contrário, o passageiro que bloqueia o caminho tem de se levantar para ceder passagem.

Deve-se salientar que as etapas 1 e 2 não precisam que o avião esteja pronto a embarcar e, portanto, não contribuem para o tempo de embarque. No entanto, o embarque normalmente só é anunciado quando os hospedeiros de bordo informam os passageiros que o avião se encontra pronto, o que faz com que a etapa 1 apenas comece posteriormente. Tal acontece para que os passageiros não se encontrem na fila de espera para o embarque durante muito tempo (Van Landeghem & Beuselinck, 2002).

Durante a etapa 2 também é possível a utilização de duas portas para a entrada de passageiros (a da frente e a de trás), o que geralmente acontece em voos de longa duração ou por companhias aéreas *lowcost*, reduzindo consideravelmente o tempo de embarque e por sua vez o *turn time* (Van Landeghem & Beuselinck, 2002). Mas, segundo Singh, (2019), este

processo não é frequentemente utilizado porque limita a comodidade dos passageiros e requer um planejamento extra por parte das companhias aéreas. Para além de ser preciso acesso ao terminal, que muitas vezes não é possibilitado. Os passageiros preferem a utilização da ponte de embarque porque, caso saiam do edifício e se desloquem a pé até ao avião ou através de autocarro, ficam sujeitos a qualquer condição climática.

2.2.3. INTERFERÊNCIAS NO EMBARQUE

A principal causa do atraso no embarque deve-se às interferências entre passageiros. Soolaki et al., (2012) declara que a interferência no embarque é definida como uma instância de um passageiro bloqueando o acesso de outro passageiro ao seu assento. Como mostrado na Figura 3, as interferências estão classificadas em dois tipos: interferências de assento e interferências de corredor.

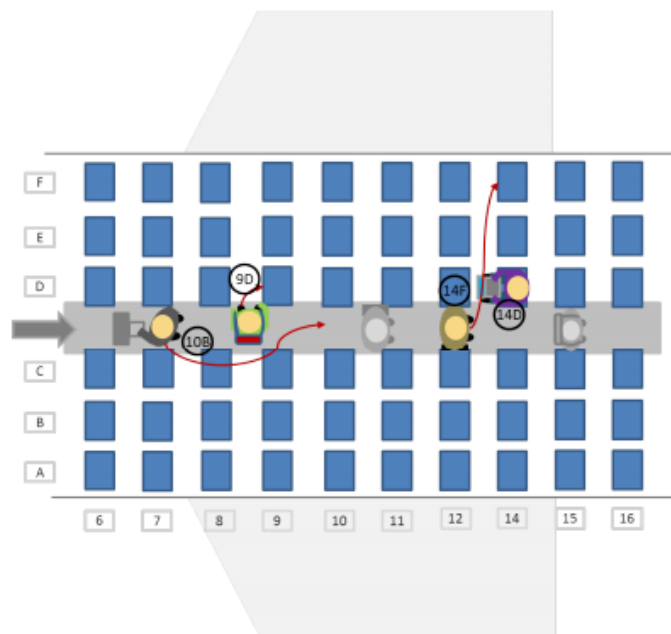


Figura 3: Interferências no embarque (Jaehn & Neumann, 2015)

As interferências no corredor ocorrem quando os passageiros desejam chegar aos seus assentos e são bloqueados no corredor por outros passageiros que estão parados à sua frente, normalmente a guardar a bagagem de mão nos armários superiores. Isso acontece, por exemplo, quando o passageiro do assento 10B é bloqueado pelo passageiro do assento 9D.

O passageiro do assento 10B não consegue passar no corredor pelo passageiro do assento 9D enquanto este não se sentar (Figura 3).

As interferências de assento acontecem, por exemplo, quando o passageiro da janela (passageiro de assento 14F) deseja sentar-se e precisa pedir ao passageiro que já se encontra sentado no assento do corredor (passageiro de assento 14D) ou no meio, para se levantar e permitir a sua passagem (Figura 3).

Segundo Delcea et al., (2018) e Cotfas et al., (2020) existem quatro tipos de interferências de assento que podem ser encontradas por um passageiro durante o seu trajeto. A Figura 4 apresenta os tipos de interferência de assento existentes.

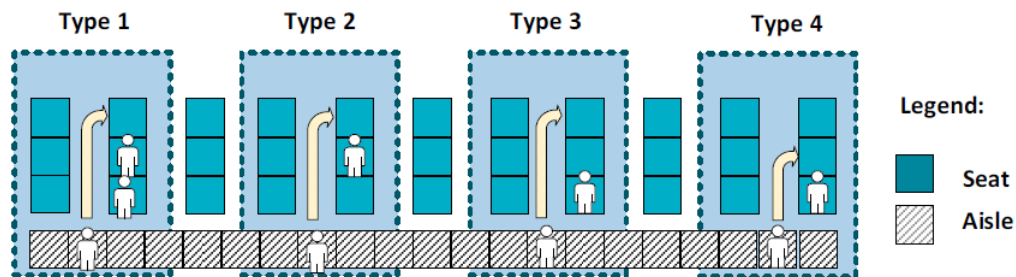


Figura 4: Tipos de interferências de assento(Cotfas et al., 2020)

A maior atraso das interferências é gerado pelo tipo 1, pois o passageiro que está no lugar perto da janela tem que esperar que os outros dois passageiros libertem o caminho. O segundo maior atraso é fornecido pela interferência de assento do tipo 2. Os tipos 3 e 4 geram um tempo de atraso idêntico sendo este o menor.

2.2.4. ESTRATÉGIAS DE EMBARQUE

Nesta seção, serão abordadas as diversas estratégias de embarque, conforme podem ser encontradas na prática e na literatura científica. Dois métodos são usados normalmente para o embarque de passageiros, tendo em consideração a atribuição de assento: o assento é atribuído ou não (método *open seating*).

No método *open seating*, os passageiros não possuem assentos atribuídos ao entrar no avião e, conseqüentemente, podem escolher qualquer lugar livre. Este método não será considerado para análise, pois depende exclusivamente do comportamento dos passageiros. Para além disso, a utilização deste método tem vindo a diminuir nos últimos anos por ser

considerado desconfortável por parte dos passageiros e por vezes stressante (Jaehn & Neumann, 2015).

Quando o assento é anteriormente atribuído, são distinguidas entre as inúmeras estratégias de embarque de passageiros três principais categorias: aleatório, por grupo e por assento. O primeiro método permite que os passageiros entrem de forma aleatória, sem ordem específica, permitindo que o embarque não necessite de nenhum anúncio ou sinal extra. Os últimos dois métodos mencionados, podem variar em diversas aplicações, dando origem a estratégias de embarque distintas (Figura 5). O método de embarque em grupo, divide, normalmente, os passageiros entre 2 a 6 grupos, que entram no avião numa ordem específica. Logo que o primeiro grupo embarcar, o segundo é chamado, e assim por diante. Dentro dos grupos, os passageiros embarcam de forma aleatória. O terceiro método, por assento, define uma sequência exata de entrada dos passageiros, ou seja, cada passageiro deve embarcar numa determinada posição, sendo esta planeada com antecedência, oferecendo a possibilidade de embarque na ordem ideal (Jaehn & Neumann, 2015).

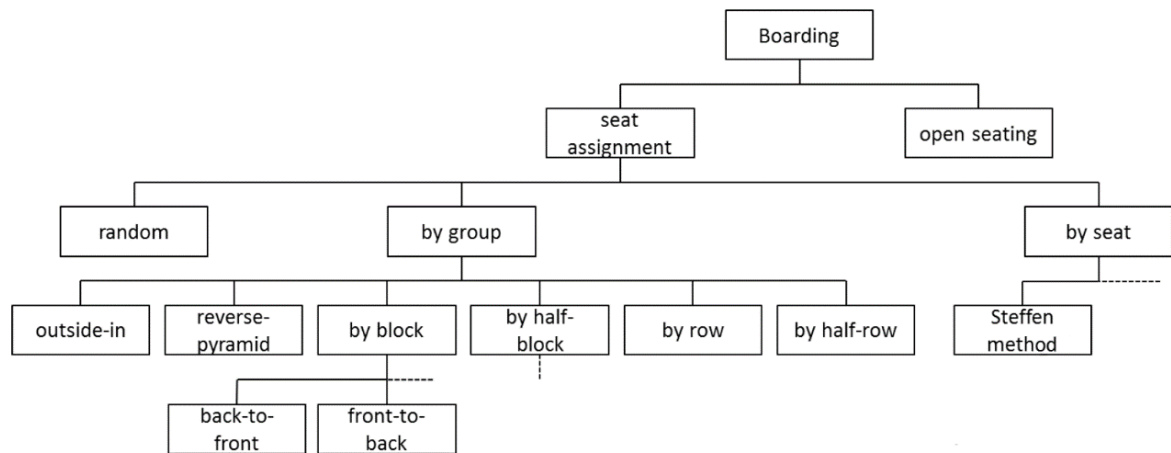


Figura 5: Classificação das estratégias de embarque (Jaehn & Neumann, 2015)

Segundo Kalic et al., (2013) as estratégias de embarque mais utilizadas pelas companhias aéreas são a *random* e a *back-to-front*. No entanto, existem outras companhias a utilizar diferentes métodos. Steffen & Hotchkiss, (2012) e Jafer & Mi, (2017), explicam as principais estratégias utilizadas por algumas companhias, entanto que Kalic et al., (2013) e Bidanda et al., (2017) referem as companhias aérea que utilizam as diversas estratégias. Estes parâmetros podem ser notados na Tabela 2.

Tabela 2: Estratégias de embarque utilizadas por companhias aéreas

Estratégia	Descrição	Companhia aérea
Random	Este método de embarque, não precisa de nenhum método específico. Todos os passageiros pertencem a um único grupo de embarque que entram no avião de forma aleatória. Esta abordagem é frequentemente usada como base para comparação com os outros métodos de embarque (Steffen & Hotchkiss, 2012).	Southwest Airlines; US Airways
Back-to-Front	Neste método de embarque os primeiros passageiros a embarcar são os que se encontram nas últimas filas do avião. O embarque continua até chegar às primeiras filas. As filas do avião são divididas por zonas. As zonas podem ter qualquer número, desde dois ao número total de filas reais. É uma estratégia fácil de implementar, porém, é facilmente sujeita à ineficiência, pois ocorrem congestionamentos nas filas de embarque (Steffen & Hotchkiss, 2012).	Air Canada; Alaska; British Airways; Frontier; JetBlue; Atlantic
Outside-In	Também conhecido como WMA ou Wilma (Window Middle Aisle) este método embarca primeiro os passageiros que têm lugares à janela. Seguem-se os passageiros com os assentos do meio e, por fim, os do corredor. Este método revelou até agora um tempo de embarque muito eficiente, eliminando completamente as interferências de assento (Jafer & Mi, 2017).	United
Reverse Pyramid	Este método consiste no embarque de passageiros desde a traseira externa até à parte da frente interna da cabine. Este método é, na verdade, uma combinação entre as estratégias <i>back-to-front</i> e <i>outside-in</i> , existindo um embarque simultâneo de um avião de trás para a frente e de fora para dentro. Neste método embarcam primeiros os passageiros que se encontram à janela e no lugar do meio na parte traseira do avião. Os passageiros que possuem lugar no	US Airways (America West)

	<p>corredor na parte da frente do avião são os últimos a embarcar. Essa estratégia é um método eficiente na AmericanWest Airlines (Jafer & Mi, 2017).</p>	
<p>Rotating zone / Blocks</p>	<p>Neste método, as filas do avião são divididas em zonas (cada zona é agrupada com um número de filas) sendo atribuído ao passageiro uma determinada zona. Este método embarca primeiro os passageiros que se encontram nas últimas filas (zona 1). De seguida embarcam os passageiros que estão nas primeiras filas (zona 2). Depois, a ordem continua novamente com a zona mais distante (últimas filas não ocupadas), as filas da frente, e assim por diante. A vantagem desse método é que os passageiros que embarcam na parte de trás e na frente não interferem entre si (Jafer & Mi, 2017).</p>	<p>AirTran</p>
<p>Steffen Method</p>	<p>O método de Steffen alinha os passageiros numa ordem específica. O embarque é realizado de trás para a frente e das janelas para o corredor. Os passageiros adjacentes na fila estão sentados duas filas afastadas uns dos outros em lugares correspondentes (por exemplo, 12F, 10F, 8F, 6F, 4F, 2F), preenchendo primeiro, em cada lado da cabine, as linhas pares e ímpares, os assentos da janela, de seguida os do meio e, por fim os assentos do corredor (Steffen & Hotchkiss, 2012).</p>	<p>-</p>
<p>Modified Optimal</p>	<p>Este método consiste no embarque dos passageiros em filas alternadas dando espaço suficiente para estes carregarem as suas bagagens. Os passageiros são separados em 4 grupos de embarque. O primeiro grupo é formado por todos os passageiros nas filas pares apenas de um lado do avião. O segundo grupo é formado por todos os passageiros do outro lado do avião. O terceiro e quarto grupo são os passageiros das filas ímpares em cada lado do avião (Jafer & Mi, 2017).</p>	<p>South West Airline</p>

Também existem outros métodos de embarque como: *Half Block*; *Half Row*; *By letter* (A B C D E F); *By seat* (Van Landeghem & Beuselinck, 2002). No entanto, estes não serão abordados, na medida que são alterações dos métodos já referidos.

Em resumo do que foi dito anteriormente é possível esquematizar as diferentes estratégias através da

Figura 6.

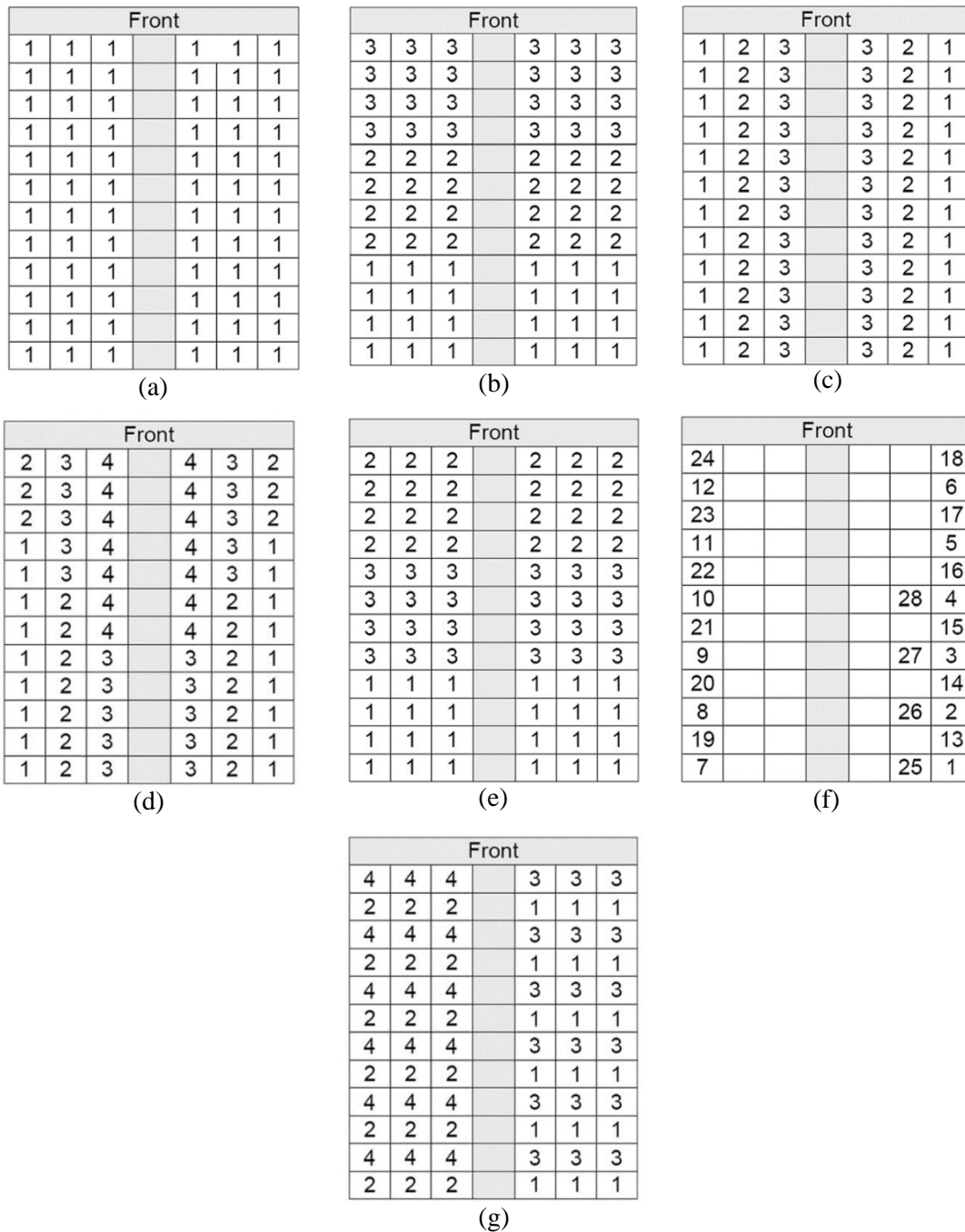


Figura 6: Estratégias de embarque: (a) *random*; (b) *back-to-front*; (c) *outside-in*; (d) *reverse pyramid*; (e) *blocks*; (f) *steffen*; (g) *modified optimal* (Steffen & Hotchkiss, 2012)

Cada método de embarque têm as suas vantagens e desvantagens, sendo estas apresentadas na Tabela 3.

Tabela 3: Vantagens e desvantagens das estratégias de embarque (Jaehn & Neumann, 2015)

Estratégia	Vantagens	Desvantagens
Random	Não exige nenhum esforço na chamada dos passageiros; estes quando viajam em grupo podem embarcar juntos; passageiros atrasados não influenciam a eficiência.	Método lento.
Back to Front	Fácil de entender; os passageiros que viajam em grupo podem embarcar juntos.	Congestionamentos em pequenas áreas do avião; método bastante lento.
Outside-In	Método rápido pois não existem interferências de assento.	Separação dos passageiros que viajam em grupo.
Reverse Pyramid	Método rápido, onde não existem interferências de assento.	Método complexo; Separação dos passageiros que viajam em grupo.
Rotating zone / Blocks	Passageiros que viajam em grupo podem embarcar juntos;	Congestionamentos concentrados numa pequena área do avião; quantos mais blocos, mais lento.
Steffen	Método muito rápido porque não existem interferências de assento e apenas algumas de corredor.	Separação dos passageiros que viajam em grupo; Método difícil de aplicar.
Modified Optimal	Passageiros que viajam em grupo podem embarcar juntos; Redução de interferências de corredor.	Método complexo;

Estas estratégias de embarque devem ser facilmente compreendidas pelos passageiros e, embora a redução do tempo de embarque seja o principal objetivo, os custos associados, a facilidade de aplicação e a viabilidade também devem ser considerados.

Os tempos de embarque indicados na revisão da literatura são puramente teóricos, ou seja, operam sobre premissa que a respetiva estratégia funciona como planeado. Caso os

passageiros não embarquem na ordem devida, cheguem atrasados, não entendam à estratégia definida ou recusarem-se a embarcar da maneira ordenada, irão dificultar o desempenho das melhores estratégias (Jaehn & Neumann, 2015).

2.3. RESTRIÇÕES DEVIDO AO COVID-19 NO EMBARQUE

A atual pandemia causada pelo COVID-19 teve um impacto enorme na mobilidade, com sérias implicações nas atividades económicas e sociais. As companhias aéreas ficaram, assim, duplamente afetadas. Para além de terem sido impostas limitações aos voos internacionais em diferentes países, assistiu-se ao receio dos passageiros em realizar viagens pelos motivos atrás descritos (Salari et al., 2020). Para manter as viagens aéreas funcionais e para as companhias aéreas restaurarem a confiança dos seus passageiros, foi necessário implementar novas medidas para evitar a contaminação e a propagação do vírus durante o processo de embarque, voo e desembarque (Elcheroth & Drury, 2020).

O grupo de aconselhamento médico da *International Air Transport Association* (IATA) recomenda por parte das companhias aéreas a utilização dos seguintes fatores (IATA, 2020):

- Gestão adequada;
- Distância mínima entre passageiros de 1 a 2 metros;
- Limitação do uso da bagagem de mão;
- Redução do número de interferências durante o processo de embarque.

Assim sendo, o distanciamento físico pode ser alcançado dentro da cabine, entre os passageiros que caminham pelo corredor até e aos seus assentos (IATA, 2020).

Para muitas companhias aéreas, o distanciamento físico reduziu a capacidade máxima do avião num terço, aplicando a ‘política do espaço vazio no meio’ – deixar um assento vazio entre a janela e o corredor –, ou seja, os assentos do meio são mantidos desocupados (Figura 7). Um estudo conduzido por Barnett & Fleming, (2020) indica que quando o assento do meio está vago, o risco de contaminação por Covid-19 é reduzido para metade. Para além disso, o distanciamento social entre os passageiros que se encontram no corredor do avião, atrasa o processo de embarque, potenciando a preocupação com as interferências dos

passageiros durante este processo. Na mesma lógica existe um risco associado à passagem dos passageiro junto daqueles que já se encontram sentados (Delcea et al., 2020).

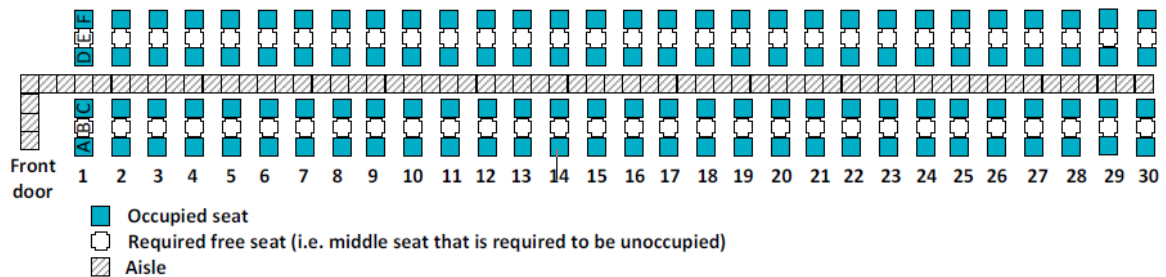


Figura 7: Exemplo de configuração do avião numa situação de pandemia (Cotfas et al., 2020)

Como foi analisado na seção 2.2.3. é possível que ocorram interferências do tipo 3, ou seja, o passageiro com o assento junto à janela terá que pedir ao passageiro com o assento junto ao corredor para poder chegar ao seu lugar. Durante este processo, se um dos passageiros estiver contaminado, o outro poderá ser infetado. Esta situação vem, por sua vez, incrementar o recurso a estratégias de embarque, que poderão ser eficazes em relação ao tempo de embarque, mas que poderão falhar no que concerne a assegurar um ambiente seguro aos passageiros (Milne et al., 2021). Ou seja, as diversas estratégias de embarque deverão incorporar normas de distanciamento físico que permitam proporcionar segurança aos passageiros.

Esta pesquisa, pretende criar um modo de funcionamento no modelo de simulação, que deixa o assento do meio desocupado na aplicação de qualquer estratégia de embarque. Contudo, pretende-se apresentar um outro quadro de funcionamento – modo família – que permita a ocupação do assento do meio, caso exista a presença de uma família com um agregado de pelo menos 3 elementos. Esta funcionalidade do modelo, tem vantagens sociais para a família que poderá viajar junta, e económicas para a companhia aérea que vê o assento contabilizado. Os resultados desses modelos poderão ser úteis para projetar estratégias de intervenção eficazes e seguras (Tariq, Roosa, & Chowell, 2020).

2.4. MODELOS PARA O PROCESSO DE EMBARQUE

De maneira a entender e otimizar um sistema, é necessário construir um modelo. Os modelos de um sistema são uma representação real usando medidas qualitativas e quantitativas, juntamente com lógica associada durante o processo. Como pode ser visto na Tabela 1

existem diferentes abordagens que podem ser utilizadas para a construção de um modelo, mas em termos gerais, estes são divididos em 2 tipos: modelos analíticos e modelos de simulação. Bidanda et al., (2017) explica as principais diferenças entre os dois modelos:

Os modelos analíticos podem ser construídos com base em definições matemáticas e teoremas, usando suposições e derivações específicas para fornecer uma solução para um determinado problema.

Os modelos de simulação são protótipos digitais de sistemas do mundo real. Utilizando como base variáveis aleatórias no contexto de um sistema real, as entradas do modelo podem ser geradas de acordo com as distribuições de entrada apropriadas. A ideia básica da simulação é usar o poder do processamento dos computadores para substituir as derivações matemáticas puras.

Através de processos analíticos, pode ser possível obter uma solução exata se o modelo a construir for suficientemente simples. Contudo, a formulação matemática é maioritariamente impossível, porque a grande parte dos sistemas existentes, representativos do mundo real, são de elevada complexidade. Para permitir modelar o comportamento de sistemas de qualquer grau de complexidade, o sistema deverá ser estudado com recurso à simulação (Ferreira, 2003).

2.4.1. CONCEITO DE SIMULAÇÃO, VANTAGENS E DESVANTAGENS

Em muitos ramos da engenharia, as experiências realizadas com o uso do computador em sistemas virtuais são uma ferramenta crítica para o estudo e análise de processos complexos (Ferreira et al., 2012). Desta forma, os protótipos são substituídos por simulações, proporcionando ganhos de produtividade complexos (Ryzhov et al., 2020). A simulação constitui um instrumento, que proporciona ao usuário a possibilidade de desenvolver um modelo baseado num sistema real, prevendo ações provenientes de dados previamente coletados, analisando áreas com potencial de melhoria e avaliando o impacto de diferentes tipos de estratégias (Ferreira et al., 2012; Ferreira et al., 2013). O conceito de simulação está relacionado com uma vasta gama de métodos e aplicações, que imitam o comportamento de sistemas reais, geralmente por meio de *softwares* apropriados. Na prática, a simulação por computador é uma ferramenta particularmente útil na análise de sistemas que incorporam a aleatoriedade nos seus parâmetros ou quando não existem soluções analíticas exatas (Ferreira et al., 2011). Com o uso desta técnica, é possível comprovar o desempenho real de diversos modelos, simplesmente variando um parâmetro e sem gastos associados (Ares et

al., 2012). Desta forma, a simulação apoia a tomada de decisão dentro das organizações, desenvolvendo um modelo que representa os comportamentos de um sistema real, permitindo, desta maneira, uma melhor percepção da sua estrutura que possibilita maximizar a produtividade num cenário de vida real (Sá et al., 2020).

Assim como qualquer ferramenta existente, a simulação também tem as suas vantagens e desvantagens. Segundo Chapuis et al., (2018), Ferreira et al., (2012) e Ferreira et al., (2003), as vantagens com maior particularidade são as seguintes:

- Permitir estudar um sistema com um longo período de tempo num tempo comprimido ou estudar o funcionamento detalhado de um sistema num período de tempo mais alargado;
- Testar o desempenho de um sistema existente sob certas condições de operação previamente projetadas;
- Explorar novas oportunidades, procedimentos ou métodos sem interromper o funcionamento do sistema atual;
- Identificar problemas e pontos de estrangulamento, com o intuito de conseguir otimizar o seu funcionamento;
- Permitir avaliar um sistema e identificar as variáveis que mais influenciam o seu desempenho.

Entre as desvantagens no uso da simulação, podem ser destacadas (Doddavaram & Corlu, 2020; Ferreira et al., 2012; Ren & Xu, 2018):

- Os *softwares* de simulação são dispendiosos;
- A construção dos modelos de simulação são dispendiosos e levam uma quantidade considerável de tempo para o seu desenvolvimento;
- O desenvolvimento destes modelos exige treino especializado, pois os seus recursos são difíceis de interpretar;
- Não fornece soluções ideais para problemas de casos de estudo, no entanto, permite avaliar o comportamento do sistema de acordo com diferentes cenários.
- Existem desvios entre a simulação de computador e o processo real. Por exemplo, nas simulações os passageiros não exibem comportamentos individuais e são considerados homogêneos, sendo impossível obter a sua percepção nas simulações.

2.4.2. FASES DE UM PROJETO DE SIMULAÇÃO

Para que o estudo da simulação seja bem-sucedido, é necessário a aplicação de diversas etapas, independentemente do tipo de problema ou objetivo do estudo. Segundo Barton, (2020), Silva et al., (2021) e Sturrock, (2020), as 9 etapas básicas do processo de simulação são as seguintes:

1. Definição do problema – A primeira etapa do processo de simulação consiste em definir os objetivos que podem ser alcançados com o estudo proposto e, posteriormente, descrever o problema que se pretende resolver.
2. Planeamento do Projeto – Divisão das várias tarefas necessárias para a realização do projeto. Para além de monitorizar o progresso, esta divisão permite criar um cronograma para determinar se existe tempo e recursos disponíveis para a sua conclusão.
3. Definição do Sistema – Identificação dos vários componentes do sistema que precisam de ser modelados e análise ao seu desempenho. Como, frequentemente, o sistema é complexo, requer um simulador experiente que possa encontrar o nível apropriado de detalhe e flexibilidade.
4. Formulação do Modelo – Compreensão do comportamento do sistema real para determinar os requisitos básicos do modelo a desenvolver. Estes requisitos envolvem as variáveis e recursos associados ao sistema real.
5. Recolha e análise de dados – Recolha de novos dados, compreensão e ajuste dos mesmo às distribuições teóricas.
6. Programação do Modelo – O modelo é traduzido em linguagem de programação, que se pauta na redução de tempo de leitura, processamento e saída de dados. Ou seja, criar uma representação adequada do sistema para apoiar os objetivos previamente estabelecidos.
7. Verificação e validação – A verificação é o processo que garante que o modelo se comporte conforme o esperado. Já a validação garante que não existe diferença significativa entre o modelo real e o modelo de simulação. A verificação e validação perfeitas geralmente são impossíveis porque o único modelo perfeito é o sistema real. Mas existem algumas maneiras de demonstrar que um determinado modelo é válido suficientemente para os propósitos do projeto. A verificação pode ser realizada por *debug* ou por animação. Já a validação pode ser alcançada por meio da análise estatística.

8. Execução de testes e análise – Realizar as simulações e comparar estatisticamente o desempenho das diferentes estratégias delineadas.
9. Documentação e análise dos relatórios – Documentação dos dados obtidos para a aplicação pretendida e devida implementação do modelo no sistema real.

2.4.3. SOFTWARE DE SIMULAÇÃO UTILIZADO ARENA

O *software* de simulação ARENA é um produto desenvolvido pela empresa Rockwell Automation para controlo e melhoria de soluções na área de automação industrial (Rockwell Automation, 2021). Este *Software* define-se num ambiente gráfico integrado de simulação, que contém todos os recursos para modelação de processos, desenho e animação, análise estatística e análise de resultados, possibilitando, desta maneira, a construção de modelos complexos, aliada a uma forte componente visual (Ferreira et al., 2005). A linguagem de programação atribuída ao ARENA é o SIMAN, uma ferramenta de programação flexível, dirigida aos objetos, combinando a construção de modelos de simulação com a integração de diferentes linguagens de uso geral como: *Visual Basic*, *C*, *C++* (Ferreira et al., 2011; Kelton et al., 1998). O SIMAN permite a criação de blocos de modelação com os seus componentes, proporcionando ao utilizador um maior controlo da lógica detalhada do sistema (Rockwell Automation, 2010).

3. ESTUDO DE SIMULAÇÃO

Neste capítulo vão ser abordados os temas referentes à elaboração dos diferentes modelos de simulação. Numa primeira fase, são descritas as distintas possibilidades da ferramenta construída, do ponto de vista do utilizador, através da interface gráfica e da animação. De seguida são apresentados os modelos utilizados e as políticas de controlo adotadas. Por último, será exposta a lógica utilizada no desenvolvimento do código.

3.1. INTERFACE GRÁFICA

O sistema desenvolvido é apresentado como uma interface gráfica, desenvolvida com recurso ao *Visual Basic for Application* (VBA) do ARENA, que permite ao utilizador configurar e colocar os parâmetros variáveis ou dados necessários para a simulação do modelo de uma forma simples e intuitiva. Melhor dizendo, não é necessário ter conhecimento sobre a modelação dos sistemas, para conseguir usar a interface gráfica. Para além desta ferramenta permitir configurar as variáveis de entrada e seleccionar a estratégia a implementar, apresenta, também, ao utilizador, outras informações que lhe poderão ser pertinentes. A programação utilizada para a construção da interface gráfica pode ser visualizada no Apêndice A. O funcionamento da interface gráfica, desde a colocação dos parâmetros

iniciais até à obtenção de resultados, pode ser compreendido a partir do fluxograma presente na Figura 8.

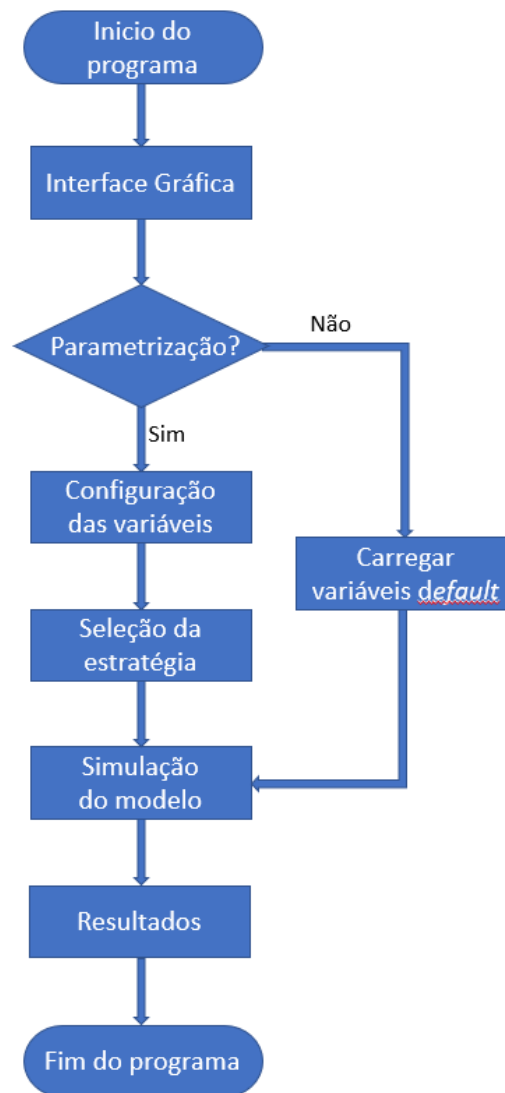


Figura 8: Fluxograma para a Interface Gráfica Desenvolvida

A Página Inicial (Figura 9) da Interface Gráfica foi desenhada de forma a fazer uma breve apresentação do trabalho e o seu principal objetivo. Descreve ainda as funcionalidades da aplicação e como configurá-la, incluindo dois botões que podem ser seleccionados. O botão *Parameterization* abre uma nova janela onde é possível configurar e alterar as variáveis do modelo. O botão *Run* tem como função dar início ao processamento do modelo.

Presentation

This application has the purpose of simulating the boarding of an Airbus 320, applying different types of strategies, with the objective of reducing the boarding time of an airplane, avoiding interferences between passengers. It is intended to use this simulation as a decision support tool, making it possible to change parameters that allow testing different configurations and consequent impact assessment.

Running a Simulation

As this application starts, the strategy 6 - Steffen (see tab Boarding Strategies on Parametrization menu) is already preloaded, you just need to click the 'Run' button.

If you want do choose other Strategy or change any default configuration click on the 'Parameterization' button.

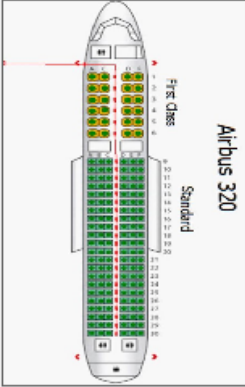
-On the 'Passengers Arrival' tab it is possible to customize the passengers parameters.

-On the 'Boarding Strategies' tab you can select and load other strategies. When you click the 'Apply Strategy' button, values will be loaded to run the simulation accordingly.

-If you select any other tab, you can see the loaded values. To adjust them, fill the fields and click the 'Change' button.

Afterwards just return to this screen and hit the 'Run' button.

Good Simulations!!




Developed By

Name: Hélio Gabriel Rocha Moreira

Email: 1161131@isep.ipp.pt

Oriented By: Prof. Dr. Luís Pinto Ferreira



Parameterization

Run

Figura 9: Interface Gráfica - Página Inicial

Caso seja selecionado o botão de parametrização da página inicial, será aberta a página que se encontra na Figura 10. Esta página inicia-se na aba *Passengers Arrival* onde é possível introduzir informação relevante relativamente aos passageiros, como: o número de passageiros que embarcam no avião; a percentagem de passageiros que leva bagagem de mão; se existem passageiros prioritários e, caso sim, quantos; e, por fim, ativar o modo de doenças infecciosas em caso de uma pandemia. Este último campo representa a aplicação de medidas contra doenças como o COVID-19. Este modo de operação não ocupa os assentos do meio do avião, exceto quando existe famílias com pelo menos 3 elementos. Para além de ocupar um lugar que antes se encontrava vago, a família tem a agora a oportunidade de viajar junta. Este ponto, para além de ser relevante quando as famílias viajam com crianças, permite também às companhias aéreas incrementar o número de passageiros possíveis no voo e, por sua vez, compensar algumas perdas pelos lugares vazios.

Figura 10: Interface Gráfica - Parametrização, *tab Passengers Arrival*

A aba *Strategies* permite ao utilizador acesso à área de definição de estratégias (Figura 11), onde pode escolher a estratégia de embarque a adaptar o seu modelo de simulação. Após a revisão de literatura do capítulo anterior, foram implementadas 7 estratégias de embarque no modelo de simulação, entre elas: *random*, *back-to-front*, *outside-In*, *reversed pyramid*, *blocks*, *Steffen* e *modified optimal*. Para a escolha destas estratégias definiram-se alguns critérios, nomeadamente:

- Facilidade de Implementação;
- Utilização pelas companhias aéreas;
- Rapidez no embarque.

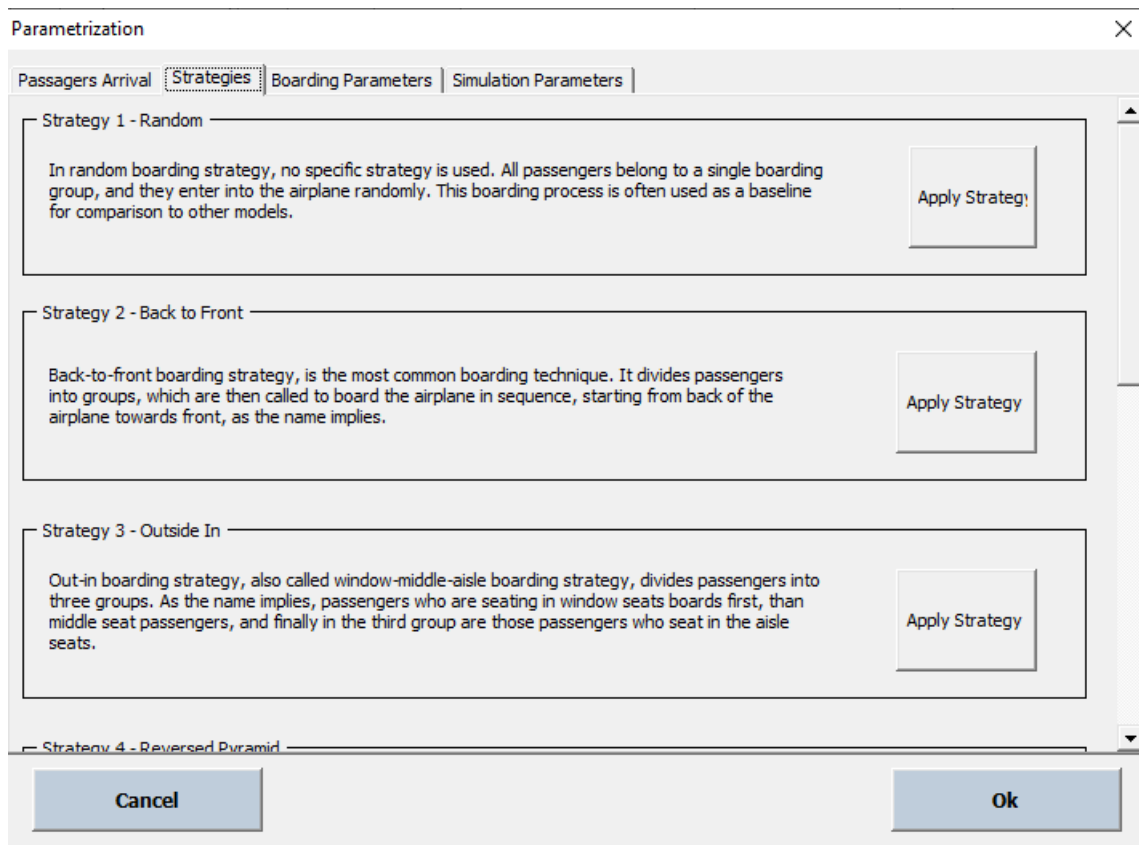


Figura 11: Interface Gráfica - Parametrização, *tab Strategies*

A aba *Boarding Parameters*, (Figura 12), possibilita ao utilizador alterar os valores referentes a 3 taxas de tempo que se efetuam durante o embarque, sendo estas: taxa de tempo relativamente à chegada dos passageiros à porta de embarque; taxa de tempo que os passageiros demoram a guardar a bagagem de mão no local apropriado; taxa de tempo correspondente às interferências de assento, caso esta ocorra. Estas taxas de tempo têm um valor pré-definido, e caso o utilizador o queira alterar, terá de seleccionar uma nova distribuição de tempo (ex: Normal; *Poisson*; Exponencial; Triangular; etc.) e preencher os campos específicos. Por fim, o utilizador apenas terá que clicar no botão correspondente para atualizar os parâmetros.

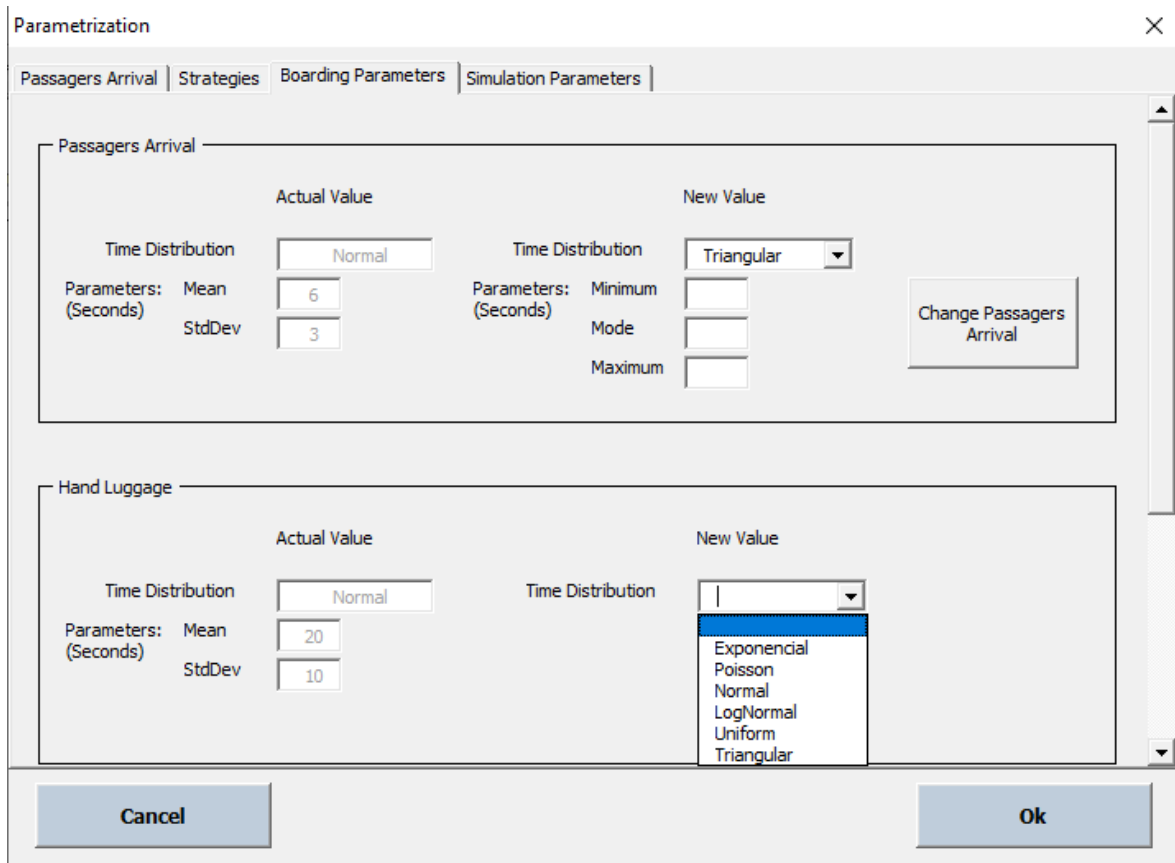


Figura 12: Interface Gráfica - Parametrização, *tab Boarding Parameters*

A aba *Simulation Parameters* (Figura 13), permite ao utilizador alterar as definições de simulação do *software*, como por exemplo o número de replicações que deseja realizar até que o modelo finalize o seu processamento. O utilizador também pode verificar os valores que se encontram pré-carregados no sistema através do botão *Read Variables*. Para alterar estes, basta preencher os campos com os valores que deseja e carregar no botão *Change Values*.

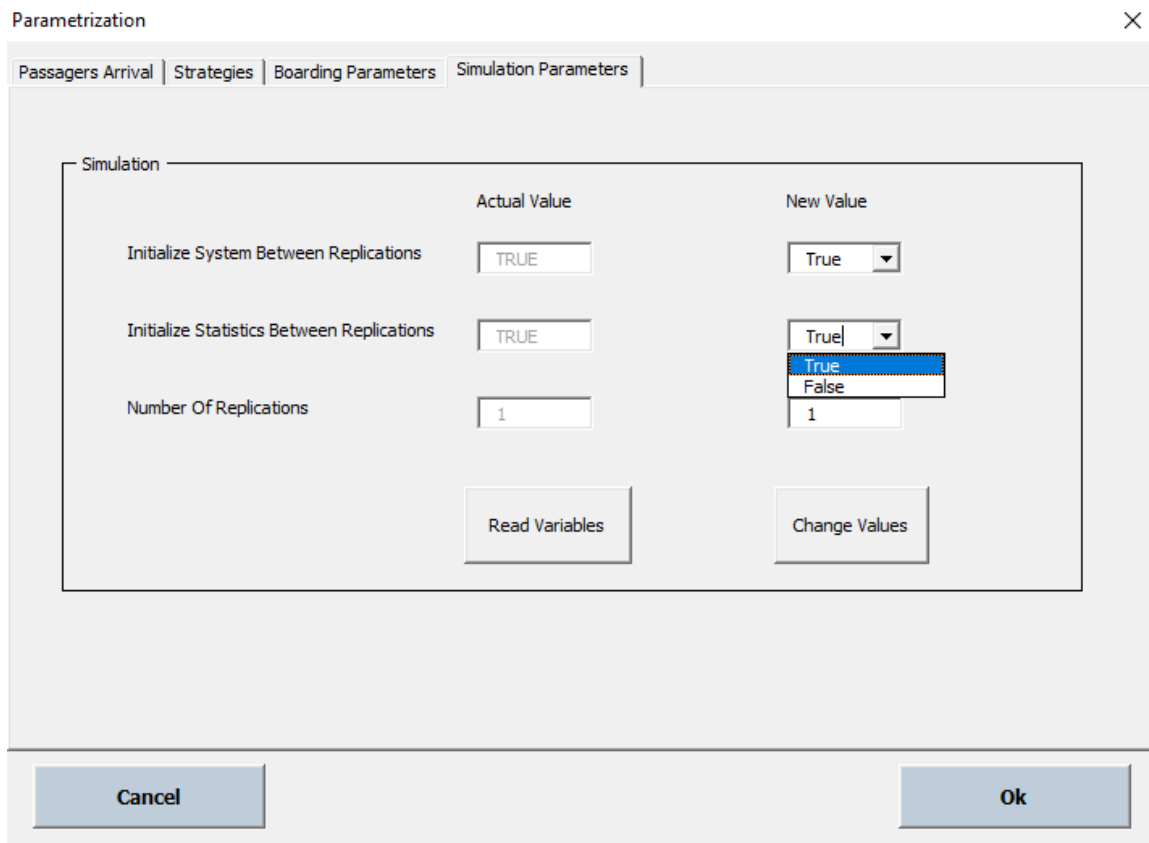


Figura 13: Interface Gráfica - Parametrização, *tab Simulation Parameters*

Assim que modelo finalize o seu processamento, é exibida uma janela, (Figura 14), com os resultados adquiridos da simulação com as variáveis previamente selecionadas. Esta janela apresenta a média, o valor máximo e mínimo dos valores obtidos através das várias replicações do sistema. Estes resultados serão analisados no capítulo 4 desta dissertação.

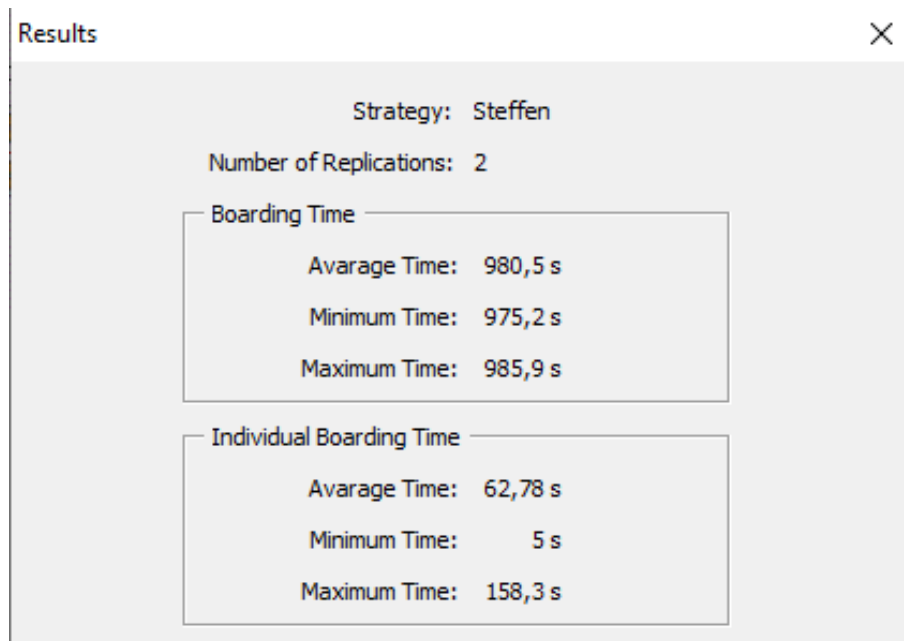


Figura 14: Interface Gráfica – Resultados

3.2. ANIMAÇÃO

A animação do modelo de simulação tem como objetivo a representação dinâmica animada do mesmo, mostrando, ao longo do tempo de execução, o embarque dos passageiros no avião. Esta ferramenta é importante na modelação de sistemas, pois fornece ao utilizador uma ilustração visual e dinâmica capaz de gerir os modelos. Desta forma, é possível obter um *feedback* do sistema construído, verificando, se este corresponde ao idealizado (Ares et al., 2012). As animações gráficas, também permitem a deteção de erros no desenvolvimento do modelo de simulação. Em conclusão, a animação é a ferramenta do *software* que poderá ajudar líderes a tomar decisões de uma forma atraente e apoiada em dados reais e permite exibir resultados, verificar e validar o modelo construído (Leathrum et al., 2019).

No desenvolvimento da animação do modelo de simulação foram utilizados os seguintes blocos do ARENA: *Station*, *Segments*, *Resources* e *Entities*. O bloco *Station* tem como propósito a entrada e saída de entidades, sendo possível a realização de tarefas. Os *Segments* têm como função interligar as estações entre si, criando uma distância para um transportador. Esta distância varia dinamicamente, sendo a animação ajustada em função da velocidade das entidades entre estações. Os *Resources* são módulos de dados que guardam informação sobre o estado e a disponibilidade dos recursos. O bloco *Entity* corresponde à animação gráfica dos passageiros a transportar no modelo.

É possível visualizar a animação do modelo desenvolvido através da Figura 15 demonstrando a simplificação da planta de um avião. A imagem com o mapa de assentos que serviu como base para o modelo desenvolvido pode ser obtida em Seatplans.com, (2021).

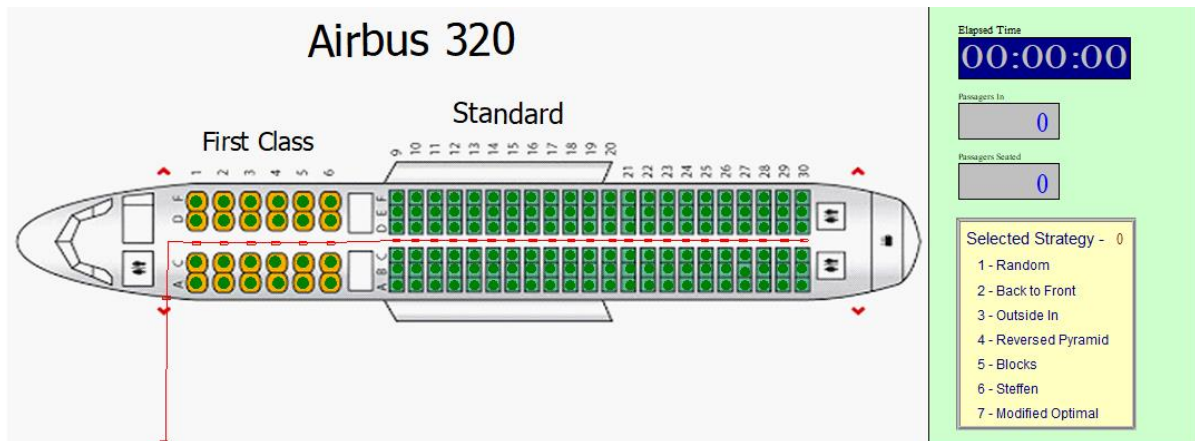


Figura 15: Exemplo da Animação do Modelo Desenvolvido - Fonte imagem base do Airbus A320: (Seatplans.com, 2021)

Como pode ser visto na Figura 15, o avião utilizado foi o Airbus A320. Para a escolha deste foram tidos em consideração vários parâmetros, entre eles:

- Voos de curta e média distância;
- Presente no quotidiano;
- Utilização por uma companhia aérea portuguesa;

Segundo a TAP AIR Portugal, (2021), o Airbus A320 cumpre com os requisitos acima definidos, sendo ainda o avião mais utilizado por esta companhia aérea. Tendo o modelo de avião definido, foi necessário adquirir uma planta do mesmo. A disposição da cabina adquirida é constituída por 30 fileiras de passageiros e 156 lugares: as fileiras 1 a 6 têm 4 assentos, onde se encontram os passageiros de primeira classe; as fileiras 7 e 8 não têm lugares disponíveis fazendo uma divisão entre classes; e as restantes fileiras têm 6 assentos, onde se encontram os passageiros de classe económica. Os assentos são numerados por fila e por uma das letras A – F de bombordo a estibordo.

Assim sendo, neste caso, as *entities* correspondem aos passageiros que embarcam no avião. Cada *station* corresponde a cada fila de passageiros, sendo visível no corredor do avião (a vermelho), passando por os pontos de início, interceção e fim de linha. Quanto aos *segments*, estes são representados pelas linhas vermelhas que conectam as estações, criando assim um

possível percurso para as entidades. Cada *resource* corresponde aos lugares disponíveis no avião podendo esses lugares estarem livres, em processo ou ocupados.

O modelo de simulação também apresenta um pequeno menu com informações que ocorrem durante a simulação do modelo. Como pode ser visto na Figura 16, estas informações contêm um *clock* que conta o tempo que passou desde o início do embarque e variáveis que informam qual foi a estratégia utilizada, quantos passageiros entraram no avião e quantos já estão sentados.

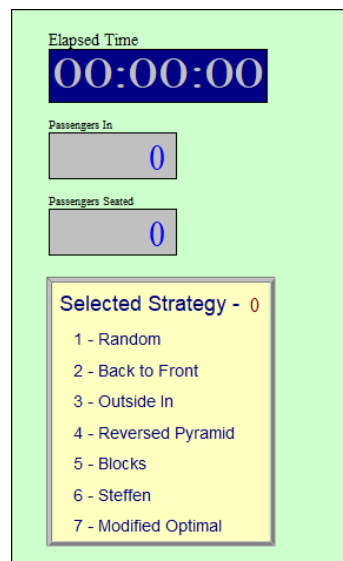


Figura 16: Menu de Animação Informativo

3.3. MODELO LÓGICO

A partir da informação adquirida, quer sobre o processo de embarque de passageiros num avião, quer sobre as estratégias de embarque existentes, foi desenvolvido um modelo de simulação que procura representar a aplicação de diferentes estratégias de embarque de passageiros num Airbus A320. Este modelo tem como principal propósito proporcionar a análise e possíveis melhorias a implementar nas companhias aéreas. Na Figura 17 está presente o modelo desenvolvido, onde foi utilizado o *software* de simulação ARENA.

Para uma melhor compreensão e pelo tamanho do modelo construído, o desenvolvimento é repartido em diversas etapas, tomando a seguinte ordem:

- 1ª Etapa: Criação das entidades, passageiros, na porta de embarque;
- 2ª Etapa: Chegada das entidades à porta do avião e atribuição de um lugar;
- 3ª Etapa: Transporte das entidades até ao lugar de destino e colocação da bagagem de mão no devido compartimento;
- 4ª Etapa: Interferências de assento;
- 5ª Etapa: Registo dos tempos de embarque necessários.

1ª Etapa: Criação das entidades, passageiros, na porta de embarque.

Nesta fase o objetivo foi criar as entidades que representam os passageiros e iniciar o transporte destes desde a porta de embarque até à entrada do avião e, por sua vez, até ao lugar de destino. Para tal utilizaram-se os blocos *Create*, *Station*, *Access* e *Convey*, como se pode observar na Figura 18.

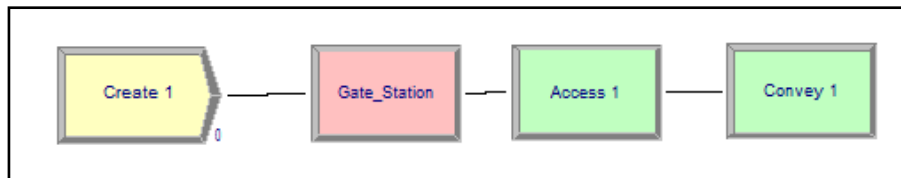


Figura 18: Conjunto de Blocos que criam as Entidades Passageiros

O bloco *Create* tem como função a criação de entidades. Como se pode ver na Figura 19, o tipo de entidade é “Entity 1” que representa os passageiros, sendo que estes são criados num intervalo de tempo correspondente ao valor da variável “Alter_Time_Entity”, até um máximo de “passagersmax”, que corresponde ao número total de passageiros. As variáveis referidas podem ser editadas na interface gráfica do sistema, permitindo uma versatilidade do modelo, mas têm o valor pré-definido de Norm (6,3) para a variável “Alter_Time_Entity” (variável com uma distribuição Normal de média 6 e desvio padrão 3) e 156 para a variável “passagersmax”. O valor desta distribuição foi escolhido porque segundo Law & Kelton,

(2007) os passageiros chegam à porta do avião a cada 6 segundos. Como a variável “Alter_Time_Entity” é do tipo *string* é necessário convertê-la para o tipo *real* e para isso utilizou-se a função *Eval()*.

The image shows a 'Create' dialog box with the following parameters:

- Name: Create 1
- Entity Type: Entity 1
- Time Between Arrivals: Type: Expression, Expression: Eval(Alter_Time_I), Units: Seconds
- Entities per Arrival: 1
- Max Arrivals: passagersmax
- First Creation: 0

Figura 19: Parâmetros do bloco *Create*

O segundo bloco, *Station*, define a estação “Gate”, como o ponto de início do processo referente aos passageiros. O terceiro bloco, *Access*, aloca uma célula da transportadora “Conveyor 1” para que a entidade (passageiro) se possa movimentar de uma estação para outra (Figura 20). Caso a primeira célula da transportadora esteja ocupada, a entidade cria uma fila de espera até que esta volte a ficar disponível.

The image shows an 'Access' dialog box with the following parameters:

- Name: Access 1
- Conveyor Name: Conveyor 1
- # of Cells: 1
- Queue Type: Queue
- Queue Name: Access 1.Queue

Figura 20: Parâmetros do bloco *Access*

O último bloco, *Convey*, é responsável pelo movimento da entidade criada da estação de origem “Gate”, até à estação de destino, que neste caso é a estação “Entrance”. Esta representa a porta do avião (Figura 21). O objetivo do *Convey* é transportar as entidades até ao assento destinado. O atraso de tempo para transportar a entidade de uma estação para a próxima é baseado na velocidade do transportador e na distância entre as estações. A velocidade está disponível no módulo da transportadora (Figura 22) e a distância está disponível no módulo dos segmentos (Figura 23).

Figura 21: Parâmetros do bloco *Convey*

Conveyor - Advanced Transfer										
	Name	Segment Name	Type	Velocity	Units	Cell Size	Max Cells Occupied	Accumulation Length	Initial Status	Report Statistics
1	Conveyor 1	Conveyor 1.Segment	Accumulating	1	Per Second	1	1	1	Active	<input checked="" type="checkbox"/>

Figura 22: Módulo *Conveyor*

Segment - Advanced Transfer			Next Stations	
Name	Beginning Station		Next Station	Length
1	Conveyor 1.Segment	Gate	1 Entrance	7
			2 Aisle1	5
			3 Aisle2	2

Figura 23: Módulo *Segment*

2ª Etapa: Chegada das entidades à porta do avião e atribuição de um lugar.

Quando as entidades chegam à estação de destino “Entrance”, através da transportadora, o conjunto de blocos representado na Figura 24 atua. A principal função deste conjunto é atribuir um lugar disponível do avião a cada entidade.

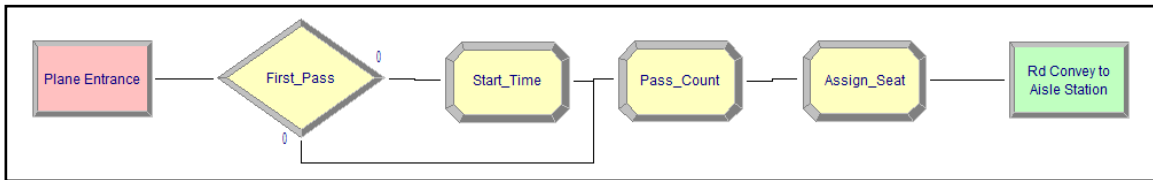


Figura 24: Conjunto de Blocos que atribuem lugares às entidades.

O primeiro bloco *Station*, define a entrada de entidades na estação, o segundo bloco, *Decide*, tem como função decidir qual o caminho a percorrer a partir deste ponto, ou seja, assim que uma entidade atinge este bloco, apenas poderá prosseguir um caminho entre as várias hipóteses existentes. Neste *Decide*, representado na Figura 25, apenas existem 2 rotas possíveis que variam dependentemente de uma condição. Caso a variável “passengerscount” seja igual a 0, a entidade prosseguirá para o terceiro bloco. Caso contrário, irá para o quarto bloco do conjunto.

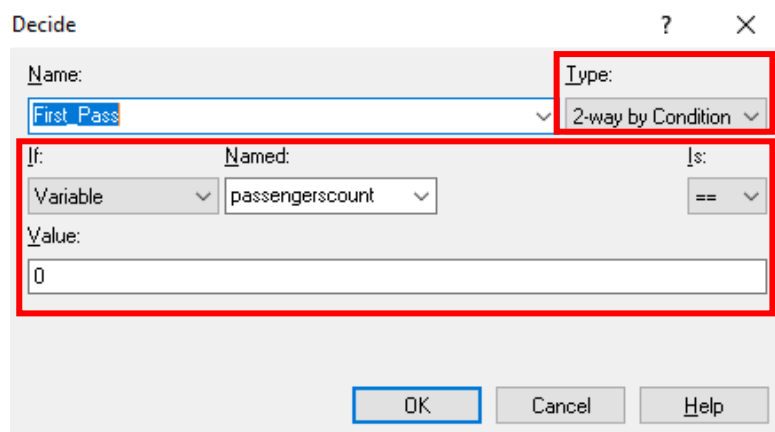


Figura 25: Parâmetros do bloco *Decide*, *First_Pass*

O terceiro, quarto e quinto blocos do conjunto são do tipo *Assign*. Este modulo é usado para atribuir novos valores a variáveis, a atributos, a entidades ou a outras variáveis do sistema. Podem ser realizadas múltiplas operações num único modulo *Assign*. O terceiro bloco, representado na Figura 25, regista o tempo que a primeira entidade passa por este bloco através da variável TNOW (*Current Simulation Time*). O TNOW regista o tempo de relógio conforme o avanço do modelo de simulação. O registo irá ser guardado na variável “InicialTime”, que será necessária, mais tarde, para o cálculo do tempo de embarque.

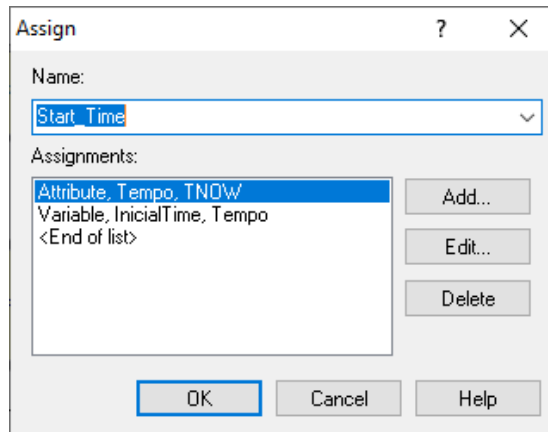


Figura 26: Parâmetros do bloco *Assign*, *Start_Time*

O quarto bloco, representado na Figura 26, também vai registrar o tempo atual da variável TNOW, mas desta vez, registra o tempo de todas as entidades. Este elemento, será também necessário mais tarde, para calcular o tempo individual de embarque. Para obter um registro diferente para cada entidade, foi atribuído o atributo “Time”. Este bloco é responsável por fazer a contagem dos passageiros que entram no avião. Isto é realizado através do incremento da variável “passengerscount”, como pode ser visualizado através da Figura 28.

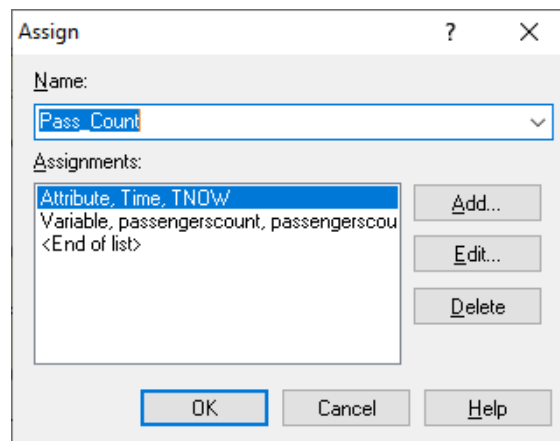


Figura 27: Parâmetros do bloco *Assign*, *Pass_Count*

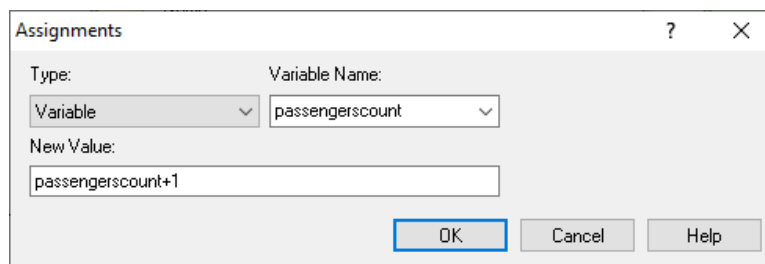


Figura 28: Incremento da variável *passengerscount*

O último *Assign*, “Assign_Seat”, representado na Figura 29, é responsável por atribuir um assento disponível às entidades. Esse assento é caracterizado por dois atributos, “Line_Attribute” e “Seat_Attribute”. O “Line_Attribute” representa a fila de assentos que está atribuído ao lugar e pode tomar valores entre 1 a 28. O “Seat_Attribute” representa o lugar da entidade na fila de destino e toma valores entre 1 a 6. Por exemplo, se a entidade tiver o lugar “13F”, o seu “Line_Attribute” vai ser “13” e o seu “Seat_Attribute”, “6”. O último atributo toma o valor “6”, porque “F” é a sexta letra do alfabeto. Antes dos valores dos assentos serem passados para os atributos, estes encontram-se na variável 3D “SeatArray”. Esta variável armazena todos os lugares dos passageiros na respetiva ordem, ou seja, a ordem pela qual os passageiros devem entrar no avião, formando, no máximo, uma variável de 156 filas e 2 colunas. Na primeira coluna está armazenado o valor da fila que dá origem ao “Line_Attribute” (Figura 30), e na segunda coluna está armazenado o valor do assento que dá origem ao “Seat_Attribute” (Figura 31). A ordem dos passageiros e o preenchimento da variável é realizado através de código VBA e será analisado na seção 3.4 desta dissertação. A variável “CountLine” que aparece nas imagens referidas é incrementada sempre que uma nova entidade chega a este bloco, permitindo que seja possível seguir a ordem dos lugares contidos na variável “SeatArray”.

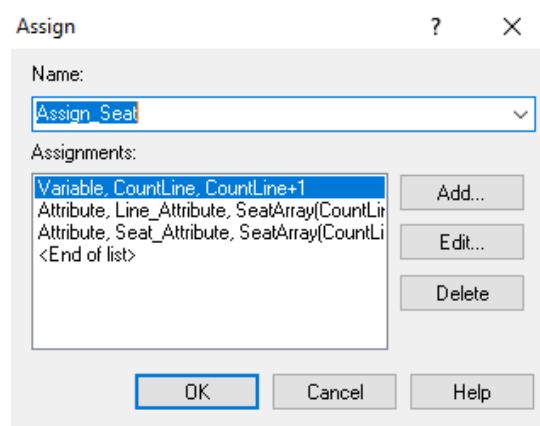


Figura 29: Parâmetros do bloco *Assign*, *Assign_Seat*

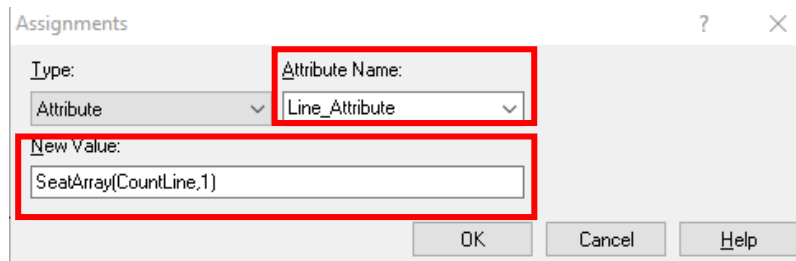


Figura 30: Atribuição do *Line_Attribute*

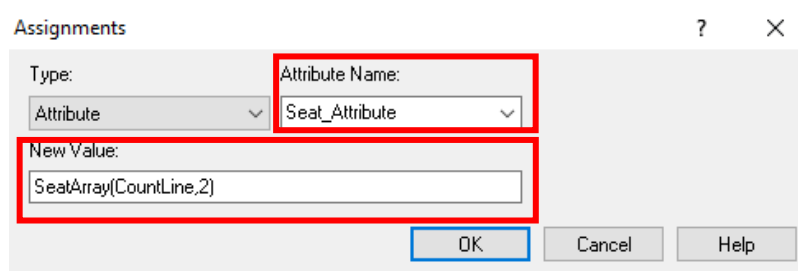


Figura 31: Atribuição do *Seat_Attribute*

O último bloco, *Convey*, “Rd convey to Aisle Station”, movimenta as entidades desde a porta do avião, estação “Entrance”, até à primeira fila de assentos do avião, a estação “Aisle1” (Figura 32).

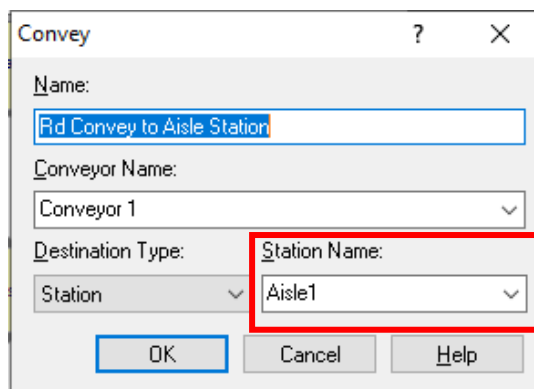


Figura 32: Parâmetros do bloco *Convey*, *Rd Convey to Aisle Station*

3ª Etapa: Transporte das entidades até ao lugar de destino e colocação da bagagem de mão no devido compartimento.

Devido à extensão do terceiro grupo de blocos, representado na Figura 17, este é subdividido em diferentes etapas. A terceira etapa representa o conjunto de blocos em que as entidades percorrem as várias filas do avião até que estas cheguem ao lugar de destino e coloquem a sua bagagem de mão no devido compartimento. A Figura 33 representa o conjunto de blocos responsáveis pela terceira etapa.

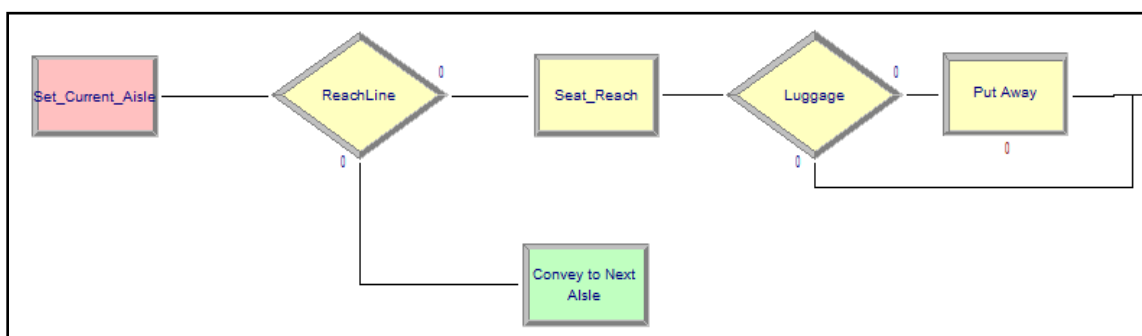


Figura 33: Conjunto de blocos que transporta os passageiros até ao lugar de destino

O primeiro bloco, *Station*, “Set_Current_Aisle” (Figura 34), define um conjunto de estações. Estas representam todas as estações existentes no corredor do avião. Para cada fila de passageiros, existe uma estação destinada, por exemplo, a primeira fila de assentos tem associada a estação, “Aisle1”. Como existem 28 filas de assentos, este bloco apresenta o mesmo número de estações. Estas estações são diferenciadas através do atributo “Current_Line”, que adquire um valor distinto dependendo da estação onde a entidade se encontra. A primeira vez que uma entidade chega e este bloco será sempre na estação

“Aisle1” e a última será a estação onde se encontra o seu assento, passando todas as estações por entre as anteriores.

The 'Station' dialog box contains the following fields and options:

- Name: Set_Current_Aisle
- Station Type: Set
- Set Name: Current_Aisle_Station
- Save Attribute: Current_Line
- Station Set Members: Aisle1, Aisle2, Aisle3, Aisle4, Aisle5, Aisle6, Aisle7, Aisle8, Aisle9 (all with 'Yes' status)
- Buttons: Add..., Edit..., Delete
- Buttons: OK, Cancel, Help

Figura 34: Parâmetros do bloco *Station*, *Set_Current_Aisle*

O segundo bloco, *Decide*, “ReachLine” (Figura 35), verifica se a entidade já chegou à sua fila de destino. O resultado decide a rota que a entidade vai tomar. Como já foi visto anteriormente, cada entidade tem atribuída uma variável denominada “Line_Attribute”, que contém a informação da fila a que pertence o seu assento. Este bloco compara este valor com o atributo que representa a estação onde se encontra de momento, “Current_Line”. Se o valor for equivalente, significa que a entidade chegou à fila de destino e termina assim o seu percurso. Caso contrário, a entidade segue para a próxima estação.

The 'Decide' dialog box contains the following fields and options:

- Name: ReachLine
- Type: 2-way by Condition
- If: Attribute
- Named: Line_Attribute
- Is: ==
- Value: Current_Line
- Buttons: OK, Cancel, Help

Figura 35: Parâmetros do bloco *Decide*, *ReachLine*

Na situação onde as entidades ainda não tiverem chegado à sua estação de destino, estas vão para o bloco *Convey*, “Convey to Next Aisle”, representado na Figura 36. Este bloco é responsável por movimentar os passageiros até à fila onde se encontra o seu lugar no avião. Sempre que a entidade passa por este bloco é enviada para a estação seguinte. Por exemplo, se a entidade estiver na estação “Aisle1”, esta vai ser enviada para a estação seguinte, “Aisle2”. Para isto, é incrementado o valor do atributo “Current_Line”.

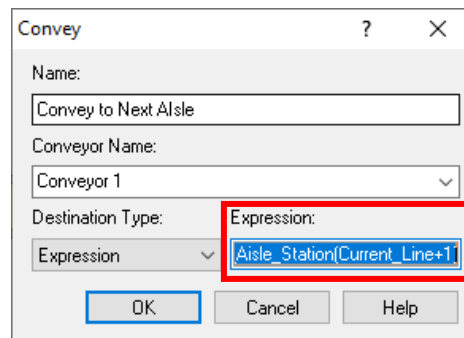


Figura 36: Parâmetros do bloco *Convey*, *Convey to Next Aisle*

Caso as entidades cheguem à estação de destino, estas seguem para o bloco *Seize*, “Seat_Reach”, representado na Figura 37. Este modulo aloca unidades dos recursos para as entidades. Como já foi dito, os recursos neste modelo simulam os assentos, que podem estar disponíveis ou não. Assim sendo, ao passar por este bloco a entidade vai alocar o recurso correspondente ao lugar, alterando-o de disponível para ocupado.

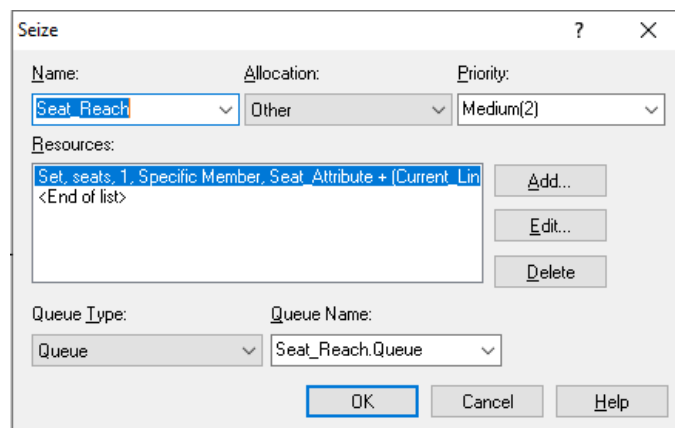


Figura 37: Parâmetros do bloco *Seize*, *Seat_Reach*

Devido à dimensão no número de lugares do avião foi criado um conjunto *set*, denominado “seats”, com todos os recursos existentes numa ordem específica. A ordem do conjunto começa com os assentos “1A”, “1B”, “1C”, e por aí fora, e termina com “28D”, “28E” e

“28F”. Para associar o recurso que a entidade vai ocupar ao conjunto criado, foi utilizada a expressão descrita em (Equação 1). Como já foi escrito anteriormente, o campo “Seat_Attribute” pode tomar valores entre 1 e 6, e o campo “Current_Line” pode tomar valores entre 1 e 28. Esta expressão foi inserida na janela “Resources” do bloco *Seize*, representado na Figura 38.

$$\text{Seats} = \text{Seat_Attribute} + (\text{Current_Line} * 6) - 6 \quad (\text{Equação 1})$$

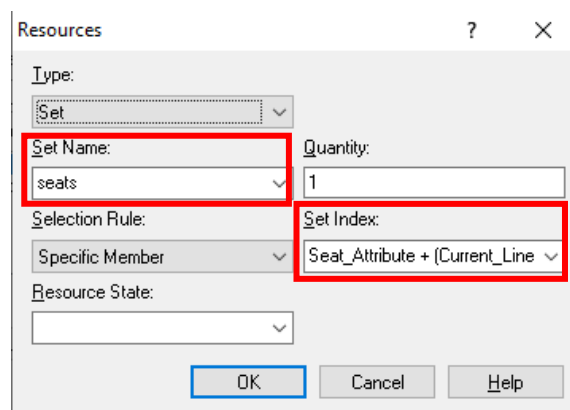


Figura 38: Alteração do estado do recurso

O bloco seguinte, *Decide*, “Luggage” (Figura 39), verifica se a entidade possui ou não bagagem de mão, baseando-se numa probabilidade. Ou seja, apenas uma percentagem de entidades vai seguir por uma rota, sendo o valor configurável através da interface gráfica.

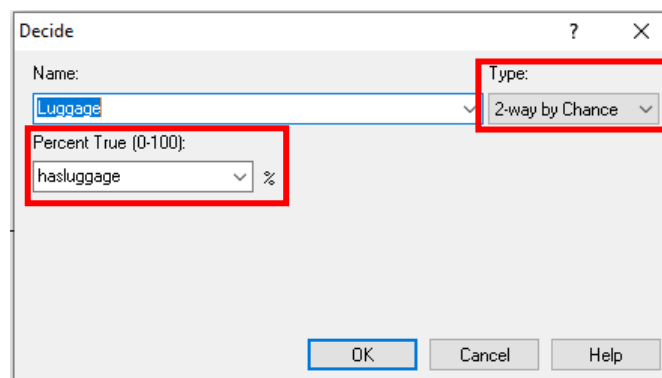
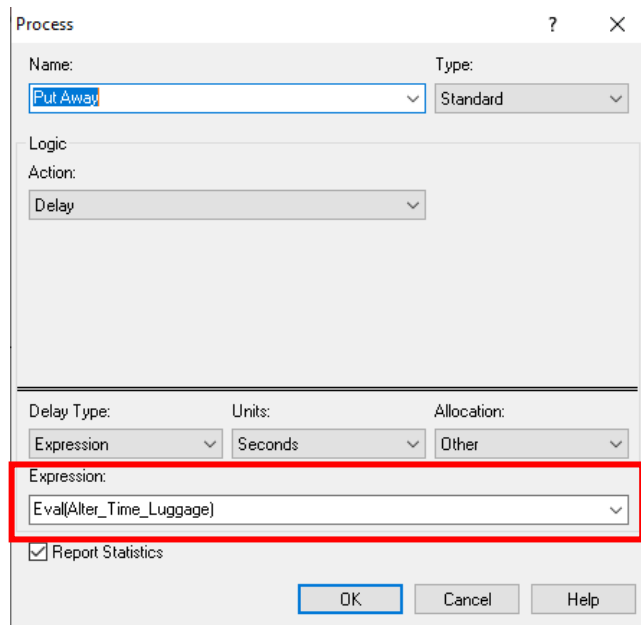


Figura 39: Parâmetros do bloco *Decide*, *Luggage*

Caso a entidade seja considerada com bagagem de mão será enviada para o bloco *Process*, “Put Away” (ver Figura 40). Este bloco vai ser responsável por simular o tempo que os passageiros demoram a guardar a sua bagagem de mão no devido compartimento, ficando prontos para se sentarem no lugar destinado. O intervalo de tempo que as entidades demoram a executar esta ação é definida pela variável “Alter_Time_Luggage” que também pode ser configurada através da interface gráfica. Esta variável tem de valor pré-definido uma distribuição de tempo normal de média 20 e desvio padrão 10.



The image shows a dialog box titled "Process" with the following fields and options:

- Name:** Put Away (dropdown menu)
- Type:** Standard (dropdown menu)
- Logic:**
 - Action:** Delay (dropdown menu)
- Delay Type:** Expression (dropdown menu)
- Units:** Seconds (dropdown menu)
- Allocation:** Other (dropdown menu)
- Expression:** Eval(Alter_Time_Luggage) (dropdown menu, highlighted with a red box)
- Report Statistics
- Buttons:** OK, Cancel, Help

Figura 40: Parâmetros do bloco *Process*, *Put Away*

4ª Etapa: Interferências de assento.

A quarta etapa representa o conjunto de blocos que simula as interferências de assento existentes durante o embarque, que é uma das principais causas de atrasos.

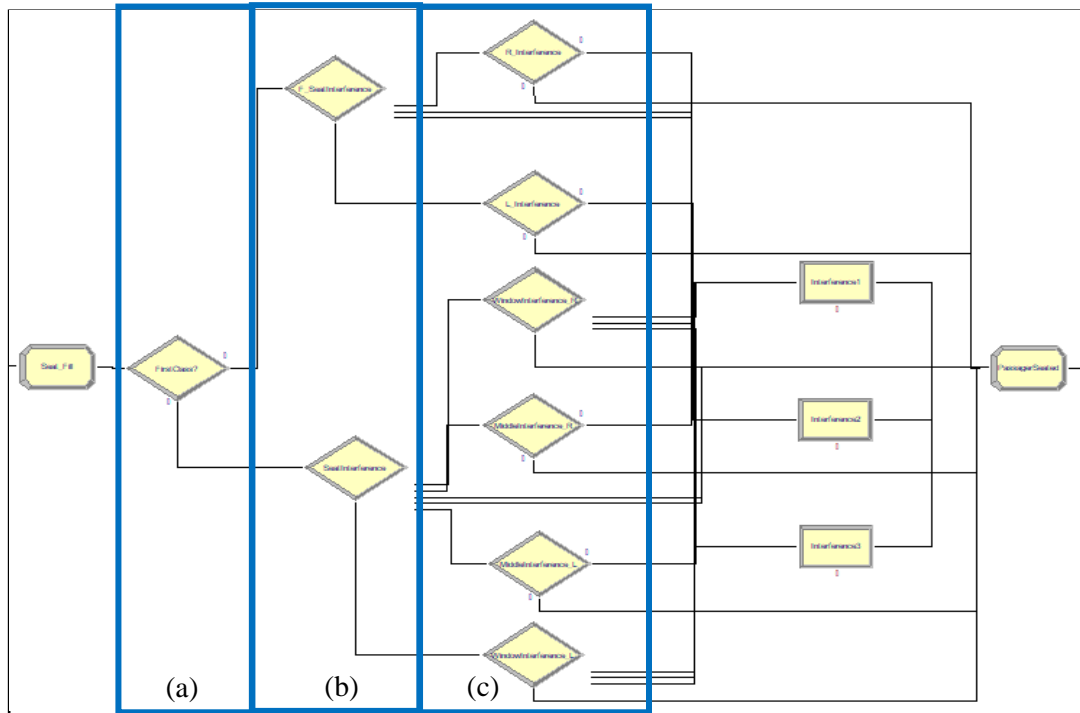


Figura 41: Conjunto de blocos que simulam as interferências de assento

Como pode ser visto através da Figura 41, as interferências de assentos são simuladas e distinguidas através de um conjunto de decisões. Em primeiro lugar é necessário saber quais são os passageiros que já estão sentados e os que ainda não estão. Para tal, foi utilizada a variável “Fill_Seats”, uma matriz de ordem 28 por 6 que guarda essa informação. Caso o elemento da matriz tenha o valor “0” significa que o assento está livre. Caso tenha o valor “1” o assento está ocupado. Quando a entidade chega ao primeiro bloco do conjunto *Assign*, incrementa o valor da matriz, correspondente ao seu assento, utilizando os seus atributos “Current_Line” e “Seat_Attribute” (Figura 42).

Figura 42: Preenchimento da matriz lugares no bloco *Assign*

De seguida, na Figura 41, estão representados um conjunto de vários blocos *decide*, divididos em vários grupos (blocos azuis). Este conjunto determina o tipo de interferência de assento (ver secção 2.2.3) que irá acontecer com a entidade em questão. O primeiro bloco de decisão, (a), separa os passageiros que embarcam na primeira classe (primeiras filas do avião), da classe económica. Isto acontece porque na primeira classe apenas existem interferências de assento do tipo 3. Então, o bloco representado na Figura 43, separa as entidades através do atributo “Current_Line”. Se o valor deste for menor que 7 seguirá por um caminho. Caso contrário, seguirá por outro.

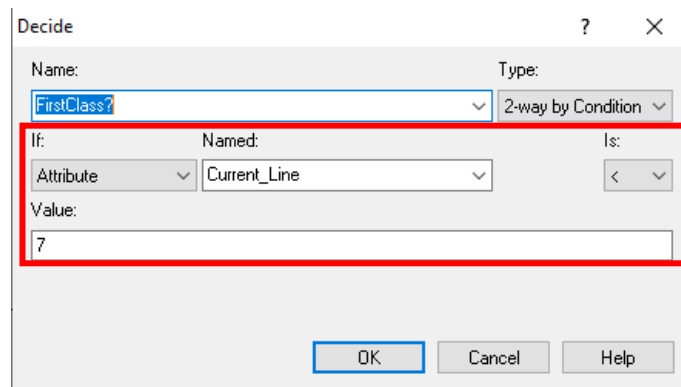


Figura 43: Parâmetros do bloco *Decide*, *FirstClass?*

Independentemente da classe do passageiro, a entidade encontrará um novo bloco de decisão, (b). Este bloco apenas verifica qual o assento da fila que a entidade vai tomar: se é o assento da janela, do meio ou do corredor, e qual o lado do corredor, se bombordo ou estibordo. Como existem 6 assentos numa fila de classe económica, existirão 6 caminhos possíveis. Estes caminhos ajudam a delinear se vão ocorrer interferências de assento e qual o tipo dessas interferências. Caso o passageiro ocupe o lugar junto ao corredor, não ocorrerá qualquer tipo de interferência. Caso o passageiro ocupe o lugar do meio, apenas poderá ocorrer uma interferência do tipo 4. Na Figura 44 estão representados os diversos caminhos que a entidade poderá tomar.

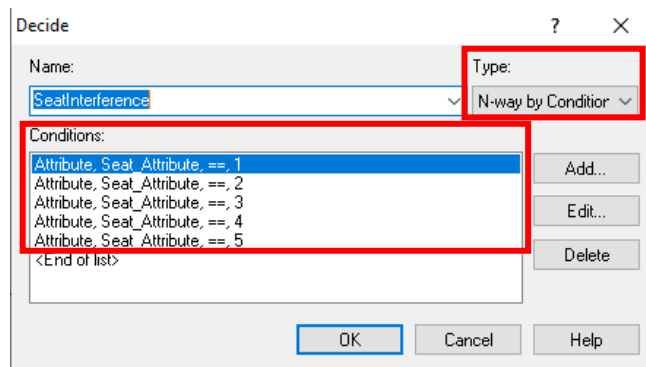


Figura 44: Parâmetros do bloco *Decide*, *SeatInterference*

Após a divisão das entidades em diferentes rotas, falta verificar se acontecerá interferências caso o passageiro se desloque para o assento do meio ou da janela. Para isso, utilizaram-se os blocos de decisão do grupo (c). É necessário saber qual é o tipo de interferência que está a acontecer para poder aplicar um tempo de atraso adequado. Além disso, é um processo que tem de ser aplicado a bombordo e a estibordo, justificando a grande quantidade de blocos de decisão neste grupo. A Figura 45 e Figura 46 representam exemplos dos blocos de decisão caso a entidade vá para o assento do meio ou da janela, respetivamente. Se a entidade vai para o assento do meio, apenas é verificado se o lugar do corredor está ocupado. Caso esteja ocupado, ocorre interferência, caso contrário não ocorre. Se a entidade for para o lugar da janela, para além de verificar se ocorre interferência, é necessário distinguir se esta é do tipo 1, 2 ou 3, averiguando as diferentes combinações possíveis entre os passageiros do meio e da janela.

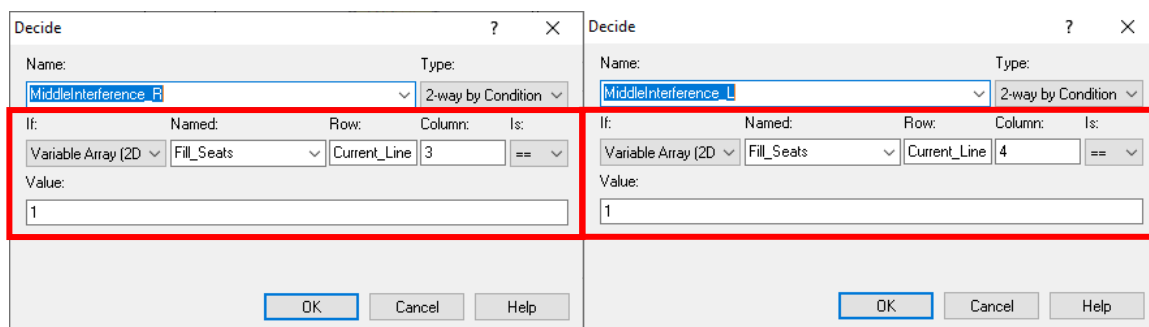


Figura 45: Exemplo de bloco *Decide*, *MiddleInterference*

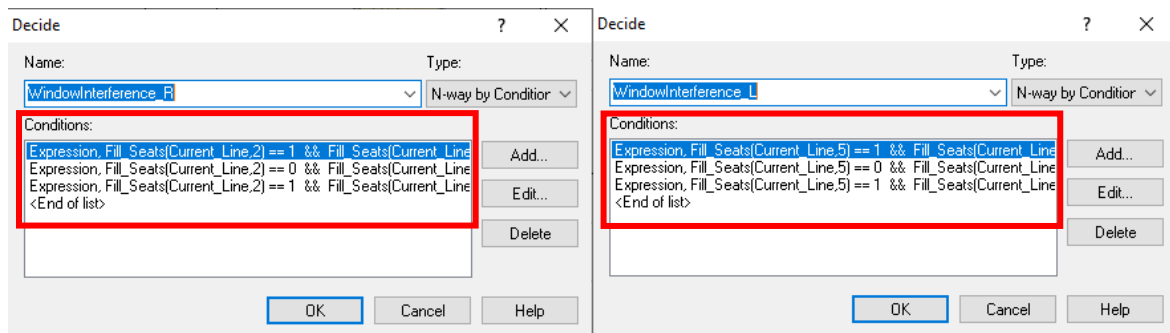


Figura 46: Exemplo de bloco *Decide*, *WindowInterference*

Depois do último bloco de decisão, (c), o número de interferências entre as entidades, caso existam, já é conhecido, assim como o seu tipo. Caso ocorram, as entidades serão enviadas para um bloco *Process*, e dependendo do seu tipo de colisão, será aplicado o devido atraso. Caso exista uma interferência do tipo 1, as entidades serão enviadas para o bloco representado na Figura 47. Se for do tipo 2, serão enviadas para o bloco da Figura 49. Caso seja aplicada uma interferência do tipo 3 ou 4, as entidades são enviadas para o bloco da Figura 48, pois estes tipos apresentam um tempo de atraso semelhante. O tempo de atraso aplicado é baseado no valor da variável “Alter_Time_Seat”, podendo esta ser alterada através da interface gráfica. Esta variável apresenta uma distribuição triangular de moda 11, mínimo 10 e máximo 13 como valores predefinidos. Esta distribuição foi também utilizada como valor pré-definido para uma interferência do tipo 2. Para a interferência do tipo 1, o conjunto de valores da variável foi duplicado, gerando uma distribuição triangular de moda 22, mínimo 20 e máximo 26. Para o tipo 3 e 4 foi decrementado uma unidade à variável, obtendo uma distribuição de moda 10, mínimo 9 e máximo 12. Os valores mencionados do tempo de atraso das interferências de assento foram baseados no estudo de Cotfas et al., (2020).

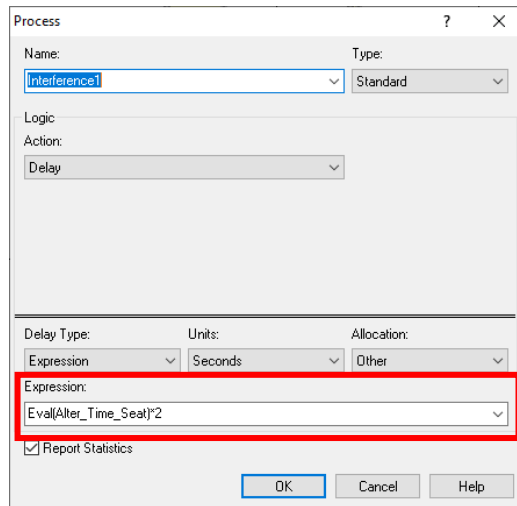


Figura 47: Interferência do tipo 1

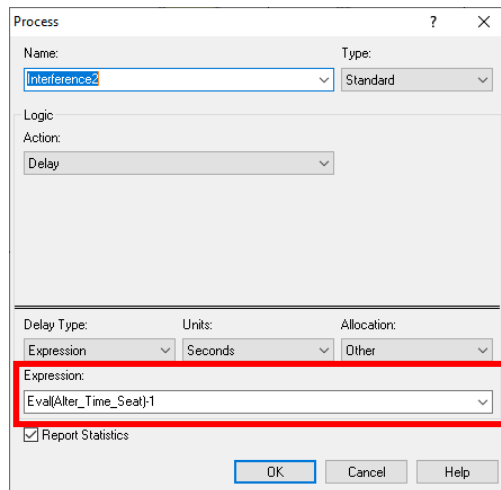


Figura 48: Interferência do tipo 3 e tipo 4

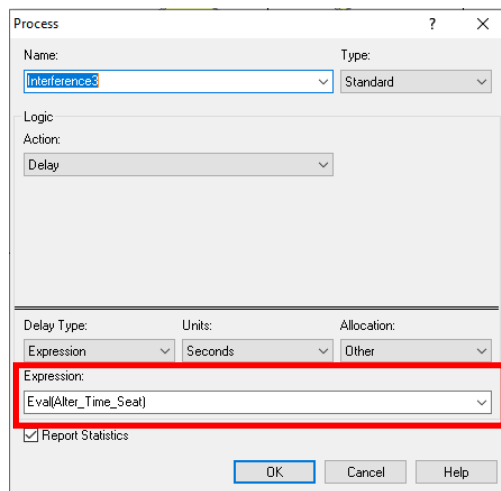


Figura 49: Interferência do tipo 2

Concluindo o processo de interferência de assentos, o passageiro está finalmente pronto a sentar-se no seu lugar de destino. Para isso, altera-se novamente o estado do recurso correspondente, passando de ocupado para “Seated” (Figura 51). Assim, com a mudança de estado anterior, utilizou-se a expressão descrita na Equação 1 para associar a entidade ao respetivo recurso. O bloco responsável por realizar esta alteração é o *Assign*, “PassagerSeated”, representado na Figura 50. Este bloco também faz a contagem dos passageiros que já se encontram sentados, incrementando a variável “Pass_Seated”.

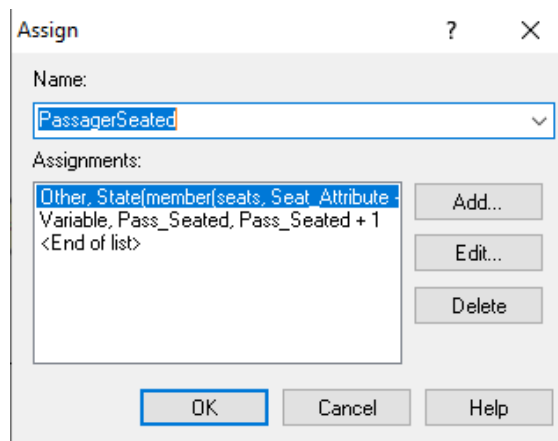


Figura 50: Parâmetros do bloco *Assign*, *PassagerSeated*

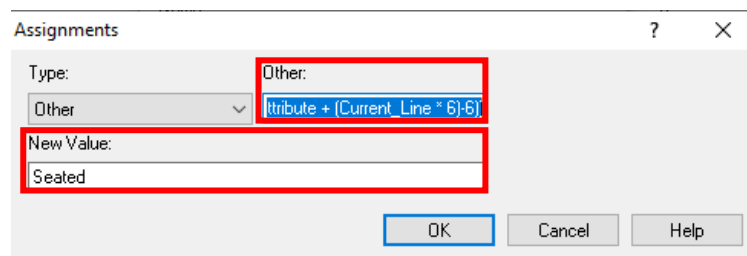


Figura 51: Tarefa para alterar o estado do recurso

5ª Etapa: Registo dos tempos de embarque necessários.

A quinta etapa, representada pela Figura 52, apresenta um conjunto de blocos que regista todos os tempos necessários para o embarque de passageiros no avião.

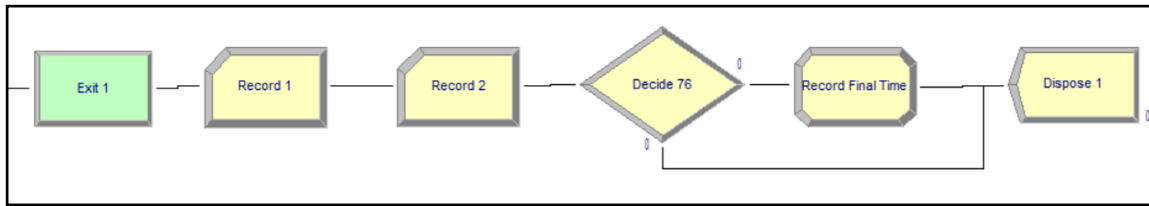


Figura 52: Conjunto de blocos que registra os tempos de embarque

O primeiro bloco, *Exit* (Figura 53), é responsável por libertar as entidades das células da transportadora. Como o passageiro acabou de se sentar, não necessita mais de ser transportado, desimpedindo, desta maneira, o corredor para os outros passageiros.

Figura 53: Parâmetros do bloco *Exit*

O segundo e terceiro blocos são do tipo *Record*. Estes módulos são usados para recolher informação sobre o modelo de simulação. O bloco representado na Figura 54 registra a cadência de saída das entidades, isto é, uma sucessão regular de intervalos de tempo em que duas entidades demoram a sentar-se. O bloco representado na Figura 55 registra o tempo individual de embarque de cada passageiro. Este último registo é possível, pois na segunda etapa foi iniciada a contagem do tempo através do atributo “Time” (Figura 27).

Figura 54: Registo da cadência de saída

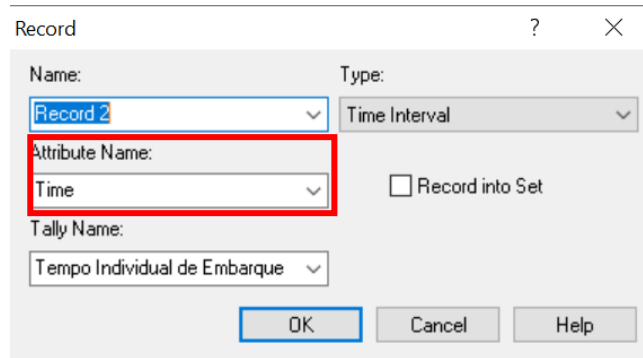


Figura 55: Registro do tempo individual de embarque

O quarto bloco *Decide*, “Last_Pass_Seated” (Figura 56), verifica se já se sentaram todos os passageiros do avião, comparando a variável “Pass_Seated” (número de passageiros sentados) com a variável “passengersmax” (número total de passageiros).

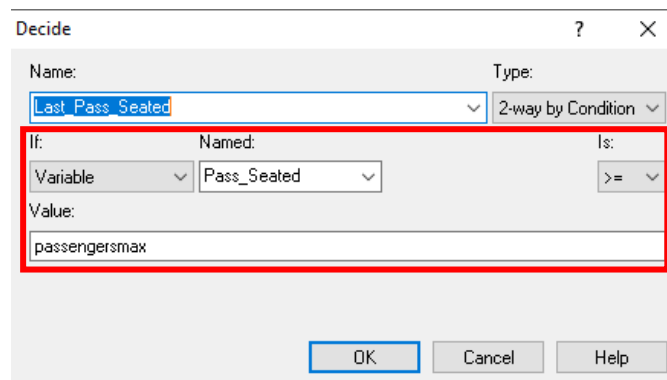


Figura 56: Parâmetros do bloco *Decide*, Last_Pass_Seated

O quinto bloco, *Assign*, “Record Final Time” (Figura 57), é responsável por registrar o tempo total de embarque e por calcular os valores da média, máximo e mínimo do tempo individual de embarque. Assim que a última entidade passa por este bloco é registrado o tempo que passou desde o início da simulação através da variável “TNOW”. Apesar deste registro, é necessário retirar o tempo que demorou a primeira entidade a chegar à porta do avião, porque segundo Van Landeghem & Beuselinck, (2002) “o tempo de embarque começa quando o primeiro passageiro entra no avião e termina quando o último passageiro se senta no assento designado”. O valor necessário para o cálculo foi registrado pela variável “InicialTime” na Figura 26 durante a segunda etapa. Assim sendo, apenas é necessário subtrair este valor ao tempo total de embarque. Este cálculo está representado na Figura 58.

Para obter a média, máximo e mínimo do tempo individual de embarque foram utilizadas as funções “TAVG(), TMAX() e TMIN()” do ARENA respectivamente. Estas funções ficam disponíveis, porque os tempos necessários ficaram registados no bloco *Record* da Figura 55.

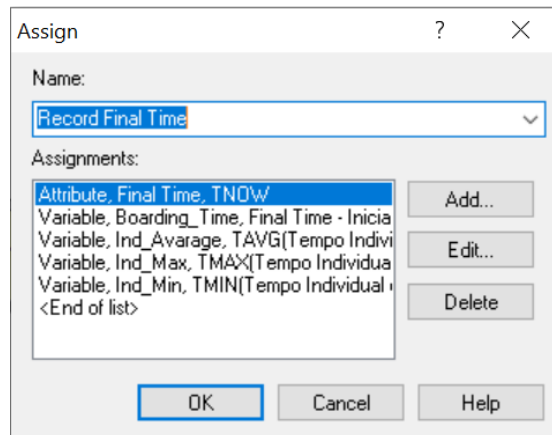


Figura 57: Parâmetros do bloco *Assign*, *Record Final Time*

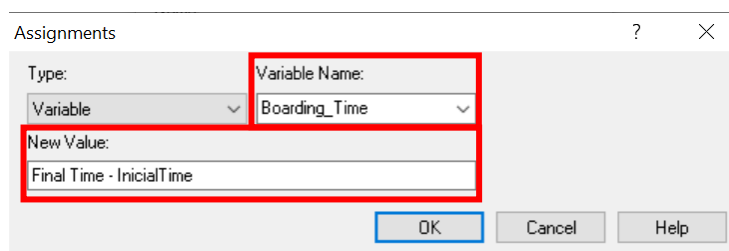


Figura 58: Cálculo do tempo de embarque

O último bloco *Dispose*, apresentado na Figura 59, representa o ponto final, terminando assim o processamento das entidades na simulação. Este modulo é indispensável em qualquer modelo construído com recurso ao ARENA.

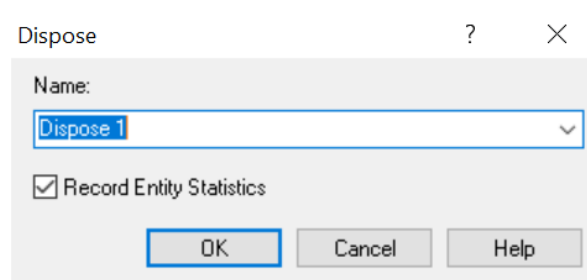


Figura 59: Parâmetros do bloco *Dispose*

3.4. CÓDIGO VBA DESENVOLVIDO

O VBA é uma ferramenta fornecida no ARENA que permite controlar as variáveis usadas no modelo. O VBA atua como uma linguagem de programação para serviços de usuário, permitindo a criação de funções para alterar as variáveis de processo do ARENA e do modelo criado.

Nesta fase foi desenvolvido o código VBA do ARENA para alterar os parâmetros na simulação do modelo desenvolvido, podendo este ser dividido em 2 partes: Código nos objetos e o código do modelo criado. O código nos objetos é utilizado para animar e personalizar a interface gráfica, como, por exemplo, dimensões ou alteração das cores dos objetos. O código do modelo criado permite a interligação da interface gráfica com as variáveis do sistema, possibilitando a leitura e a alteração de dados associados ao programa ARENA. Todo o código do modelo desenvolvido está disponível no Apêndice A.

Este código é responsável por gerar aleatoriamente os números dos assentos, mas são inseridos no modelo de simulação numa ordem de sequência específica, dependendo da estratégia de embarque escolhida na interface gráfica. São 7 as estratégias de embarque que podem ser implementadas no modelo, onde cada uma acomoda certas restrições. Para além disso, o código VBA também é responsável por restringir o número de entidades que embarca no avião, por garantir primeiro a entrada de passageiros da primeira, por alterar a ordem dos restantes caso existam prioritários e por aplicar as medidas necessárias contra doenças infecciosas como o COVID-19.

Para uma melhor compreensão das funções do código construído, este é repartido em diversas etapas, tomando a seguinte ordem:

- 1ª Etapa: Separação dos passageiros que embarcam na primeira classe;
- 2ª Etapa: Ordenar os passageiros que viajam na classe económica, dependendo da estratégia de embarque selecionada;
- 3ª Etapa: Separação dos passageiros prioritários da classe económica;
- 4ª Etapa: Aplicação contra doenças infecciosas;
- 5ª Etapa: Análise dos resultados obtidos.

O código das 4 primeiras etapas é executado no início de cada replicação do modelo de simulação. A replicação define-se como o número de vezes que a simulação é executada antes do programa terminar. Já a quinta etapa é executada no final de cada replicação.

1ª Etapa: Separação dos passageiros que embarcam na primeira classe.

O primeiro passo para a atribuição de um assento específico a um passageiro consiste em separar os que embarcam na primeira classe e os que embarcam na classe económica. Esta divisão é bastante útil porque, independentemente da estratégia escolhida, o processamento da primeira classe é sempre o mesmo. A divisão foi realizada de acordo com a planta da Figura 15, donde resulta a informação que se encontra na Tabela 4.

Tabela 4: Divisão de classes de embarque

Assentos disponíveis	
Primeira Classe	Classe Económica
24 Lugares Filas 1 a 6 Lugares de A a D	132 Lugares Filas 7 a 28 Lugares de A a F

No extrato de código seguinte encontra-se a divisão destes grupos de passageiros:

```
'First Class Seats
For i = 1 To fsc
  For j = 1 To 2
    bf(i - 1, j - 1) =
s.VariableArrayValue(s.SymbolNumber("FSC_Vector",
i, j))
  Next j
Next i

'Normal Seats
For i = 1 To 132
  For j = 1 To 2
    bn(i - 1, j - 1) =
s.VariableArrayValue(s.SymbolNumber("NormalSeats"
, i, j))
  Next j
Next i
```

A variável “FSC_Vector” contém todos os lugares correspondentes à primeira classe e a variável “NormalSeats” contém os de classe económica. Assim como acontece no modelo lógico, as variáveis que contêm os assentos disponíveis são matrizes de duas colunas: a primeira representa a fila do avião, e a segunda o lugar da fila. Por exemplo, na matriz, a fila que contém o lugar 5D apresenta na primeira coluna o valor 5, e na segunda o valor 4. Apesar destes conjuntos serem utilizados como variáveis para o ARENA, no VBA são considerados objetos, dificultando o seu processamento. Para facilitar este processo, as duas classes vão ficar contidas nas variáveis “bf” e “bn”, representando a primeira classe e a classe económica, respetivamente.

No caso dos assentos da primeira classe é necessário torná-los aleatórios. Para isto foi utilizado o excerto de código seguinte:

```
'swap array element i with a random element
For i = 0 To fsc - 1
    rnum = Int(Rnd() * (UBound(bf) - LBound(bf) +
1) + LBound(bf)) 'generate random number and
store as variable rnum
    rnum1 = bf(rnum, 1)
    temp = bf(i, 0) 'temporarily store array
element i to a variable called temp
    temp1 = bf(i, 1)
    bf(i, 0) = bf(rnum, 0) 'set array element i
equal to random array element rnum
    bf(i, 1) = rnum1
    bf(rnum, 0) = temp 'set array element rnum to
old value of array element i stored as temp
    bf(rnum, 1) = temp1
Next i
```

Este excerto permite trocar o elemento “i” do vetor com outro elemento aleatório. Como este processo é realizado para todos os elementos do vetor torna, desta maneira, a ordem dos passageiros completamente aleatória. A utilização desta função é bastante importante pois é utilizada várias vezes ao longo de todo o código.

2ª Etapa: Ordem dos passageiros da classe económica dependendo da estratégia de embarque selecionada.

No caso dos assentos de classe económica o processo é inverso porque os assentos já estão previamente ordenados e contidos na variável “NormalSeats”. A ordem dos passageiros

desta classe vai depender da estratégia escolhida pelo utilizador. Estas estratégias são denominadas: *random*, *back-to-front*, *outside-in*, *reverse pyramid*, *blocks*, *Steffen* e *modified optimal*.

Na estratégia *random* todos os passageiros da classe económica entram de forma aleatória no avião, originando um único grupo de embarque (Tabela 5). Nesse grupo todos os assentos são baralhados de forma aleatória, utilizando a função referida anteriormente. Após baralhado, o resultado da ordem dos passageiros é convertido na variável “NormalSeats”.

Tabela 5: Atribuição de assentos na estratégia *Random*

Grupos	Lugares
1	132 Lugares Filas 7 a 28 Lugares de A a F

Na estratégia *back-to-front* a classe económica está dividida em 5 grupos de embarque distinto onde o primeiro grupo a embarcar são os passageiros que se encontram no final do avião, e os últimos encontram-se no início, originando a seguinte divisão: grupo 1 inclui todos os assentos desde a fila 24 até ao final do avião; grupo 2 inclui os assentos desde a fila 19 até à 23; grupo 3 da fila 15 até à fila 18; grupo 4 da fila 11 à fila 15; grupo 5 da fila 7 à fila 10 (Tabela 6).

Tabela 6: Atribuição de assentos na estratégia *back-to-front*

Grupos	Lugares
1	30 Lugares Filas 24 a 28 Lugares de A a F
2	30 Lugares Filas 19 a 23 Lugares de A a F
3	24 Lugares Filas 15 a 18 Lugares de A a F
4	24 Lugares Filas 11 a 14 Lugares de A a F
5	24 Lugares

	<p style="text-align: center;">Filas 7 a 10 Lugares de A a F</p>
--	--

Após a divisão dos assentos em grupos é necessário baralhá-los dentro do próprio grupo de embarque. Para tal foi utilizada, mais uma vez, a função correspondente conhecida. Por fim, os grupos baralhados são colocados ordenadamente na variável “NormalSeats”, para que apenas sejam chamados os passageiros do segundo grupo, assim que termine o embarque de todos do primeiro grupo, e assim por diante, provocando, desta maneira, a divisão representada na Tabela 7.

Tabela 7: Divisão da Classe Económica na estratégia *back-to-ront*

Classe Económica – 132 Lugares					
Divisão:	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5
	30 Lugares	30 Lugares	24 Lugares	24 Lugares	24 Lugares

Já na estratégia *outside-in*, a classe é apenas dividida em 3 grupos de embarque distintos: grupo 1 inclui todos os assentos à janela do avião, ou seja, os assentos com o lugar A e F; grupo 2 inclui os lugares do meio, ou seja, os assentos B e E; grupo 3 representa os lugares do corredor, sobrando as letras C e D (Tabela 8).

Tabela 8: Atribuição de assentos na estratégia *outside-in*

Grupos	Lugares
1	44 Lugares Filas 7 a 28 Lugares A e F
2	44 Lugares Filas 7 a 28 Lugares B e E
3	44 Lugares Filas 7 a 28 Lugares C e D

Após a divisão dos assentos em grupos é necessário baralhá-los dentro do próprio grupo de embarque utilizando a função conhecida. Por fim, todos os grupos são colocados ordenadamente na variável “NormalSeats” originado a Tabela 9.

Tabela 9: Divisão da Classe Económica na estratégia *outsider-in*

Classe Económica – 132 Lugares			
Divisão:	Grupo 1 – Lugares junto à janela	Grupo 2 – Lugares do meio	Grupo 3 – Lugares junto ao corredor

A estratégia *reverse pyramid*, é uma mistura das estratégias *back-to-front* e *outside-in*, pois os primeiros passageiros a embarcar são os que se encontram nos lugares à janela do final do avião, e os últimos, são os que estão no lugar junto ao corredor na frente do avião. Para a aplicação desta estratégia, os passageiros são divididos em 5 grupos: o grupo 1 contém os passageiros à janela das filas 16 à 28; o grupo 2 apresenta os assentos à janela das filas 11 à 15 e os assentos dos meio das filas 21 à 28; o grupo 3 contém os assentos à janela das filas 7 à 10 e os assentos do meios das filas 12 à 20; o grupo 4 apresenta os assentos do meio das filas 7 à 11 e os assentos junto ao corredor das fila 21 à 28; e por fim, o grupo 5 contém os assentos junto ao corredor desde a fila 7 até à fila 20 (Tabela 10)

Tabela 10: Atribuição de assentos na estratégia *reverse pyramid*

Grupos	Lugares
1	26 Lugares Filas 16 a 28 Lugares A e F
2	26 Lugares Filas 11 a 15 - Lugares A e F Filas 21 a 28 – Lugares B e E
3	26 Lugares Filas 7 a 10 - Lugares A e F Filas 12 a 20 – Lugares B e E
4	26 Lugares Filas 7 a 11 - Lugares B e E Filas 21 a 28 – Lugares C e D
5	28 Lugares Filas 7 a 20

	Lugares C e D
--	---------------

Assim como acontece com as estratégias anteriores, é necessário baralhar os grupos de embarque e ordená-los na devida variável, originando a Tabela 11.

Tabela 11: Divisão da Classe Económica na estratégia *reverse pyramid*

Classe Económica – 132 Lugares					
Divisão:	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5
	26 Lugares	26 Lugares	26 Lugares	26 Lugares	28 Lugares

Por ser semelhante à estratégia *back-to-front*, a *blocks* utiliza o mesmo processo e os mesmos grupos de embarque que a anterior, trocando, apenas, a ordem de embarque dos grupos disponíveis, originando assim a Tabela 12. Como pode ser visualizado, o grupo 1 continua a apresentar os lugares do final do avião; o grupo 2 os passageiros do início do avião; o grupo 3 as filas 19 à 23; o grupo 4 as filas 11 à 14; e o grupo 5 as filas 15 à 18 (Tabela 12).

Tabela 12: Atribuição de assentos na estratégia *blocks*

Grupos	Lugares
1	30 Lugares Filas 24 a 28 Lugares de A a F
2	24 Lugares Filas 7 a 10 Lugares de A a F
3	30 Lugares Filas 19 a 23 Lugares de A a F
4	24 Lugares Filas 11 a 14 Lugares de A a F
5	24 Lugares Filas 15 a 18 Lugares de A a F

Assim como acontece com as estratégias anteriores, é necessário baralhar os grupos de embarque e ordená-los na devida variável, originando a Tabela 13.

Tabela 13: Divisão da Classe Económica na estratégia *blocks*

Classe Económica – 132 Lugares					
Divisão:	Grupo 1 30 Lugares	Grupo 2 24 Lugares	Grupo 3 30 Lugares	Grupo 4 24 Lugares	Grupo 5 24 Lugares

Já a estratégia de *Steffen*, não apresenta uma divisão em grupos. A divisão dos passageiros é individual e segue uma ordem de trás para a frente e das janelas para o corredor, deixando uma fila livre entre passageiros adjacentes. Esta estratégia preenche primeiro em cada lado da cabine, nas linhas pares e ímpares, os assentos da janela, de seguida os do meio e, por fim os do corredor. A Tabela 14 mostra a atribuição dos 20 primeiros assentos utilizando nesta estratégia. Como os lugares já estão ordenados de uma forma específica, são passados diretamente para a variável “NormalSeats”.

Tabela 14: Atribuição dos 20 primeiros assentos na estratégia *Steffen*

	1	2
1	28	6
2	26	6
3	24	6
4	22	6
5	20	6
6	18	6
7	16	6
8	14	6
9	12	6
10	10	6
11	8	6
12	28	1
13	26	1
14	24	1
15	22	1
16	20	1
17	18	1
18	16	1
19	14	1
20	12	1

Por fim, a estratégia *modified optimal* separa os passageiros da classe económica em 4 grupos distintos: o grupo 1 engloba os passageiros a estibordo (lugares A a C) das fila com número par; o grupo 2 apresenta os passageiros a bombordo (lugares D a F) das fila com número par; o grupo 3 engloba os passageiros a estibordo (lugares A a C) das fila com número ímpar; o grupo 4 apresenta os passageiros a bombordo (lugares D a F) das fila com número ímpar (Tabela 15).

Tabela 15: Atribuição de assentos na estratégia *modified optimal*

Grupos	Lugares
1	33 Lugares Filas nº par Lugares de A a C
2	33 Lugares Filas nº par Lugares de D a F
3	33 Lugares Filas nº ímpar Lugares de A a C
4	33 Lugares Filas nº ímpar Lugares de D a F

Assim como acontece com as estratégias anteriores, é necessário baralhar os grupos de embarque e ordená-los na devida variável, originando a Tabela 16.

Tabela 16: Divisão da Classe Económica na estratégia *modified optimal*

Classe Económica – 132 Lugares				
Divisão:	Grupo 1 33 Lugares	Grupo 2 33 Lugares	Grupo 3 33 Lugares	Grupo 4 33 Lugares

3ª Etapa: Separação dos passageiros prioritários da classe económica.

Uma das opções existentes na interface gráfica é a adição de passageiros prioritários onde é possível configurar o número de passageiros correspondente a esta classe. Caso esta classe

seja selecionada, os seus passageiros embarcam no avião depois da primeira classe e antes da classe económica, partilhando os mesmos assentos da última (Tabela 17).

Tabela 17: Divisão de classes de embarque com classe prioritária.

Assentos disponíveis		
Primeira Classe	Classe Prioritária	Classe Económica
24 Lugares Filas 1 a 6 Lugares de A a D	132 Lugares Filas 7 a 28 Lugares de A a F	

A classe prioritária é criada a partir da classe económica, pois ambas se encontram, de momento, no vetor “bn”. Para tal, são escolhidos elementos aleatórios deste vetor para representarem a classe prioritária. Esses elementos são guardados num vetor à parte denominado “vectorout”. Após guardado, é necessário eliminar estes elementos do vetor “bn”, mas como o visual basic não tem função própria, foi necessário deslocar todos os elementos, uma unidade, a partir do elemento selecionado, rescrevendo o vetor. No extrato de código seguinte encontra-se a separação dos passageiros da classe prioritária da classe económica:

```

For i = 0 To prioritynum - 1
    rnum = Int(Rnd() * (UBound(bn) -
LBound(bn) + 1 - i - (pmax - 24 - n)) +
LBound(bn)) 'generate random number and store as
variable rnum
    vectorout(i, 0) = bn(rnum, 0)
    vectorout(i, 1) = bn(rnum, 1)
    'save the random number in a different
vector
    For j = rnum To (n - 1)
        bn(j, 0) = bn(j + 1, 0)
        bn(j, 1) = bn(j + 1, 1)
        'This loop will shift array position one
place to the left. This way the number at your
position will be replaced by the next number.
    Next j
    n = n - 1 'redeem the size of the array
Next i

```

Após separados, ambos os vetores são colocados na variável “SeatArray” (variável lida pelo modelo lógico do ARENA) na ordem da Tabela 17.

4ª Etapa: Aplicação contra doenças infecciosas.

Assim como acontece com a etapa anterior, também é possível adicionar ao programa um modo de funcionamento, que aplica medidas contra doenças infecciosas como o COVID-19, através da interface gráfica. Neste modo, todos os assentos no meio do avião ficarão inutilizáveis, a menos que haja famílias com 3 elementos. Para o funcionamento deste modo, é necessário eliminar os elementos do vetor “bn” que correspondem a esses lugares. No entanto, como estes poderão ser usados pelas famílias, é necessário guardar a informação do mesmo. Para isto, foi utilizado a mesma função descrita na terceira etapa, mas desta vez os elementos removidos ficarão guardados no vetor “middlevector”. De seguida, os elementos deste vetor são tornados aleatórios, porque poderão estar ordenados por causa da estratégia selecionada. Por fim, um conjunto de elementos que corresponde ao número de famílias existentes é reintroduzido no vetor “bn”. O extrato de código seguinte mostra a reintrodução desses elementos:

```
For i = 0 To middlenuber - 1
    fg = 1
    For j = 0 To (n - 1)
        If bn(j, 0) = middlevector(i, 0) And fg =
1 Then
            If bn(j, 1) + 1 = middlevector(i, 1)
Or bn(j, 1) - 1 = middlevector(i, 1) Then
                fg = 0
                rnum = j
            End If
        End If
        'This loop will find the position a position
on the same row and side as the middle number
    Next j
    n = n + 1
    For j = rnum To (n - 1)
        bn(n - j + rnum, 0) = bn(n - j + rnum -
1, 0)
        bn(n - j + rnum, 1) = bn(n - j + rnum -
1, 1)
        'This loop will shift array position one
place to the right. This way the next number will
be replaced by this number.
    Next j
    bn(rnum, 0) = middlevector(i, 0)
    bn(rnum, 1) = middlevector(i, 1)
Next i
```

Cada elemento é reintroduzido numa posição específica. Essa posição será junto de um elemento do vetor “bn” que se encontra na mesma fila e ao lado do avião, ou seja, é um elemento do mesmo agregado familiar. Esta inserção permite que não existam interferências com a estratégia selecionada. Além do mais, é possível que o elemento seja uma criança, permitindo que esta seja acompanhada por um dos seus parentes. Assim como aconteceu com a remoção de elementos do vetor, não existe função para adicionar, por isso, a cada reintrodução é necessário deslocar todos os elementos, uma unidade, a partir do elemento inserido, rescrevendo e redimensionando o vetor.

5ª Etapa: Análise dos resultados obtidos.

O modelo lógico é capaz de guardar os resultados obtidos de uma replicação do modelo de simulação, mas não consegue comparar e analisar os resultados entre as várias replicações. Para resolver este problema, foram impostas um conjunto de condições. Como já foi visto, o modelo lógico guarda o valor do tempo total e individual de embarque. O visual basic recolhe estes valores, e no final de cada replicação do sistema são atualizadas a média, o máximo e mínimo do conjunto de replicações já executadas. Para calcular a média do tempo total e individual de embarque do conjunto de replicações foi utilizada a expressão descrita na Equação 2, onde “Xn” representa o valor guardado pelo modelo lógico durante a execução daquela replicação e “n” representa o número de replicações já analisadas. Logicamente, na seguinte replicação a “Média (atual)” vai passar a ser a “Média (anterior)”.

$$\text{Média (atual)} = (\text{Média (anterior)} * (n - 1) + X_n) / n \quad (\text{Equação 2})$$

Para atualizar o valor máximo e mínimo do tempo total e individual de embarque é realizada uma comparação entre os valores já analisados e o atual. O sistema apenas guarda os valores mais altos ou mais baixos obtidos até ao momento e sempre que uma nova replicação é analisada, este compara se os novos valores ultrapassam os limites (máximo ou mínimo) dos valores previamente guardados. Caso um valor seja excedido, o valor guardado correspondente será substituído por esse novo valor, tornando-se um novo limite. O extrato de código seguinte mostra os cálculos para a média, mínimo e máxima explicados anteriormente:

```

'Atualizar medias boarding, total / individual
UserForm3.Label5.Caption =
((UserForm3.Label5.Caption * var_rep) + var_boa)
/ UserForm3.Label12.Caption
UserForm3.Label6.Caption =
((UserForm3.Label6.Caption * var_rep) + var_ava)
/ UserForm3.Label12.Caption

'Atualizar boarding máximo
If var_boa > Val(UserForm3.Label16.Caption)
Then
    UserForm3.Label16.Caption = var_boa
End If

'Atualizar boarding mínimo
If var_boa < Val(UserForm3.Label15.Caption) Then
    UserForm3.Label15.Caption = var_boa
End If

```

No final de todas as replicações, o programa termina e os valores guardados são enviados para a interface gráfica onde estes serão exibidos.

4. ANÁLISE DE RESULTADOS

Neste capítulo são apresentados os resultados relativos ao trabalho desenvolvido e, posteriormente, a análise dos mesmos. Na secção 4.1 são apresentados os resultados a partir da animação do modelo de simulação e na secção 4.2 apresentam-se análise aos resultados obtidos das diferentes estratégias de embarque.

4.1. ANIMAÇÃO DO MODELO DE SIMULAÇÃO

Nesta secção apresentam-se os resultados obtidos através da animação do modelo de simulação para a configuração das diversas estratégias de embarque (onde são geradas atribuições específicas dependendo da estratégia).

Primeiro são mostrados vários cenários de simulação para o embarque de passageiros da primeira classe, pois a sua configuração é sempre idêntica, independentemente da estratégia seleccionada. A Figura 61 expõe os cenários correspondentes: (a) um passageiro entrou no avião; (b) praticamente metades dos passageiros da primeira classe já estão sentados; (c) a zona da primeira classe está totalmente sentada.

De seguida, são mostradas as capturas da animação das diferentes estratégias de embarque resultando nas seguintes imagens. Por exemplo, conforme ilustrado na Figura 62 para a

estratégia *back-to-front*, os quatro cenários de simulação descrevem-se: (a) os passageiros da primeira classe estão sentados; (b) os dois primeiros grupos estão sentados; (c) os quatro primeiros grupos estão sentados; (d) o último passageiro está prestes a sentar-se. As capturas da animação para as estratégias *random*, *outside-in*, *reverse pyramid*, *blocks*, *Steffen* e *modified optimal* são fornecidas desde a Figura 63 até à Figura 68 respetivamente, ilustrando quatro cenários de simulação (inicial, intermédio, quase final, final) para cada estratégia. Estas capturas foram realizadas com o avião na sua máxima capacidade, sem passageiros prioritários e com o modo de doenças infecciosas inativo, ou seja, com os parâmetros definidos da Figura 60. O estado dos recursos das capturas segue a descrição da Tabela 18.

Tabela 18: Estado dos Recursos

Estado dos recursos:	
Verde	Assento desocupado
Vermelho	Passageiro já chegou ao lugar, mas ainda não está sentado
Azul	Assento ocupado pelo passageiro

Parametrization

Passengers Arrival | Strategies | Boarding Parameters | Simulation Parameters

Passengers

Number of Passengers (max: 156): Enable Priority (default: no): Yes No

Percentage of Passengers with Hand Luggage (%): Number of Passengers with Priority (number):

Infectious Disease

This field represents the application of measures against diseases like Covid-19. This mode of operation does not occupy the middle seats of the airplane, except when there are families with at least 3 elements. In addition to occupy a place that was previously empty, the family is given the opportunity to travel together. This point is relevance when families travel with children.

On this mode the number of passengers will be adjusted.

Enable Method of Protection Against Diseases (default: no): Yes No

Number of Families with 3 members (number):

Cancel Ok

Figura 60: Parâmetros para a animação do modelo

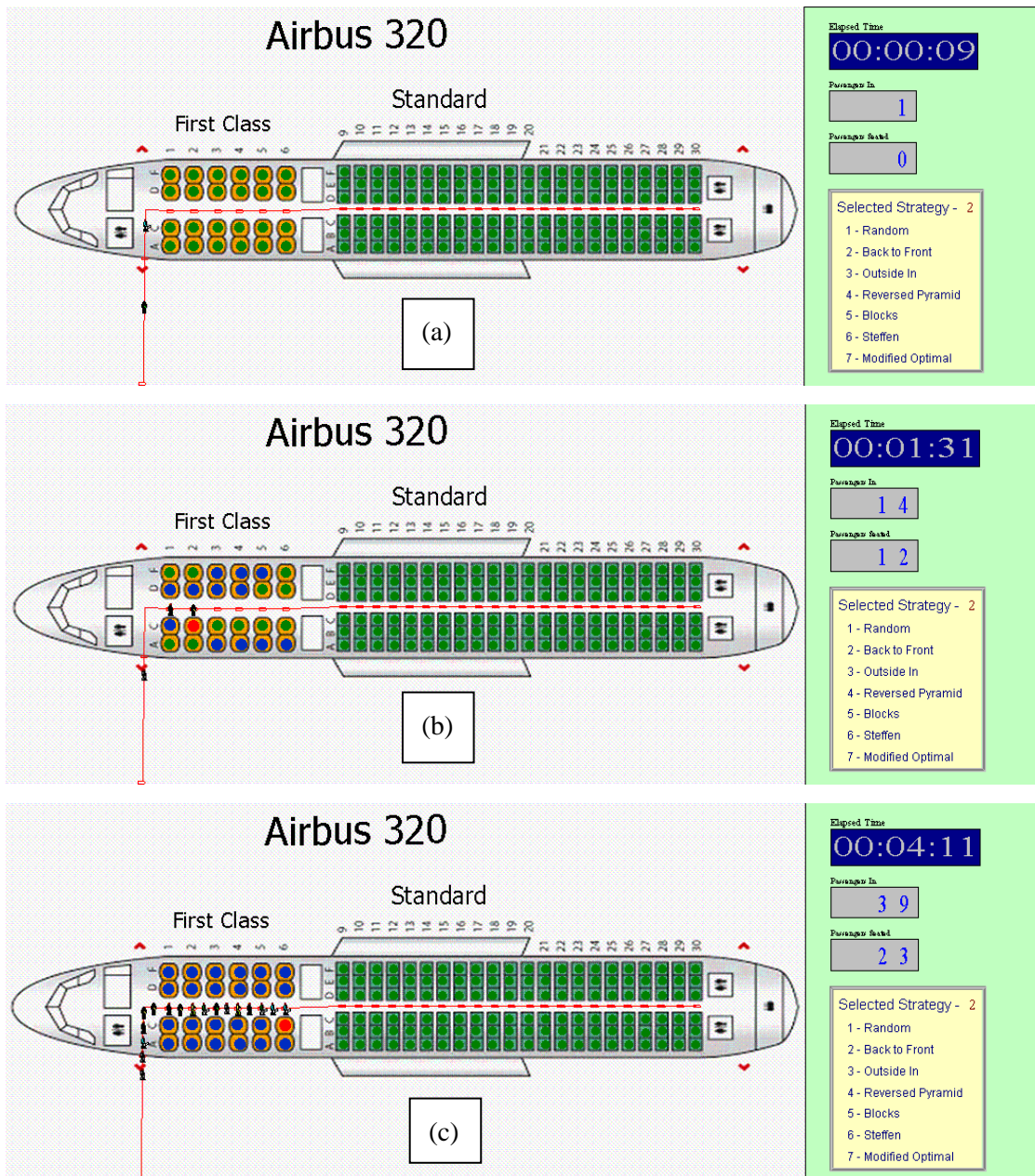


Figura 61: Cenários do embarque para os passageiros da primeira classe

Fonte imagem base do Airbus A320: (Seatplans.com, 2021)

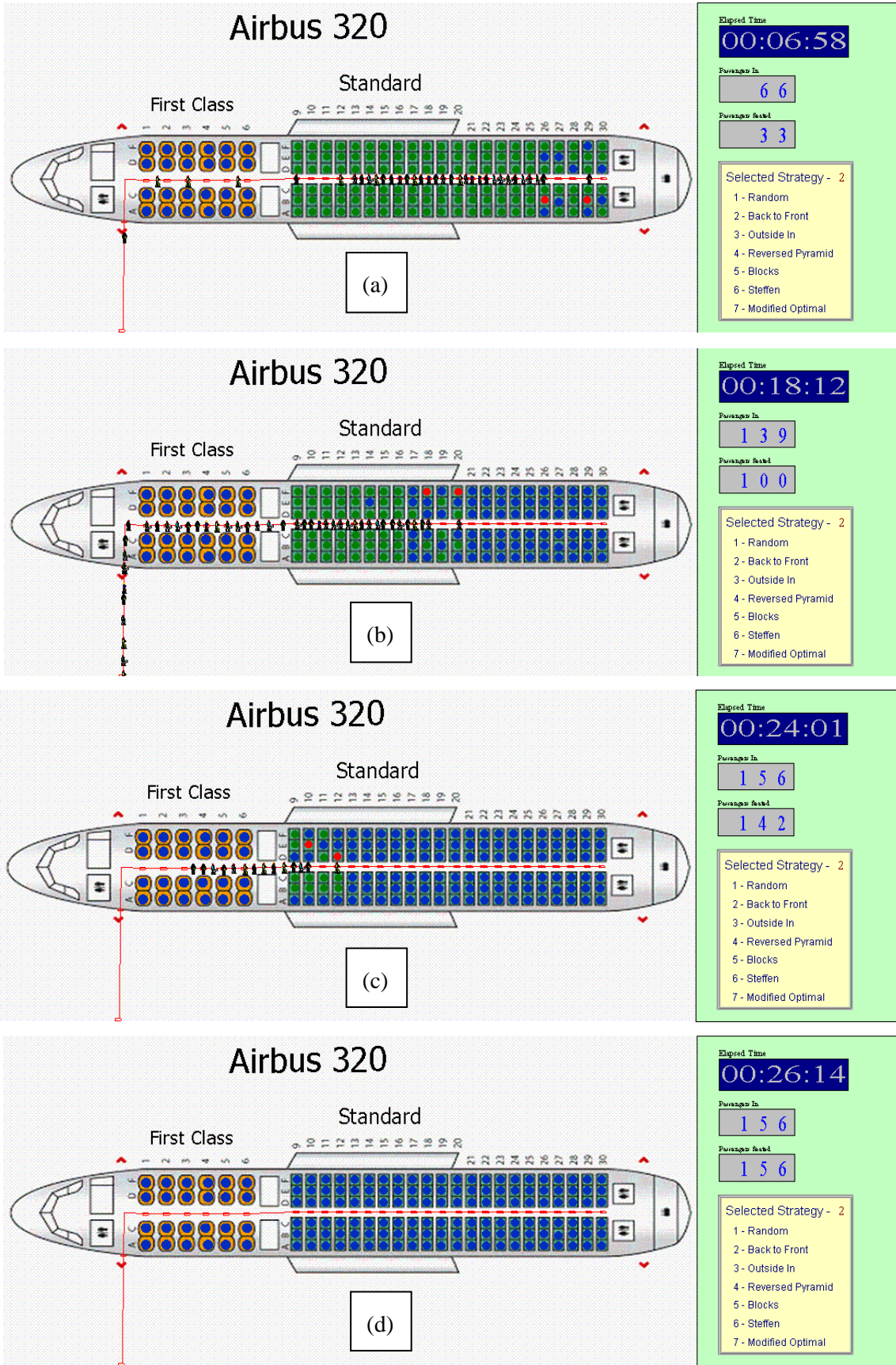


Figura 62: Estratégia *back-to-front* - Fonte imagem base do Airbus A320: (Seatplans.com, 2021)

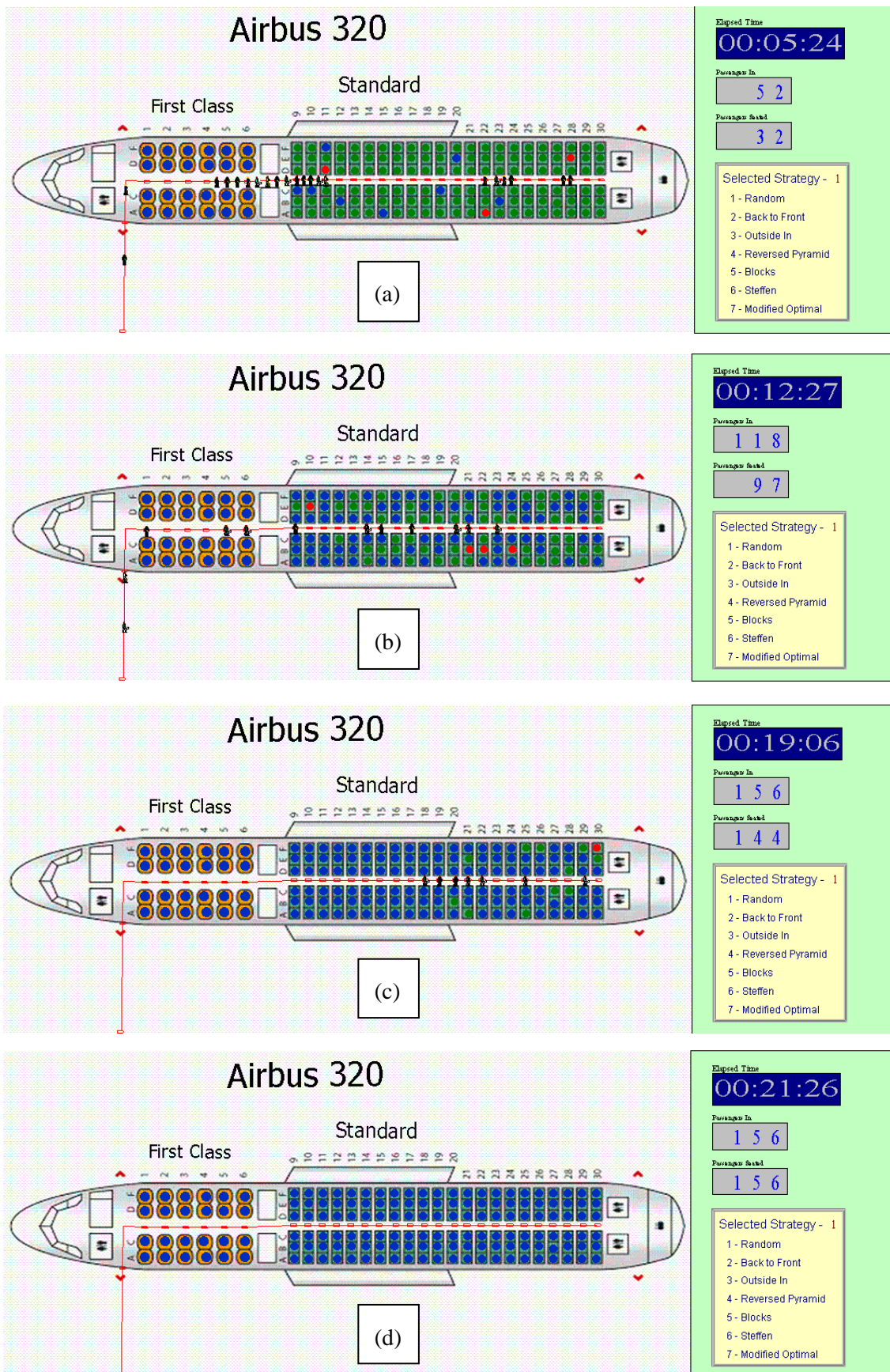


Figura 63: Estratégia *random* - Fonte imagem base do Airbus A320: (Seatplans.com, 2021)

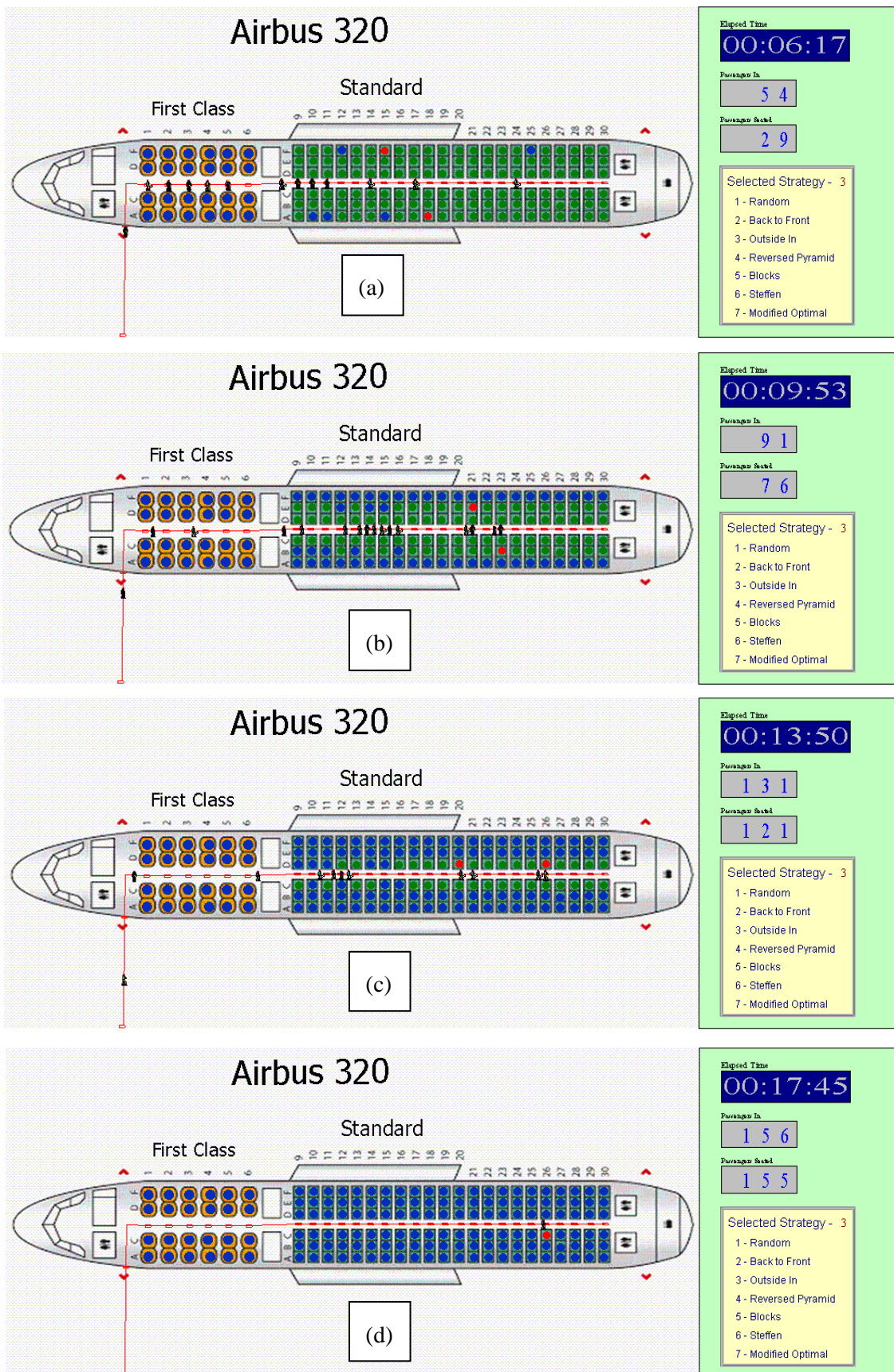


Figura 64: Estratégia *outside-in* - Fonte imagem base do Airbus A320: (Seatplans.com, 2021)

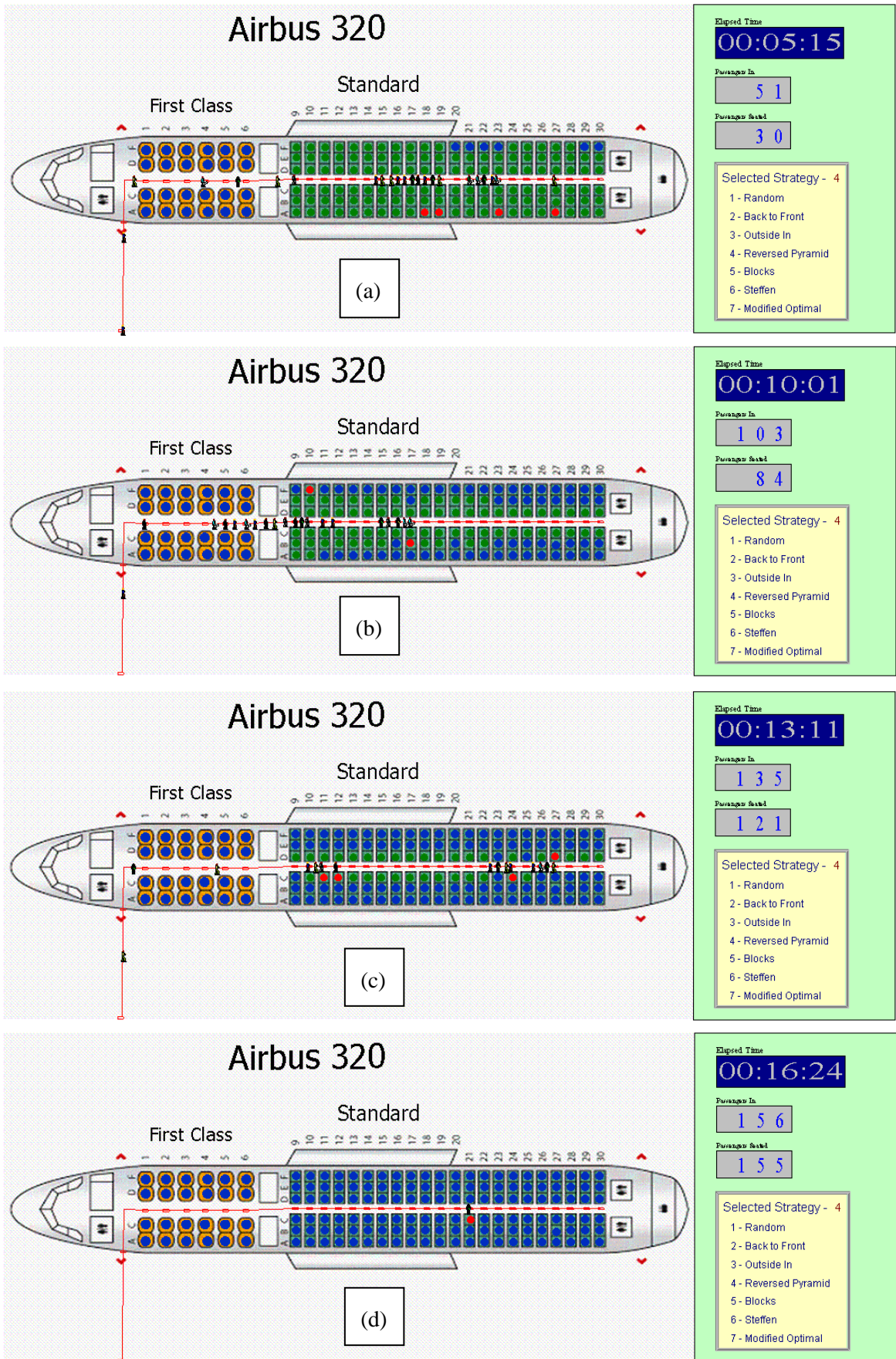


Figura 65: Estratégia *reverse pyramid* - Fonte imagem base do Airbus A320: (Seatplans.com, 2021)

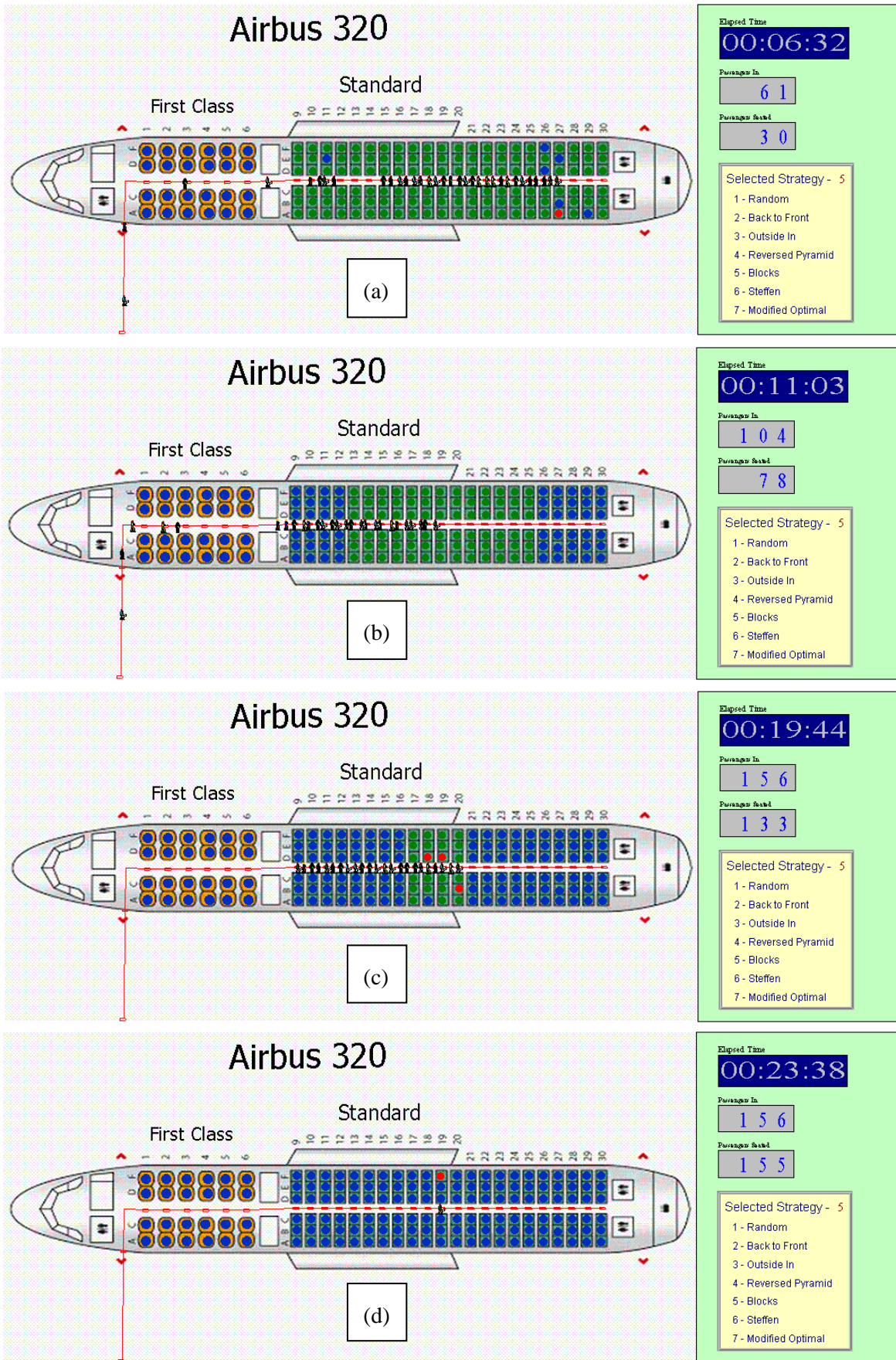


Figura 66: Estratégia *blocks* - - Fonte imagem base do Airbus A320: (Seatplans.com, 2021)

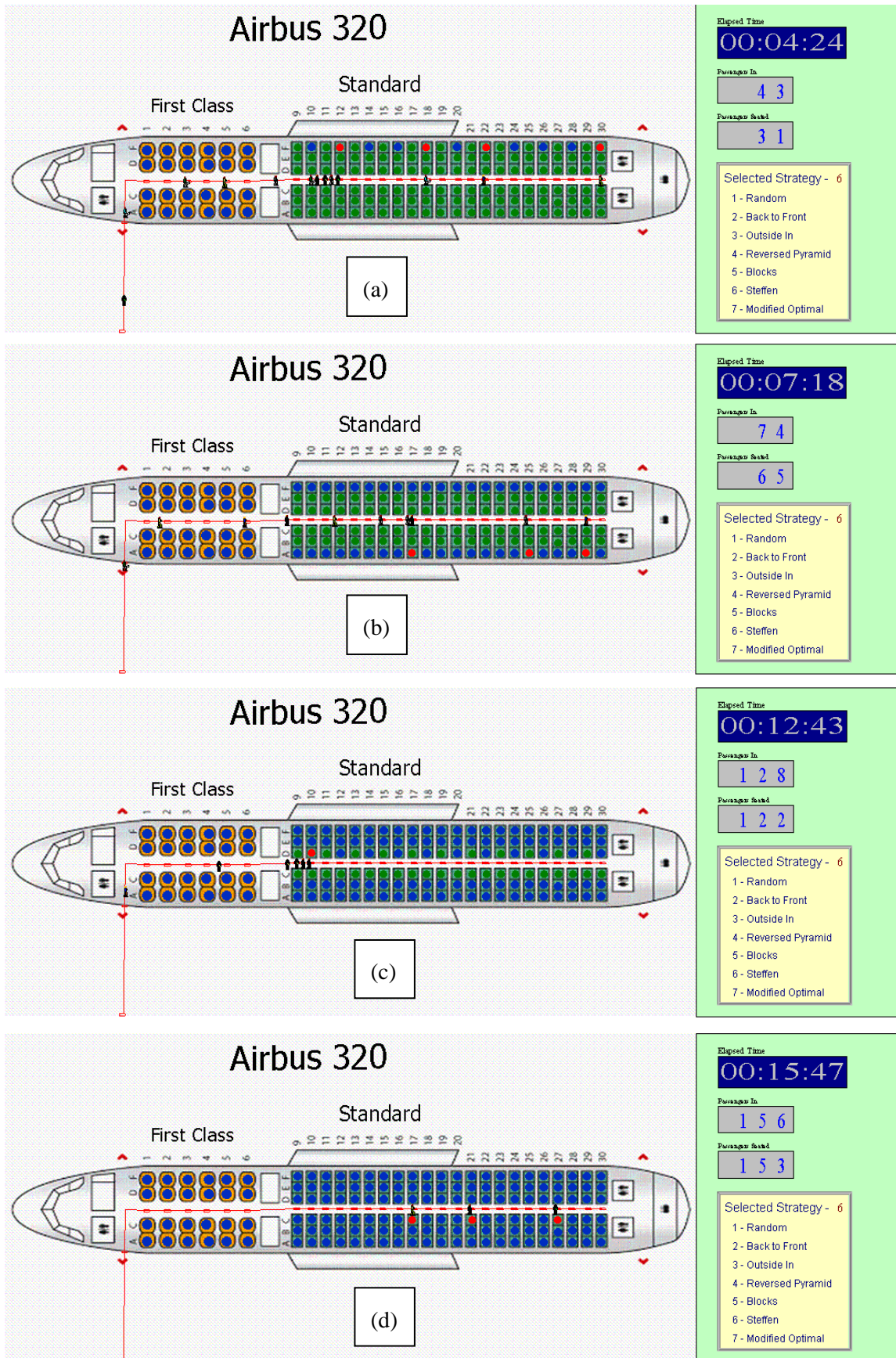


Figura 67: Estratégia Steffen - Fonte imagem base do Airbus A320: (Seatplans.com, 2021)

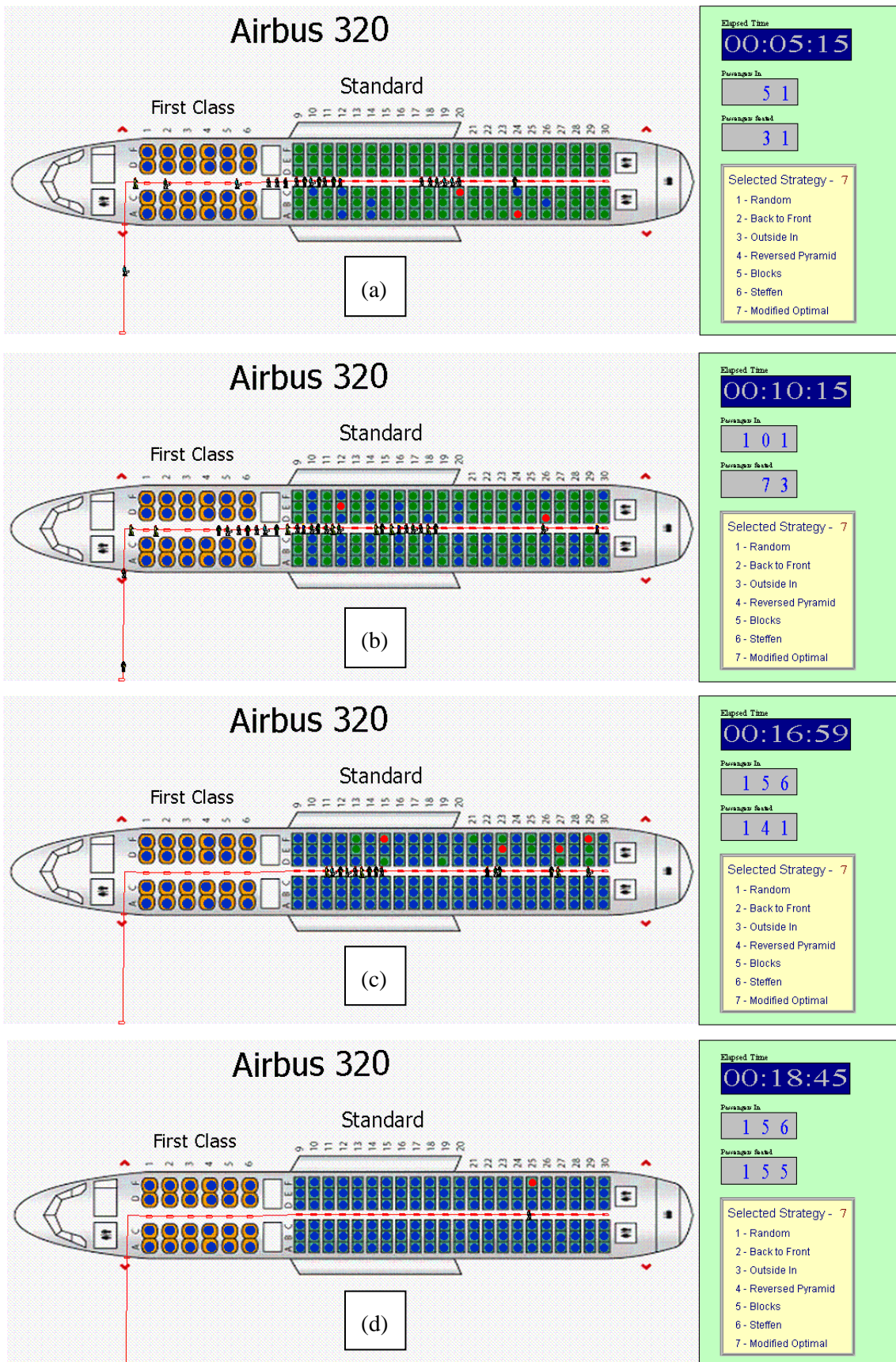


Figura 68: Estratégia modified optimal - Fonte imagem base do Airbus A320: (Seatplans.com, 2021)

Segue-se a simulação para o embarque de passageiros prioritários. Como já foi referido anteriormente, estes passageiros embarcam após a primeira classe. Para simular esta classe, utilizaram-se os parâmetros definidos na Figura 60, com a exceção da prioridade ativa e do campo correspondente ao número de passageiros prioritários, cujo valor é 25. Ou seja, estas capturas realizaram-se com 25 passageiros prioritários. Como é necessária a utilização de uma estratégia na simulação, foi escolhida a *back-to-front*. A Figura 69 expõe os vários cenários resultando: (a) os passageiros da primeira classe estão sentados; (b) praticamente metades dos passageiros prioritários já estão sentados; (c) os passageiros prioritários já estão sentados, dando início à aplicação da estratégia escolhida.

Por fim, é simulado o modo de doenças infecciosas. Para simular este modo, mais uma vez utilizaram-se os parâmetros definidos da Figura 60, com a exceção do modo de doenças infecciosas ativo, do campo correspondente ao número de famílias com 3 elementos com o valor 10. Como é necessária a utilização de uma estratégia na simulação, foi escolhida a *back-to-front*. A Figura 70 expõe os cenários correspondentes: (a) os passageiros da primeira classe estão sentados; (b) os dois primeiros grupos estão sentados; (c) os quatro primeiros grupos estão sentados; (d) o último passageiro está prestes a sentar-se.

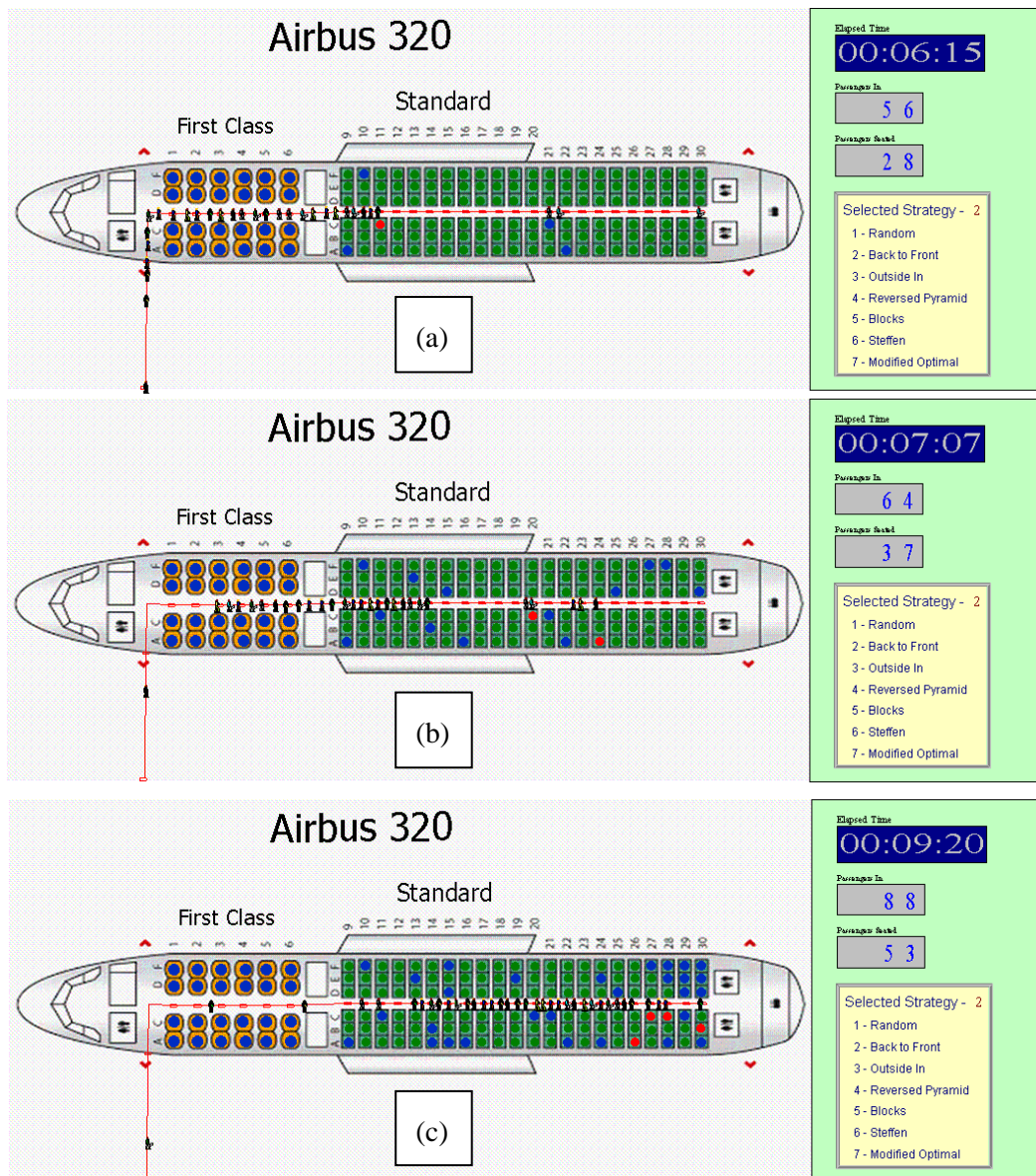


Figura 69: Possíveis cenários de embarque para os passageiros de classe prioritária

Fonte imagem base do Airbus A320: (Seatplans.com, 2021)

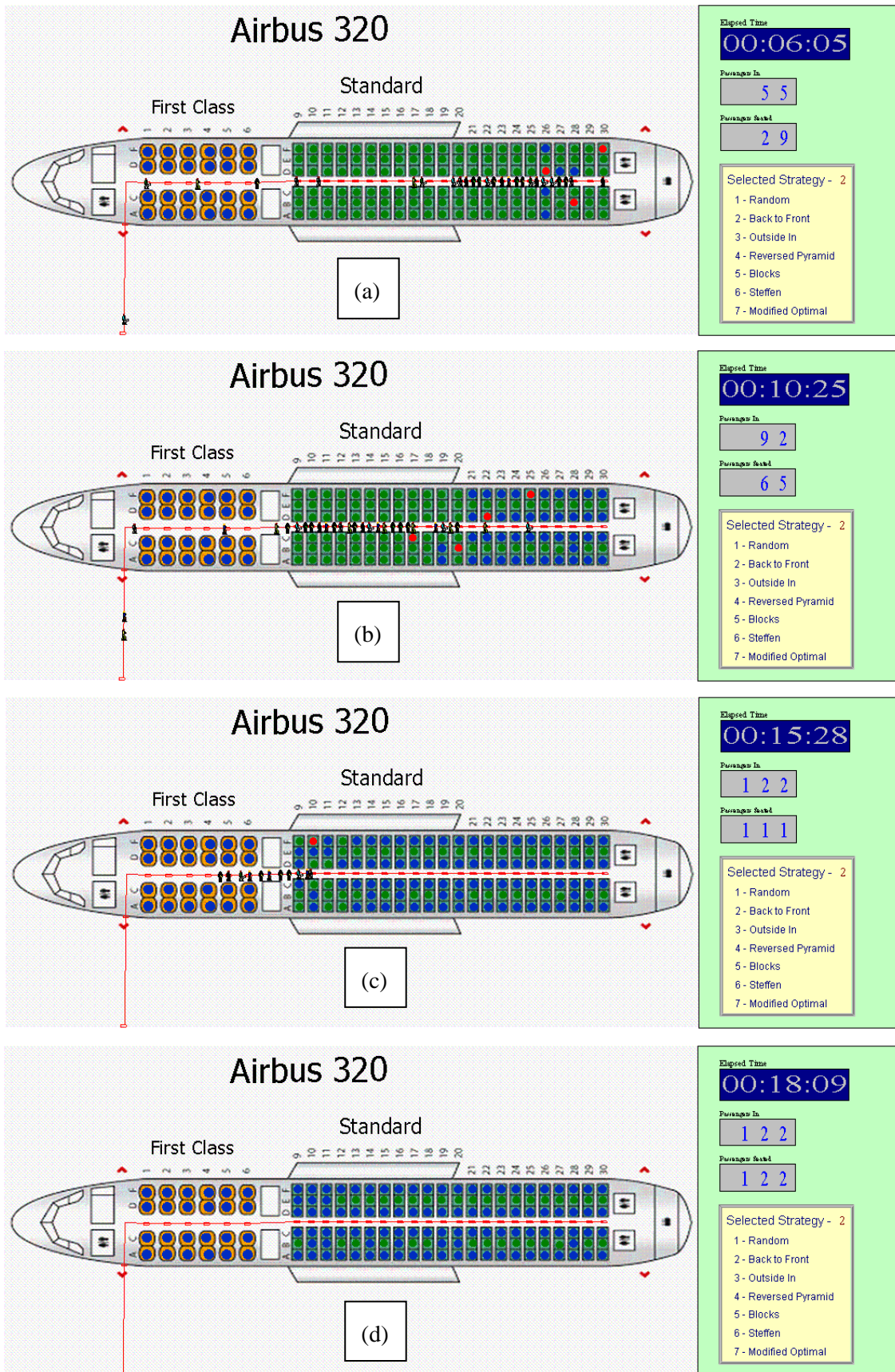


Figura 70: Possíveis cenários de embarque, modo de doenças infecciosas
 Fonte imagem base do Airbus A320: (Seatplans.com, 2021)

4.2. ANÁLISE DE RESULTADOS

Todos os resultados obtidos foram simulados com valores pré-definidos para os seguintes parâmetros:

- Taxa de entrada: Tempo entre a passagem de dois passageiros seguidos pela porta de embarque;
- Atraso na bagagem: tempo de atraso necessário para um passageiro guardar a sua bagagem de mão no devido compartimento;
- Atraso de assento tipo 1: tempo de atraso provocado pelas interferências de assento do tipo 1;
- Atraso de assento tipo 2: tempo de atraso provocado pelas interferências de assento do tipo 2;
- Atraso de assento tipo 3/4: tempo de atraso provocado pelas interferências de assento do tipo 3 e do tipo 4;

Na Tabela 19 estão descritos os vários parâmetros utilizados para a simulação do modelo com as suas respectivas distribuições temporais e valores.

Tabela 19: Valores dos vários atrasos

Parâmetro	Distribuição			Unidades	
	Tipo	Média	Desvio Padrão		
Taxa de entrada	Normal	6	3	s	
Atraso da bagagem de mão		20	10	s	
		Moda	Mínimo	Máximo	
Atraso de assento tipo 1	Triangular	22	20	26	s
Atraso de assento tipo 2		11	10	13	s
Atraso de assento tipo 3/4		10	9	12	s

4.2.1. IMPACTO DAS ESTRATÉGIAS DE EMBARQUE

Para comparar o desempenho das estratégias implementadas, foi analisado o tempo total de execução das simulações, ou seja, o tempo total de embarque. Cada replicação por si só não garante uma representação da amostragem, e para alcançar esse fim, replicou-se cada

estratégia 100 vezes, permitindo, desta maneira, obter o tempo médio, o desvio padrão, o máximo e o mínimo para cada estratégia. Na Figura 71 é possível comparar os tempos médios de embarque entre as diferentes estratégias de embarque. As simulações foram realizadas com o avião na sua máxima capacidade, e assumiu-se que a porcentagem de passageiros com bagagem de mão é de 75%. Para além disso, não existem passageiros prioritários nem o modo de doenças infecciosas nestas simulações, ou seja, a sua configuração está de acordo com os parâmetros definidos na Figura 60.

Tempo de embarque

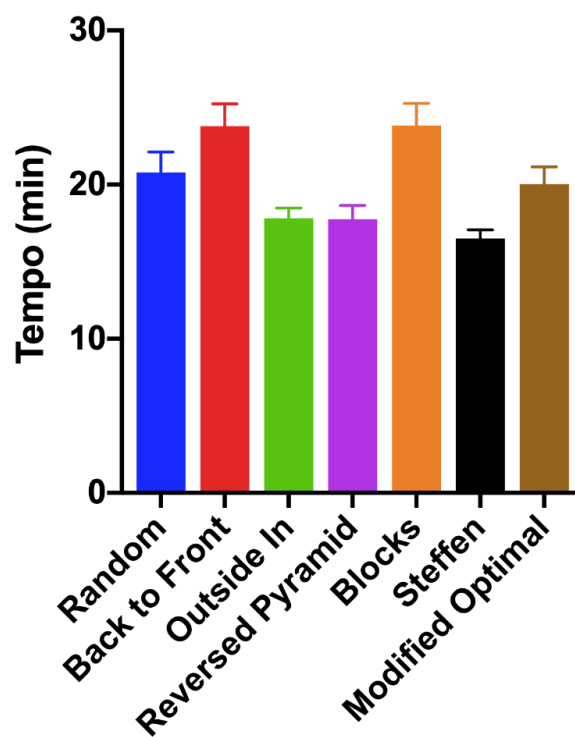


Figura 71: Tempo total de embarque

Conforme os resultados da Tabela 20, a estratégia de *Steffen* fornece um ponto de referência como o método de embarque “mais rápido”. No entanto, a estratégia *Steffen* não é prática, pois causará enormes filas na porta de embarque, tentando com que os passageiros entrem no avião um a um em ordem decrescente ao número de assento (com exceção dos passageiros da primeira classe). No entanto, as estratégias mais próximas a este ponto de referência, foram obtidas pelas estratégias *outside-in* e *reverse pyramid*. As estratégias de embarque *back-to-front* e *blocks* apresentam o pior desempenho. Já as estratégias *random* e *modified*

optimal apresentam um desempenho um pouco melhor, mas não tão bom como as estratégias *outside-in* e *reverse pyramid*, que a observação da Figura 71 e da Tabela 20 revela.

Tabela 20: Tempo de embarque obtido de acordo com as diferentes estratégias

Estratégia:	Tempo total de embarque (minutos)			
	Média	Desvio Padrão	Mínimo	Máximo
Random	20,72	1,32	18,2	24,05
Back-to-front	23,92	1,44	20,37	29,0
Outside-in	17,80	0,67	15,85	19,48
Reverse-Pyramid	17,53	0,90	15,9	19,57
Blocks	23,80	1,45	20,18	27,7
Steffen	16,40	0,56	14,94	17,77
Modified Optimal	20,05	1,12	17,73	22,97

A Tabela 21 mostra o tempo médio de embarque por passageiro. Esta é uma métrica importante que implica o nível de interferências de corredor e de assento que um passageiro deve passar antes de ocupar um assento. O tempo médio, como o próprio nome indica, refere-se ao tempo que um passageiro demora a sentar-se durante o embarque. O tempo máximo, refere-se ao maior tempo registrado num embarque por um passageiro durante as 100 replicações. A cadência de saída apresenta o intervalo médio de tempo que demoram dois passageiros a ocupar o seu respetivo lugar. Entre as diferentes estratégias, a maior diferença no tempo individual de embarque encontra-se nas estratégias que têm interferências de assento (*random*, *back-to-front*, *block* e *modified optimal*) e as que não têm (*outside-in*, *reverse pyramid* e *Steffen*). A diferença entre as estratégias *random* e *modified optimal* com as estratégias *back-to-front* e *blocks* deve-se ao nível de congestionamento de passageiros na mesma área do avião.

Tabela 21: Tempo individual de embarque obtido de acordo com as diferentes estratégias

Estratégia:	Tempo individual de embarque (minutos)				Cadência(s) (de saída)
	Média	Desvio Padrão	Máximo	Desvio Padrão	
Random	2,74	0,56	5,58	0,96	7,87
Back-to-front	3,55	0,44	7,62	0,93	9,08
Outside-in	1,90	0,38	3,97	0,60	6,75
Reverse-Pyramid	1,80	0,45	3,87	0,67	6,73
Blocks	3,57	0,42	7,59	0,82	9,08
Steffen	1,18	0,14	2,82	0,46	6,26
Modified Optimal	2,68	0,53	5,28	0,84	7,63

Na Figura 72 é apresentado um gráfico onde é refletida a sensibilidade de cada estratégia de embarque relativamente à percentagem de passageiros com bagagem de mão. Para qualquer uma das estratégias o gráfico mostra uma correlação positiva entre a percentagem de passageiros com uma mala e o tempo de embarque. Assim como aconteceu anteriormente, replicou-se cada estratégia, com a específica bagagem de mão, 100 vezes para obter um valor médio concreto. Como pode ser analisado, à medida que a percentagem de passageiros aumenta, o tempo total de embarque também aumenta, independentemente da estratégia implementada. Mas a taxa de aumento, pode variar com a estratégia escolhida. A estratégia de *Steffen* quase apresenta uma taxa de aumento nula devido à quase inexistência de interferências. Já as estratégias *outside-in* e *reverse pyramid* tendem a fornecer um desempenho muito mais próximo do ótimo. Isso é atribuído à eliminação das interferências de assento no embarque. Esta taxa apresenta natureza moderada nas estratégias *random* e *modified optimal*. As piores taxas de aumento registadas foram referenciadas nas estratégias *back-to-front* e *blocks*. Este desempenho deve-se ao aumento de interferências de corredor e de assento causadas pelo longo tempo de embarque.

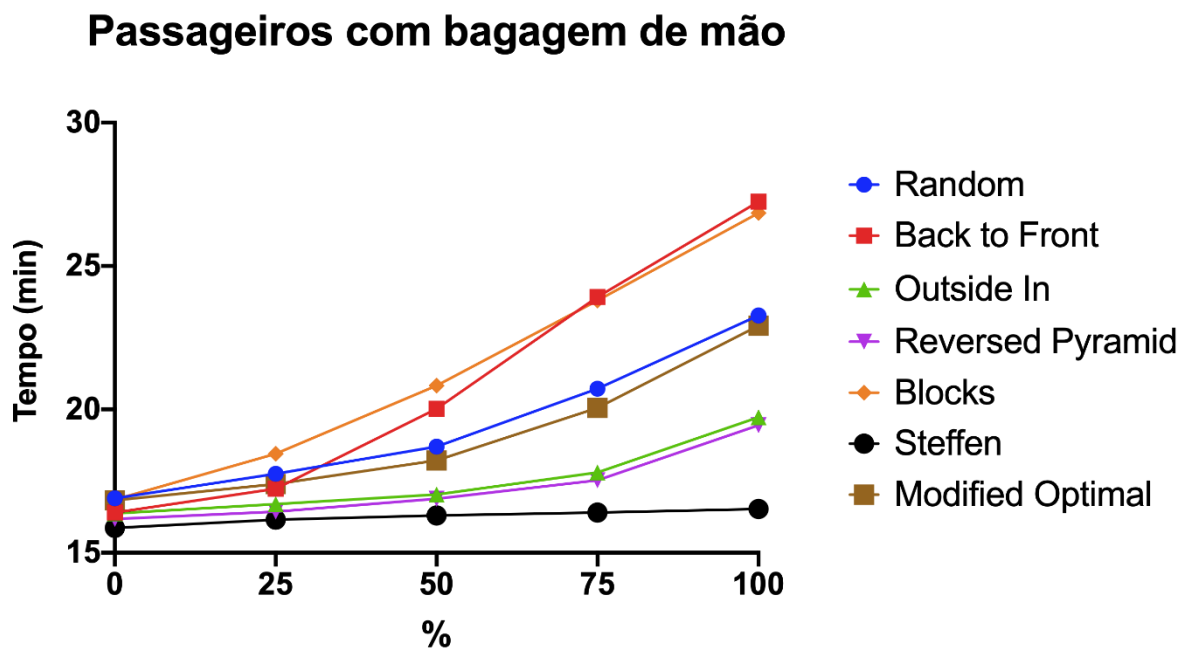


Figura 72: Tempo de embarque dependente do número de passageiros com bagagem de mão

A Tabela 22 faz uma análise dos efeitos da bagagem de mão, comparando o tempo de embarque em que todos ou nenhum dos passageiros apresenta bagagem de mão. Quando estes não possuem bagagem, praticamente não existem diferenças entre as diversas estratégias de embarque. Como pode ser visualizado na Figura 72, quando todos os passageiros apresentam bagagem de mão observa-se maior disparidade no tempo de embarque entre as diferentes estratégias.

Tabela 22: Comparação do tempo de embarque com e sem bagagem de mão

Estratégia:	Tempo de embarque (minutos)			
	Sem bagagem de mão		Com bagagem de mão	
	Média	Desvio Padrão	Média	Desvio Padrão
Random	16,89	0,74	23,26	1,29
Back-to-front	16,49	0,75	27,97	1,70
Outside-in	16,38	0,68	19,57	1,26
Reverse-Pyramid	16,18	0,63	19,43	1,18
Blocks	16,77	0,60	26,92	1,53
Steffen	15,95	0,56	16,41	0,68
Modified Optimal	16,87	0,65	22,75	1,24

A Tabela 23 compara o tempo individual de embarque dos passageiros quando estes têm ou não bagagem de mão. Quando estes não trazem bagagem não existem interferências no corredor. Como pode ser analisado, o tempo médio individual é bastante reduzido para qualquer uma das estratégias e apenas apresenta 2 tipos de resultados, sendo de 0,77 minutos para as estratégias com interferência de assentos e de 0,62 minutos para as estratégias sem interferência de assentos. O mesmo acontece para o valor máximo atingido por um passageiro durante as 100 replicações. Quando todos os passageiros apresentam bagagem de mão observa-se a maior diferença no tempo individual de embarque entre as diferentes estratégias. O melhor cenário foi obtido pela estratégia *reverse pyramid*, enquanto o pior cenário foi obtido pela *back-to-front*. Nos piores cenários, o tempo individual máximo atingido por um passageiro pode ser considerado bastante desconfortável.

Tabela 23: Comparação do tempo individual de embarque com e sem bagagem de mão

Estratégia:	Tempo individual de embarque (minutos)					
	Sem bagagem de mão			Com bagagem de mão		
	Média	Máximo	Cadência (s)	Média	Máximo	Cadência (s)
<i>Random</i>	0,77±0,04	1,89±0,23	6,47±0,29	3,54±0,39	6,82±0,71	8,84±0,50
<i>Back-to-front</i>	0,77±0,03	1,74±0,18	6,32±0,29	5,47±0,52	9,61±1,03	10,66±0,65
<i>Outside-in</i>	0,62±0,02	1,04±0,02	6,28±0,26	2,83±0,49	4,99±0,61	7,34±0,62
<i>Reverse-Pyramid</i>	0,62±0,004	1,06±0,06	6,22±0,26	2,79±0,51	5,32±0,86	7,33±0,46
<i>Blocks</i>	0,78±0,04	1,76±0,22	6,43±0,23	4,46±0,43	8,92±0,82	10,22±0,68
<i>Steffen</i>	0,62±0,005	1,11±0,09	6,12±0,22	1,42±0,25	3,26±0,46	6,18±0,29
<i>Modified Optimal</i>	0,77±0,04	1,82±0,18	6,40±0,50	3,58±0,28	6,61±0,53	8,64±0,48

Na Figura 73 é apresentado um gráfico onde é refletida a sensibilidade de cada estratégia de embarque relativa ao número de passageiros prioritários existentes no avião. Para qualquer uma das estratégias o gráfico mostra uma correlação positiva entre o número de passageiros prioritários existentes e o tempo de embarque. Para obter um valor médio, replicou-se 25 vezes cada estratégia, como número específico de passageiros prioritários. Este número de replicações não garante um tempo concreto, mas é o suficiente para determinar qual o tipo de alteração que a estratégia sofre. As estratégias *random* e *modified optimal* aparentam não sofrer alteração com a adição de passageiros prioritários. As estratégias *outside-in*, *reverse pyramid* e *Steffen* tendem a aumentar o tempo de embarque à medida que o número de passageiros prioritários aumenta. Isto acontece porque estas estratégias são desenhadas para eliminar as interferências de assento, mas com a aplicação de passageiros prioritários, essas interferências passam a existir. Já as estratégias *back-to-front* e *blocks* melhoram a sua performance à medida que o número de passageiros prioritários aumenta. Este desempenho deve-se à redução do congestionamento dos passageiros na mesma parte do avião. Apesar da melhoria do desempenho nestas estratégias, não é suficiente para ultrapassar o desempenho das *outside-in*, *reverse pyramid* e *Steffen*, mesmo quando a eficiência destas é reduzida e as *back-to-front* e *blocks* é aumentada. Caso o número de passageiro prioritários seja igual ao número de assentos disponível para a classe económica (132 lugares) qualquer uma das estratégias terá o mesmo tempo que a *random*, pois a sua programação é idêntica, ou seja, existe apenas um grupo de embarque.

Passageiros prioritários

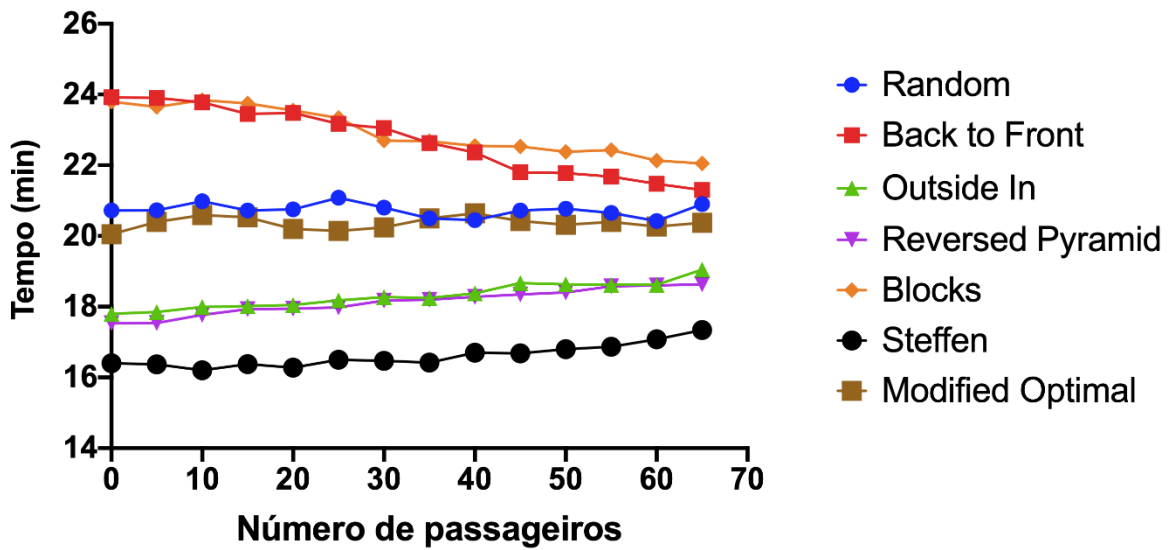


Figura 73: Tempo de embarque dependente do número de passageiros prioritários

4.2.2. IMPACTO DAS ESTRATÉGIAS DE EMBARQUE - MODO DE DOENÇAS INFECIOSAS

Nesta seção serão repetidas as mesmas experiências realizadas na seção 4.2.1., embora não estejam incluídos os agregados familiares, ou seja, eliminou-se por completo a utilização do assento do meio, ficando, desta maneira, o avião, com uma capacidade máxima de 112 passageiros. Apenas na última experiência será analisado o impacto que a inclusão dos agregados familiares tem nas diversas estratégias.

Na Figura 74 é comparado o desempenho das várias estratégias de embarque implementadas com o modo de doenças infecciosas ativo. Assim como aconteceu anteriormente, replicou-se cada estratégia 100 vezes para obter um valor médio concreto e utilizou-se uma percentagem de passageiros com bagagem de mão de 75%.

Tempo de embarque - COVID-19

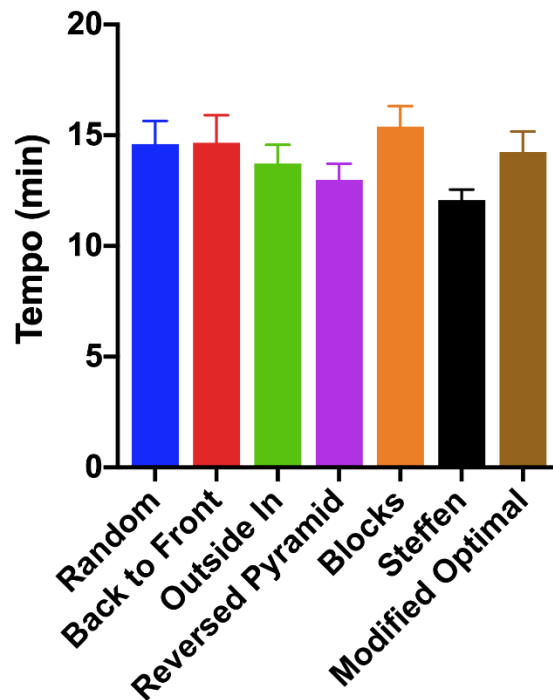


Figura 74: Tempo total de embarque, modo doenças infecciosas

Como pode ser analisado na Tabela 24, não existem grandes diferenças entre a aplicação das diferentes estratégias comparativamente com os resultados obtidos na Figura 71. Este resultado deve-se ao facto de o número de passageiros ser menor e de não existirem interferências de assento do tipo 1 (tipo que provoca o maior atraso). Apesar disso, ainda é possível distinguir dois grupos entre as estratégias de embarque. As estratégias *outside-in*, *reverse pyramid* e *Steffen* apresentam um melhor desempenho que as estratégias *random*, *back-to-front*, *blocks* e *modified optimal*. Isto deve-se ao facto de não existir qualquer tipo de interferências de assentos nas primeiras estratégias.

Tabela 24: Tempo total de embarque obtido, modo doenças infecciosas

Estratégia:	Tempo total de embarque (minutos)			
	Média	Desvio Padrão	Mínimo	Máximo
<i>Random</i>	14,85	1,05	12,55	17,57
<i>Back-to-front</i>	14,87	1,26	12,23	17,67
<i>Outside-in</i>	13,78	0,86	11,55	17,17
<i>Reverse-Pyramid</i>	13,05	0,74	11,58	15,17
<i>Blocks</i>	15,72	0,93	13,20	17,93
<i>Steffen</i>	12,00	0,48	10,72	13,35
<i>Modified Optimal</i>	14,33	0,94	11,95	16,67

Na Tabela 25 é visível que o nível de interferências de corredor e de assento diminuiu nas estratégias *back-to-front* e *blocks* com a utilização deste modo, pois existe uma aproximação do tempo médio de embarque por passageiro comparativamente a outras estratégias. Para além disso, o tempo máximo que um passageiro demorou a embarcar, nas 100 replicações, também diminuiu para as estratégias referidas e para as estratégias *random* e *modified optimal*. Neste modo de funcionamento, as estratégias *outside-in*, *reverse pyramid* e *Steffen* não sofreram qualquer alteração no tempo individual de embarque. Isto deve-se ao facto destas estratégias já anteriormente não apresentarem interferências de assentos do tipo 1.

Tabela 25: Tempo individual de embarque obtido, modo doenças infecciosas

Estratégia:	Tempo individual de embarque (minutos)				Cadência(s) (de saída)
	Média	Desvio Padrão	Máximo	Desvio Padrão	
<i>Random</i>	2,28	0,46	4,52	0,75	7,68
<i>Back-to-front</i>	2,64	0,53	4,74	0,80	7,71
<i>Outside-in</i>	2,02	0,39	3,84	0,59	7,24
<i>Reverse-Pyramid</i>	1,78	0,41	3,86	0,68	6,84
<i>Blocks</i>	2,43	0,42	4,91	0,92	8,14
<i>Steffen</i>	1,22	0,17	2,74	0,51	6,33
<i>Modified Optimal</i>	2,17	0,45	4,30	0,69	7,52

Assim como acontece na seção anterior, a Figura 75 apresenta um gráfico onde é refletida a sensibilidade de cada estratégia de embarque relativa à percentagem de passageiros com bagagem de mão, mas, neste caso, com o modo de doenças infecciosas ativo. Para qualquer uma das estratégias o gráfico mostra uma correlação positiva entre a percentagem de passageiros com uma mala e o tempo de embarque. Cada estratégia com a percentagem de bagagem de mão específica foi replicada 100 vezes. Como pode ser verificado, à medida que a percentagem de bagagem de mão aumenta, o tempo total de embarque também aumenta, independentemente da estratégia implementada. Mas a taxa de aumento, pode variar com a estratégia escolhida. A estratégia de *Steffen* apresenta uma taxa de aumento quase nula devido à quase inexistência de interferências. Das restantes estratégias a *reverse pyramid* apresenta uma menor taxa de aumento, enquanto a *blocks* apresenta a maior. É ainda de notar que a estratégia *back-to-front* obteve um bom desempenho nas amostras retiradas até à percentagem de passageiros com bagagem de mão igual a 50%, mas obteve um dos piores resultados quando a percentagem chegou aos 100%.

Passageiros com bagagem de mão

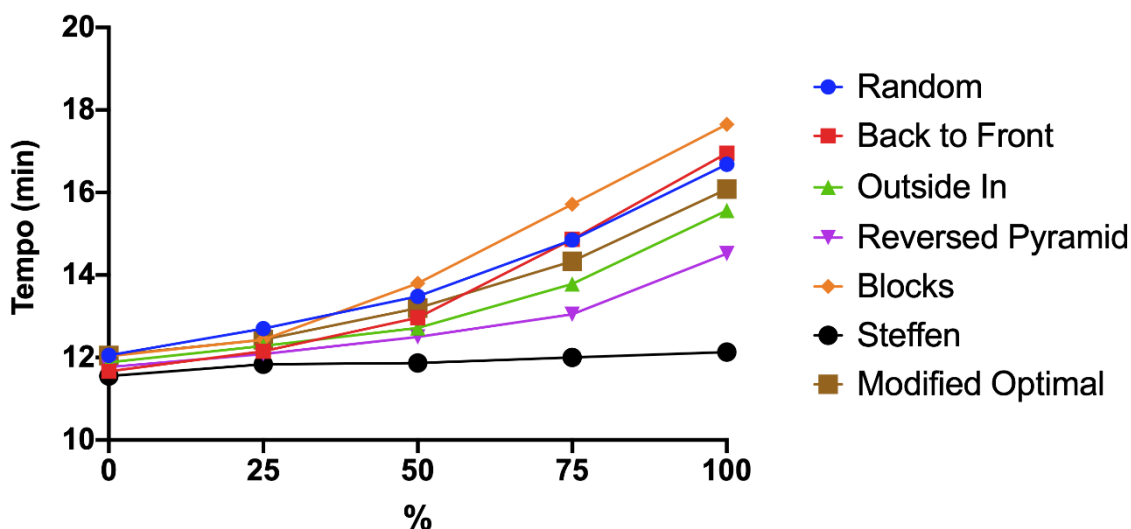


Figura 75: Tempo de embarque dependente do número de passageiros com bagagem de mão, modo doenças infecciosas

A Tabela 26 faz uma análise dos efeitos da bagagem de mão comparando o tempo de embarque quando todos ou nenhum dos passageiros possui bagagem de mão. Quando estes não levam bagagem, praticamente não existem diferenças entre as diversas estratégias de embarque. Quando todos os passageiros apresentam bagagem de mão, já existem diferenças significativas entre as diferentes estratégias.

Tabela 26: Comparação do tempo de embarque com e sem bagagem de mão, modo doenças infecciosas

Estratégia:	Tempo de embarque (minutos)			
	Sem bagagem de mão		Com bagagem de mão	
	Média	Desvio Padrão	Média	Desvio Padrão
Random	12,10	0,54	16,69	0,93
Back-to-front	11,62	0,49	16,95	1,29
Outside-in	11,97	0,50	15,57	1,03
Reverse-Pyramid	11,72	0,52	14,51	0,82
Blocks	11,93	0,59	17,67	0,98
Steffen	11,50	0,49	12,13	0,53
Modified Optimal	12,04	0,54	16,10	1,04

A Tabela 27 compara o tempo individual de embarque dos passageiros quando estes têm ou não bagagem de mão. Quando estes não levam bagagem, não existem interferências no corredor. Como pode ser analisado, o tempo médio individual é bastante reduzido para

qualquer uma das estratégias e não existem grandes diferenças entre estas. Apenas as estratégias *reverse pyramid* e *Steffen* se destacam com tempo individual de embarque reduzido. Quando todos os passageiros apresentam bagagem de mão observa-se a maior diferença no tempo individual de embarque entre as diferentes estratégias. Excluindo a estratégia *Steffen*, o melhor cenário para o tempo individual de embarque foi obtido pela estratégia *reverse pyramid* enquanto o pior cenário foi obtido pela *back-to-front*. Apesar disso, a estratégia *blocks* obteve a maior cadência de embarque, pois é a estratégia que atinge o maior tempo máximo individual de embarque.

Tabela 27: Comparação do tempo individual de embarque com e sem bagagem de mão, modo doenças infecciosas

Estratégia:	Tempo individual de embarque (minutos)					
	Sem bagagem de mão			Com bagagem de mão		
	Média	Máximo	Cadência (s)	Média	Máximo	Cadência (s)
Random	0,64±0,01	1,33±0,11	6,48±0,32	3,01±0,32	5,55±0,67	8,75±0,58
Back-to-front	0,64±0,01	1,26±0,07	6,20±0,32	3,57±0,58	6,16±1,03	8,91±0,70
Outside-in	0,64±0,01	1,26±0,07	6,40±0,26	2,77±0,40	5,03±0,61	8,16±0,55
Reverse-Pyramid	0,59±0,006	1,07±0,06	6,25±0,28	2,58±0,37	4,99±0,71	7,58±0,44
Blocks	0,64±0,01	1,27±0,09	6,37±0,32	3,24±0,36	6,72±0,98	9,29±0,54
Steffen	0,59±0,008	1,13±0,11	6,15±0,29	1,52±0,24	3,18±0,44	6,32±0,28
Modified Optimal	0,63±0,01	1,29±0,07	6,43±0,28	2,87±0,41	5,43±0,75	8,45±0,57

Assim como anteriormente, a Figura 76 apresentada um gráfico onde é refletida a sensibilidade de cada estratégia de embarque relativa ao número de passageiros prioritários existentes no avião, mas, neste caso, com o modo de doenças infecciosas ativo. Para qualquer uma das estratégias o gráfico mostra uma correlação positiva entre o número de passageiros prioritários existentes e o tempo de embarque. As estratégias *random* e *modified optimal* aparentam não sofrer nenhuma alteração com a adição de passageiros prioritários. As estratégias *outside-in* e *reverse pyramid* tendem a aumentar ligeiramente o tempo de embarque à medida que o número de passageiros prioritários aumenta, enquanto a estratégia *Steffen* tem um aumento significativo. Isto acontece porque estas estratégias são desenhadas para eliminar as interferências de assento, mas com a aplicação de passageiros prioritários, essas interferências passam a existir. Já a estratégia *blocks* melhora ligeiramente a sua performance à medida que o número de passageiros prioritários aumenta enquanto a

estratégia *back-to-front* tem uma diminuição significativa. Este desempenho deve-se à redução do congestionamento dos passageiros na mesma parte do avião.

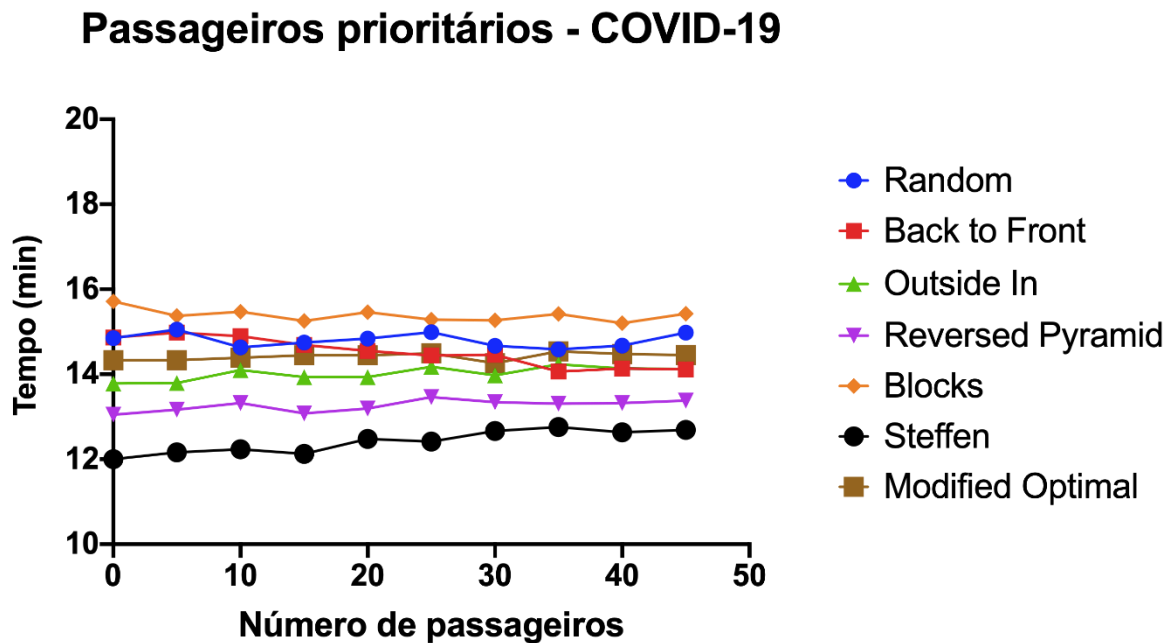


Figura 76: Tempo de embarque dependente do número de passageiros prioritários, modo doenças infecciosas

Na Figura 77 é apresentado um gráfico onde é analisada a sensibilidade de cada estratégia de embarque relativa ao número de famílias existentes no avião durante uma situação de pandemia. Para qualquer uma das estratégias o gráfico mostra uma correlação positiva entre o número de famílias existentes e o tempo individual de embarque. Para obter um valor médio, replicou-se 25 vezes cada estratégia, com o número específico de passageiros prioritários. Este número de replicações não garante um tempo concreto, mas é o suficiente para determinar qual o tipo de alteração que a estratégia sofre. A estratégia *Steffen* é a única que não sofrer qualquer alteração no tempo individual de embarque com a adição de passageiros prioritários. Todas as outras estratégias tendem a aumentar o tempo individual de embarque à medida que o número de famílias aumenta. Isto acontece porque com o aumento do número de famílias, o número de passageiros também aumenta, o que por sua vez faz incrementar o número de interferências existentes. É de frisar ainda que, a redução de interferências numa situação de pandemia é extremamente importante para reduzir o risco de contágio entre os passageiros.

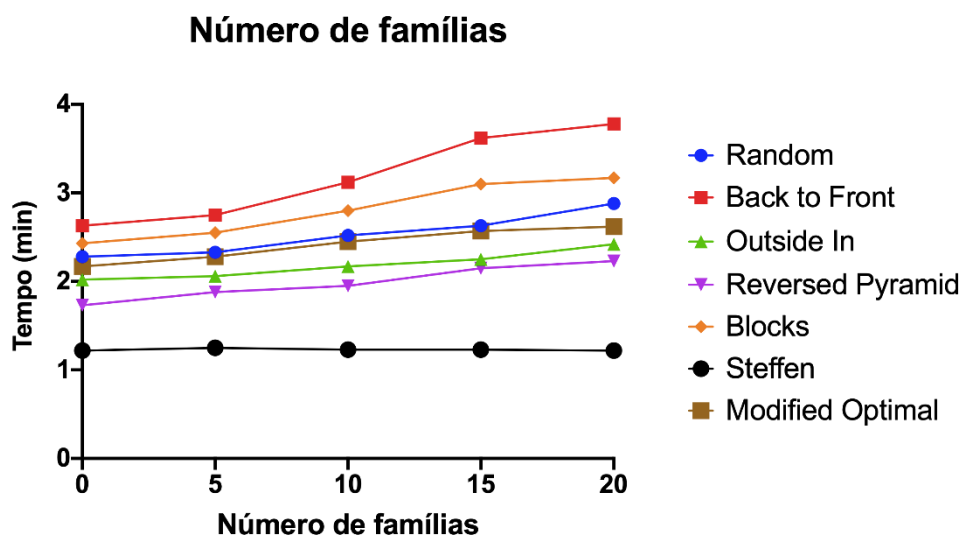


Figura 77: Tempo individual de embarque, dependendo do número de famílias

4.3. VALIDAÇÃO DO SISTEMA DESENVOLVIDO

Após o desenvolvimento do modelo de simulação, existe a necessidade de realizar um estudo para o validar. Esta fase mostra-se de crucial importância para o projeto, uma vez que define se o modelo criado é representativo do modelo real ou não. Como não foi possível ter acesso a dados reais para o embarque de passageiros, foi realizada uma comparação dos resultados obtidos com estudos semelhantes existentes na literatura. A Tabela 28 resume as melhores práticas de estratégias de embarque na literatura:

Tabela 28: Comparação de resultados existentes na literatura

Autor	Melhor estratégia
(Marelli et al., 1998)	<i>Outside-in</i>
(Van Landeghem & Beuselinck, 2002)	<i>Outside-in</i>
(Steffen & Hotchkiss, 2012)	<i>Outside-in</i>
(Soolaki et al., 2012)	<i>Reverse Pyramid</i>
(Kalic et al., 2013)	<i>Outside-in</i>
(Jafer & Mi, 2017)	<i>Outside-in</i>
(Ren & Xu, 2018)	<i>Reverse Pyramid</i>

Ao analisar a Tabela 28, conclui-se que as estratégias de embarque mais eficientes que utilizam zonas de embarque são a *outside-in* e *reverse pyramid*. Ou seja, as mesmas estratégias que obtiveram os melhores resultados no modelo de simulação desenvolvido.

Deste modo, conclui-se que os modelos de simulação desenvolvidos são válidos, correspondendo a uma representação correta do processo de embarque em aviões, havendo, assim, a possibilidade de ser usado como ferramenta de apoio.

5. CONCLUSÕES E PROPOSTAS DE TRABALHO FUTURO

5.1. CONCLUSÕES

Este projeto possibilitou o estudo e a análise da eficiência das estratégias de embarque de passageiros num Airbus A320. Isto foi feito com base na simulação discreta utilizando o *software* ARENA. Através das simulações de várias situações de embarque foram avaliadas sete estratégias: *random*, *back-to-front*, *outside-in*, *reverse pyramid*, *blocks*, *Steffen* e *modified optimal*.

Existiram algumas dificuldades na implementação de certas funcionalidades do modelo, especialmente no que concerne às regras que travavam a movimentação dos passageiros no embarque originadas por uma interferência específica, e na atribuição dos assentos às devidas entidades. Estas dificuldades exigiram um estudo cuidadoso da linguagem de programação utilizada pelo ARENA para corresponder à definição teórica dos diversos conceitos.

Em relação aos resultados obtidos pode-se concluir que:

- A estratégia de *Steffen* apesar de obter excelentes resultados na maioria das situações, a sua aplicação não é prática, pois causará enormes filas na porta de embarque, tentando com que os passageiros entrem no avião um a um na ordem pretendida.
- Das sete estratégias de embarque implementadas, a *outside-in* e a *reverse pyramid* proporcionaram o menor tempo de embarque, enquanto as estratégias *back-to-front* e *blocks* consagraram-se as mais ineficientes. A estratégia *reverse pyramid* obteve uma melhoria de 18% em relação à estratégia *random*, assumindo que 75% dos passageiros possui bagagem de mão. Comparativamente à estratégia *back-to-front* a *reverse pyramid* obteve uma melhoria de 36% nas mesmas condições. Por esta razão, a estratégia *back-to-front*, usada por várias companhias aéreas, é ineficiente;
- Quando praticamente todos os passageiros possuem bagagem de mão, as estratégias de embarque mais eficazes são a *reverse pyramid* e a *outside-in*. Estas permitem à companhia aérea uma melhoria no tempo de embarque de até 20% comparativamente à estratégia *random*. Comparando o melhor cenário, *reverse pyramid*, com o pior, *back-to-front*, existe uma redução de 40% no tempo de embarque;
- Os dados mostram que as simulações de estratégias de embarque realizadas não se justificam para modelos onde a totalidade dos passageiros não apresentem bagagem de mão ou são maioritariamente pertencentes à classe prioritária.
- A implementação do modo de doenças infecciosas foi realizada com sucesso, assim como a adição de agregados familiares. Das sete estratégias de embarque implementadas, a *reverse pyramid* proporcionou o menor tempo de embarque, enquanto a estratégia *blocks* consagrou-se o método mais ineficiente. A estratégia *reverse pyramid* obteve uma melhoria de 14% comparativamente com a estratégia *random*, assumindo que 75% dos passageiros possui bagagem de mão. Comparativamente à estratégia *blocks*, a *reverse pyramid* obteve uma melhoria de 21% nas mesmas condições. No caso de todos os passageiros possuírem bagagem de mão, a estratégia *reverse pyramid* obteve uma melhoria de 15% comparativamente com a *random* e 22% com a estratégia *blocks*.

5.2. CARACTER INOVADOR DO TRABALHO

Esta investigação apresenta alguns aspetos de carácter inovador. Trata-se do primeiro estudo em Portugal centrado em estratégias de embarque da aviação, encontrando-se, também, ajustado à realidade aeroportuária portuguesa. Paralelamente, foi concebido um modo de funcionamento – modo de doenças infecciosas – que reconverte as estratégias de embarque seleccionadas neste trabalho. O objetivo deste modo é encaixar as estratégias nas recomendações estabelecidas pela IATA, especialmente a medida que permite manter o assento do meio desocupado para reduzir o risco de contaminação por COVID-19 durante o processo de embarque (IATA, 2020). Para além disso, foi incluída uma funcionalidade complementar às recomendações da IATA, que permite ocupar o assento do meio, caso exista uma família com 3 ou mais elementos. Esta funcionalidade traz vantagens para a companhia aérea e para as famílias. No primeiro caso, o lugar que antes se encontrava vago, encontra-se agora ocupado, permitindo aumentar a lotação do avião e compensar perdas económicas referentes aos lugares desocupados. No segundo caso, permite à família embarcar e viajar (parcialmente) junta, mesmo que na estratégia escolhida se encontre inicialmente separada em diferentes grupos. Este ponto tem grande relevância quando as famílias viajam com crianças.

5.3. PROPOSTAS DE TRABALHO FUTURO

Sabendo que este trabalho pode ser estendido em diversos domínios, deixam-se aqui algumas propostas complementares:

- Testar os modelos COVID-19 e família em ambiente real.
- Testar o impacto que os *late passengers* teriam sobre as estratégias de embarque, criando condições mais fiéis ao sistema real, uma vez que os dados utilizados no modelo apenas representam uma situação ideal de funcionamento.
- Simulação de diferentes velocidades de passageiros, correspondentes às entidades.
- Testar o impacto que um número controlado de passageiros da primeira classe teria sobre as diversas estratégias de embarque.

- Adicionar distanciamento social entre os passageiros no corredor do avião quando o modo de doenças infecciosas estiver ativo.

Referências Bibliográficas

- Ares, E., Ferreira, L. P., Pelaez, G., Prieto, D., & Chao, A. (2012). Optimisation of a Production Line using Simulation and Lean Techniques. In E. B. Tjahjono, C. Heavey, S. Onggo, and D-J. van der Zee (Ed.), *Proceedings of the Operational Research Society Simulation Workshop* (pp. 177–183). Worcestershire, United Kingdom.
- Barnett, A., & Fleming, K. (2020). Covid-19 Risk Among Airline Passengers: Should the Middle Seat Stay Empty? <https://doi.org/10.1101/2020.07.02.20143826>
- Barton, R. R. (2020). Tutorial: Metamodeling for Simulation. In E. K.-H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing (Ed.), *Proceedings of the 2020 Winter Simulation Conference* (pp. 1102–1116). Online: IEEE. <https://doi.org/10.1109/WSC48552.2020.9384059>
- Bhandari, P. (2020). An introduction to quantitative research. Retrieved February 20, 2021, from <https://www.scribbr.com/methodology/quantitative-research/>
- Bidanda, R., Winakor, J., Geng, Z., & Vidic, N. (2017). A review of optimization models for boarding a commercial airplane. In *Proceedings of the 24th International Conference on Production Research* (pp. 1–6). Poznan, Poland.
- Chapuis, B., Calmon, P., & Jenson, F. (2018). Advantages and Limitations of the Simulation. In Springer (Ed.), *Best Practices for the Use of Simulation in POD Curves Estimation* (pp. 9–16).
- Cotfas, L. A., Delcea, C., Milne, R. J., & Salari, M. (2020). Evaluating Classical Airplane Boarding Methods Considering COVID-19 Flying Restrictions. *Symmetry*, 20(7), 1087. <https://doi.org/10.3390/sym12071087>
- Delcea, C., Cotfas, L.-A., Chirita, N., & Nica, I. (2018). A Two-Door Airplane Boarding Approach When Using Apron Buses. *Sustainability*, 10(10), 3619. <https://doi.org/10.3390/su10103619>

- Delcea, C., Milne, R. J., & Cotfas, L.-A. (2020). Determining the Number of Passengers for Each of Three Reverse Pyramid Boarding Groups with COVID-19 Flying Restrictions. *Symmetry*, 12(12), 2038. <https://doi.org/10.3390/sym12122038>
- Doddavaram, R., & Corlu, C. G. (2020). Teaching Risk Analytics Using R. In E. K.-H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing (Ed.), *Proceedings of the 2020 Winter Simulation Conference* (pp. 3272–3281). Online: IEEE. <https://doi.org/10.1109/WSC48552.2020.9384106>
- Donovan, B. (2020). COVID-19 Social Distancing Having Significant Impact on Mental Health, Study Shows. Retrieved February 10, 2021, from <https://www.swansea.ac.uk/press-office/news-events/news/2020/04/covid-19-social-distancing-having-significant-impact-on-mental-health-study-shows.php>
- Elcheroth, G., & Drury, J. (2020). Collective resilience in times of crisis: Lessons from the literature for socially effective responses to the pandemic. *British Journal of Social Psychology*, 59(3), 703–713. <https://doi.org/10.1111/bjso.12403>
- Ferreira, L.P., Ares, E., Peláez, G., Resano, A., Luis, C. J., & Tjahjono, B. (2012). Simulation of a Closed-Loops Assembly Line. *Key Engineering Materials*, 502, 127–132. <https://doi.org/10.4028/www.scientific.net/KEM.502.127>
- Ferreira, L.P., Ares, E., Peláez, G., Tjahjono, B., & Areal, J. J. (2012). Production Planning and Control in an Automobile Closed-Loops Assembly Line. *Key Engineering Materials*, 502, 103–108. <https://doi.org/10.4028/www.scientific.net/KEM.502.103>
- Ferreira, Luís Pinto. (2003). *Geração Automática de Modelos de Simulação de uma Linha de Produção na Indústria Electrónica*. Dissertação de Mestrado em Engenharia Industrial, Área de Especialização em Logística e Distribuição, Escola de Engenharia, Universidade do Minho.
- Ferreira, Luis Pinto, Gómez, E. A., Lourido, G. C. P., Quintas, J. D., & Tjahjono, B. (2012). Analysis and optimisation of a network of closed-loop automobile assembly line using simulation. *Int J Adv Manuf Technol*, 59, 351–366. <https://doi.org/10.1007/s00170-011-3502-4>
- Ferreira, Luís Pinto, Gómez, E. A., Lourido, G. P., Salgado, M., & Quintas, J. D. (2011).

- Analysis on the Influence of the Number of Pallets Circulating on an Automobile Closed-Loops Assembly Line. *International Journal of Advanced Engineering Sciences and Technologies*, 2(2), 119–123.
- Ferreira, Luís Pinto, Gómez, E. A., Lourido, G. P., & Tjahjono, B. (2013). Modeling a Complex Production Line Using Virtual Cells. In *Electrical Engineering and Intelligent Systems* (pp. 361–373). New York, NY: Springer.
- Ferreira, Luís Pinto, Pereira, G. A., & Machado, R. J. (2003). A simulação como ferramenta de apoio à decisão na engenharia da produção—O sistema GAMS. In *IV Congresso Galego de Estatística e Investigación de Operacións*. Vigo.
- Ferreira, Luís Pinto, Pereira, G. A., & Machado, R. J. (2005). Geração Automática de Modelos de Simulação de uma Linha de Montagem de Auto-Rádios. *Investigação Operacional*, 25(1), 37–62.
- IATA. (2020). Restarting aviation following COVID-19. Retrieved February 11, 2021, from <https://www.iata.org/contentassets/f1163430bba94512a583eb6d6b24aa56/covid-medical-evidence-for-strategies-200423.pdf>
- Jaehn, F., & Neumann, S. (2015). Airplane Boarding. *European Journal of Operational Research*, 244(2), 339–359. <https://doi.org/10.1016/j.ejor.2014.12.008>
- Jafer, S., & Mi, W. (2017). Comparative Study of Aircraft Boarding Strategies Using Cellular Discrete Event Simulation. *Aerospace*, 4(4), 57. <https://doi.org/10.3390/aerospace4040057>
- James F. Leathrum, J., Sokolowski, J. A., Shen, Y., & Audette, M. (2019). Software Skills Required by M&S Graduates for Des Development. In E. N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son (Ed.), *Proceedings of the 2019 Winter Simulation Conference* (pp. 3285–3294). National Harbor, Maryland: IEEE. <https://doi.org/10.1109/WSC40007.2019.9004908>
- Kalic, M., Markovic, B., & Kuljanin, J. (2013). The airline boarding problem: simulation based approach from different players' perspective. In M. Vidović, M. Kilibarda, S. Zečević, M. Miljuš, & G. Radivojević (Eds.), *1st Logistics International Conference* (pp. 49–54). Belgrade, Serbia.

- Kelton, W., Sadowski, R., & Sadowski, D. (1998). *Simulation with Arena*. (WCB/McGraw-Hill, Ed.) (2nd ed.). Boston.
- Kisiel, T. (2020). Resilience of passenger boarding strategies to priority fares offered by airlines. *Journal of Air Transport Management*, 87, 101853. <https://doi.org/10.1016/j.jairtraman.2020.101853>
- Law, A. M., & Kelton, W. D. (2007). *Simulation Modeling and Analysis* (Fifth Edit). McGraw-Hill.
- Marelli, S., Mattocks, G., & Merry, R. (1998). The role of computer simulation in reducing airplane turn time. Retrieved October 26, 2020, from https://www.boeing.com/commercial/aeromagazine/aero_01/textonly/t01txt.html
- Meyer, C. (2001). A Case in Case Study Methodology. *Field Methods*, 13(4), 329–352.
- Milne, R. J., Delcea, C., & Cotfas, L.-A. (2021). Airplane Boarding Methods that Reduce Risk from COVID-19. *Safety*, 134, 105061. <https://doi.org/10.1016/j.ssci.2020.105061>
- Muzik, M. (2017). Airline Turnaround Management. Retrieved November 5, 2020, from https://travel.report/Resources/Whitepapers/4f9da1d4-fc2e-40a2-ae16-b72f38c27ac9_lufthansasystems_turnaround_management_whitepaper_16feb2017.pdf
- Nyquist, D., & McFadden, K. (2008). A study of the airline boarding problem. *Journal of Air Transport Management*, 14(54), 197–204.
- Qiang, S.-J., Jia, B., Xie, D.-F., & Gao, Z.-Y. (2014). Reducing airplane boarding time by accounting for passengers' individual properties: A simulation based on cellular automaton. *Journal of Air Transport Management*, 40, 42–47. <https://doi.org/10.1016/j.jairtraman.2014.05.007>
- Regnier, E., Sanchez, S. M., & Sanchez, P. J. (2020). Testing-Based Interventions for COVID Pandemic Policies. In E. K.-H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing (Ed.), *Proceedings of the 2020 Winter Simulation Conference* (p. 2). Online. Retrieved from <https://informs-sim.org/wsc20papers/074.pdf>

- Ren, X., & Xu, X. (2018). Experimental analyses of airplane boarding based on interference classification. *Journal of Air Transport Management*, 71, 55–63. <https://doi.org/10.1016/j.jairtraman.2018.06.007>
- Rockwell Automation. (2010). Arena Professional Edition. *Premier Simulation Software Delivers Results, ARPRO-PP00*, 1–4. Retrieved from <https://www.hgds.de/assets/arena@-professional-edition.pdf>
- Rockwell Automation. (2021). Discrete Event Simulation Software. Retrieved April 16, 2020, from <https://www.arenasimulation.com/what-is-simulation/discrete-event-simulation-software>
- Ryzhov, I. O., Zhang, Q., & Chen, Y. (2020). Advanced Statistical Methods: Inference, Variable Selection and Experimental Design. In E. K.-H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing (Ed.), *Proceedings of the 2020 Winter Simulation Conference* (pp. 1–15). Online: IEEE. <https://doi.org/10.1109/WSC48552.2020.9383994>
- Sá, J., Ferreira, L. P., Dieguez, T., Sá, J. C., & Silva, F. J. G. (2020). Role of the Industry 4.0 in the Wine Production and Enotourism Sectors. In J. V. de Carvalho, Á. Rocha, P. Liberato, & A. P. Editors (Eds.), *International Conference on Tourism, Technology and Systems* (pp. 171–180). Singapore: Springer.
- Salari, M., Milne, R. J., Delcea, C., Kattan, L., & Cotfas, L.-A. (2020). Social distancing in airplane seat assignments. *Journal of Air Transport Management*, 89, 101915. <https://doi.org/10.1016/j.jairtraman.2020.101915>
- Schultz, M. (2017). Dynamic change of aircraft seat condition for fast boarding. *Transportation Research Part C: Emerging Technologies*, 85, 131–147. <https://doi.org/10.1016/j.trc.2017.09.014>
- Seatplans.com. (2021). Airbus A320. Retrieved March 20, 2020, from <http://seatplans.krakikom.de/airlines/azerbaijan-hava-yollary/azerbaijan-airlines-a320-seating-map/airbus-a320-business-short-haul>
- Silva, V., Ferreira, L. P., Silva, F. J. G., Tjahjono, B., & Ávila, P. (2021). Simulation-Based Decision Support System to Improve Material Flow of a Textile Company.

Sustainability, 13(5), 2947. Retrieved from 10.3390/su13052947

- Singh, J. (2019). Why Don't More Narrowbody Planes Use Multiple Doors To Board? *Simple Flying*. Retrieved from <https://simpleflying.com/airlines-board-deplane-doors/>
- Snyder, H. (2019). Literature review as a research methodology: An overview and guidelines. *Journal of Business Research*, 104, 333–339. <https://doi.org/10.1016/j.jbusres.2019.07.039>
- Soolaki, M., Mahdavi, I., Mahdavi-Amiri, N., Hassanzadeh, R., & Aghajani, A. (2012). A new linear programming approach and genetic algorithm for solving airline boarding problem. In *Applied Mathematical Modelling* (Vol. 36, pp. 4060–4072). <https://doi.org/10.1016/j.apm.2011.11.030>
- Steffen, J., & Hotchkiss, J. (2012). Experimental test of airplane boarding methods. *Journal of Air Transport Management*, 18(1), 64–67. <https://doi.org/10.1016/j.jairtraman.2011.10.003>
- Sturrock, D. T. (2020). Tested Success Tips for Simulation Project Excellence. In E. K.-H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing (Ed.), *Proceedings of the 2020 Winter Simulation Conference* (pp. 1143–1151). Online: IEEE. <https://doi.org/10.1109/WSC48552.2020.9384109>
- Tang, T., Wu, Y. H., Huang, H., & Caccetta, L. (2012). An aircraft boarding model accounting for passengers' individual properties. *Transportation Research Part C: Emerging Technologies*, 22, 1–16. <https://doi.org/10.1016/j.trc.2011.11.005>
- TAP AIR Portugal. (2021). Conhecer a Frota. Retrieved October 2, 2021, from <https://www.flytap.com/pt-pt/a-bordo/conhecer-a-frota?tabid=filters-tab5>
- Tariq, A., Roosa, K., & Chowell, G. (2020). Using Simple Dynamic Analytic Framework to Characterize and Forecast Epidemics. In E. K.-H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing (Ed.), *Proceedings of the 2020 Winter Simulation Conference* (pp. 30–44). Online: IEEE. <https://doi.org/10.1109/WSC48552.2020.9383968>
- Van Landeghem, H., & Beuselinck, A. (2002). Reducing passenger boarding time in

airplanes: A simulation based approach. *European Journal of Operational Research*, 142(2), 294–308. [https://doi.org/10.1016/S0377-2217\(01\)00294-6](https://doi.org/10.1016/S0377-2217(01)00294-6)

Vos, J. De. (2020). The effect of COVID-19 and subsequent social distancing on travel behavior. *Transportation Research Interdisciplinary Perspectives*, 5, 100121. <https://doi.org/10.1016/j.trip.2020.100121>

Zeineddine, H. (2017). A dynamically optimized aircraft boarding strategy. *Journal of Air Transport Management*, 58, 144–151. <https://doi.org/10.1016/j.jairtraman.2016.10.010>

Apêndice A – Código VBA

```
Private Sub ModelLogic_RunBeginReplication()
```

```
Set m = ThisDocument.Model
```

```
Set s = m.SIMAN
```

```
'Atribui objetos às variáveis
```

```
Set m = ThisDocument.Model
```

```
Set s = m.SIMAN
```

```
'luggage
```

```
s.VariableArrayValue(s.SymbolNumber("hasluggage")) = UserForm2.TextBox2.value
```

```
'arrival
```

```
Dim arrival As Integer 'Number of passagers
```

```
Dim pmax As Integer 'Number max of passagers
```

```
Dim fsc As Integer 'Number of first class passagers
```

```
pmax = 156
```

```
fsc = 24
```

```
arrival = UserForm2.TextBox1.value
```

```
s.VariableArrayValue(s.SymbolNumber("passagersmax")) = arrival
```

```
var_sta = UserForm3.Label10.Caption
```

```
s.VariableArrayValue(s.SymbolNumber("S_Strategy")) = var_sta
```

```
Select Case var_sta
```

```
'-----Random Strategy-----
```

```
Case 1
```

```
'dimension array with 132x2 integer elements
```

```
Dim b(131, 1) As Integer
```

```
'Normal Seats
```

```
For i = 1 To 132
```

```

For j = 1 To 2
    b(i - 1, j - 1) = s.VariableArrayValue(s.SymbolNumber("Random_Vector", i, j))
Next j
Next i

```

Randomize Timer

'swap array element i with a random element

```

For i = 0 To 131
    'generate random number and store as variable rnum
    rnum = Int(Rnd() * (UBound(b) - LBound(b) + 1) + LBound(b))
    rnum1 = b(rnum, 1)
    'temporarily store array element i to a variable called temp
    temp = b(i, 0)
    temp1 = b(i, 1)
    'set array element i equal to random array element rnum
    b(i, 0) = b(rnum, 0)
    b(i, 1) = rnum1
    'set array element rnum to old value of array element i stored as temp
    b(rnum, 0) = temp
    b(rnum, 1) = temp1
Next i

```

```

For i = 0 To 131
    For j = 0 To 1
        s.VariableArrayValue(s.SymbolNumber("NormalSeats", i + 1, j + 1)) = b(i, j)
    Next j
Next i

```

'-----Back to Front Strategy-----

Case 2

```

Dim b2(29, 1) As Integer 'dimension array with 30x2 integer elements
Dim b3(29, 1) As Integer 'dimension array with 30x2 integer elements
Dim b4(23, 1) As Integer 'dimension array with 24x2 integer elements
Dim b5(23, 1) As Integer 'dimension array with 24x2 integer elements

```

Dim b6(23, 1) As Integer 'dimension array with 24x2 integer elements

'-----Allocate Seats-----

'1st Block

For i = 1 To 30

For j = 1 To 2

b2(i - 1, j - 1) = s.VariableArrayValue(s.SymbolNumber("Bk_Vector1", i, j))

Next j

Next i

'2nd Block

For i = 1 To 30

For j = 1 To 2

b3(i - 1, j - 1) = s.VariableArrayValue(s.SymbolNumber("Bk_Vector2", i, j))

Next j

Next i

'3rd Block

For i = 1 To 24

For j = 1 To 2

b4(i - 1, j - 1) = s.VariableArrayValue(s.SymbolNumber("Bk_Vector3", i, j))

Next j

Next i

'4th Block

For i = 1 To 24

For j = 1 To 2

b5(i - 1, j - 1) = s.VariableArrayValue(s.SymbolNumber("Bk_Vector4", i, j))

Next j

Next i

'5th Block

For i = 1 To 24

```
For j = 1 To 2
    b6(i - 1, j - 1) = s.VariableArrayValue(s.SymbolNumber("Bk_Vector5", i, j))
Next j
Next i
```

```
'-----Randomize-----
```

```
Randomize Timer
```

```
For i = 0 To 29
    rnum = Int(Rnd() * (UBound(b2) - LBound(b2) + 1) + LBound(b2))
    rnum1 = b2(rnum, 1)
    temp = b2(i, 0)
    temp1 = b2(i, 1)
    b2(i, 0) = b2(rnum, 0)
    b2(i, 1) = rnum1
    b2(rnum, 0) = temp
    b2(rnum, 1) = temp1
Next i
```

```
For i = 0 To 29
    rnum = Int(Rnd() * (UBound(b3) - LBound(b3) + 1) + LBound(b3))
    rnum1 = b3(rnum, 1)
    temp = b3(i, 0)
    temp1 = b3(i, 1)
    b3(i, 0) = b3(rnum, 0)
    b3(i, 1) = rnum1
    b3(rnum, 0) = temp
    b3(rnum, 1) = temp1
Next i
```

```
For i = 0 To 23
    rnum = Int(Rnd() * (UBound(b4) - LBound(b4) + 1) + LBound(b4))
    rnum1 = b4(rnum, 1)
```

```

temp = b4(i, 0)
temp1 = b4(i, 1)
b4(i, 0) = b4(rnum, 0)
b4(i, 1) = rnum1
b4(rnum, 0) = temp
b4(rnum, 1) = temp1
Next i

For i = 0 To 23
    rnum = Int(Rnd() * (UBound(b5) - LBound(b5) + 1) + LBound(b5))
    rnum1 = b5(rnum, 1)
    temp = b5(i, 0)
    temp1 = b5(i, 1)
    b5(i, 0) = b5(rnum, 0)
    b5(i, 1) = rnum1
    b5(rnum, 0) = temp
    b5(rnum, 1) = temp1
Next i

For i = 0 To 23
    rnum = Int(Rnd() * (UBound(b6) - LBound(b6) + 1) + LBound(b6))
    rnum1 = b6(rnum, 1)
    temp = b6(i, 0)
    temp1 = b6(i, 1)
    b6(i, 0) = b6(rnum, 0)
    b6(i, 1) = rnum1
    b6(rnum, 0) = temp
    b6(rnum, 1) = temp1
Next i

'-----Fill the final array elements shuffled-----

For i = 0 To 29
    For j = 0 To 1

```

```

        s.VariableArrayValue(s.SymbolNumber("NormalSeats", i + 1, j + 1)) = b2(i, j)
    Next j
Next i

For i = 0 To 29
    For j = 0 To 1
        s.VariableArrayValue(s.SymbolNumber("NormalSeats", i + 1 + 30, j + 1)) = b3(i, j)
    Next j
Next i

For i = 0 To 23
    For j = 0 To 1
        s.VariableArrayValue(s.SymbolNumber("NormalSeats", i + 1 + 30 + 30, j + 1)) = b4(i, j)
    Next j
Next i

For i = 0 To 23
    For j = 0 To 1
        s.VariableArrayValue(s.SymbolNumber("NormalSeats", i + 1 + 30 + 30 + 24, j + 1)) = b5(i, j)
    Next j
Next i

For i = 0 To 23
    For j = 0 To 1
        s.VariableArrayValue(s.SymbolNumber("NormalSeats", i + 1 + 30 + 30 + 24 + 24, j + 1)) = b6(i, j)
    Next j
Next i

```

'----- Outside In Strategy-----'

Case 3

Dim c2(43, 1) As Integer 'dimension array with 44 integer elements

Dim c3(43, 1) As Integer 'dimension array with 44 integer elements

Dim c4(43, 1) As Integer 'dimension array with 44 integer elements

'-----Allocate Seats-----'

'Window Seats

For i = 1 To 44

For j = 1 To 2

c2(i - 1, j - 1) = s.VariableArrayValue(s.SymbolNumber("OutIn_Vector1", i, j))

Next j

Next i

'Middle Seats

For i = 1 To 44

For j = 1 To 2

c3(i - 1, j - 1) = s.VariableArrayValue(s.SymbolNumber("OutIn_Vector2", i, j))

Next j

Next i

'Aisle Seats

For i = 1 To 44

For j = 1 To 2

c4(i - 1, j - 1) = s.VariableArrayValue(s.SymbolNumber("OutIn_Vector3", i, j))

Next j

Next i

'-----Randomize-----'

Randomize Timer

For i = 0 To 43

rnum = Int(Rnd() * (UBound(c2) - LBound(c2) + 1) + LBound(c2))

rnum1 = c2(rnum, 1)

temp = c2(i, 0)

temp1 = c2(i, 1)

c2(i, 0) = c2(rnum, 0)

c2(i, 1) = rnum1

c2(rnum, 0) = temp

c2(rnum, 1) = temp1

Next i

For i = 0 To 43

 rnum = Int(Rnd() * (UBound(c3) - LBound(c3) + 1) + LBound(c3)

 rnum1 = c3(rnum, 1)

 temp = c3(i, 0)

 temp1 = c3(i, 1)

 c3(i, 0) = c3(rnum, 0)

 c3(i, 1) = rnum1

 c3(rnum, 0) = temp

 c3(rnum, 1) = temp1

Next i

For i = 0 To 43

 rnum = Int(Rnd() * (UBound(c4) - LBound(c4) + 1) + LBound(c4))

 rnum1 = c4(rnum, 1)

 temp = c4(i, 0)

 temp1 = c4(i, 1)

 c4(i, 0) = c4(rnum, 0)

 c4(i, 1) = rnum1

 c4(rnum, 0) = temp 'set array element rnum to old value of array element i stored as temp

 c4(rnum, 1) = temp1

Next i

'-----Fill the final array elements shuffled-----

For i = 0 To 43

 For j = 0 To 1

 s.VariableArrayValue(s.SymbolNumber("NormalSeats", i + 1, j + 1)) = c2(i, j)

 Next j

Next i

For i = 0 To 43

 For j = 0 To 1

 s.VariableArrayValue(s.SymbolNumber("NormalSeats", i + 1 + 44, j + 1)) = c3(i, j)

```

Next j
Next i

For i = 0 To 43
  For j = 0 To 1
    s.VariableArrayValue(s.SymbolNumber("NormalSeats", i + 1 + 44 + 44, j + 1)) = c4(i, j)
  Next j
Next i

```

'-----Reversed Pyramid Strategy-----'

Case 4

```

Dim d2(25, 1) As Integer 'dimension array with 26 integer elements
Dim d3(25, 1) As Integer 'dimension array with 26 integer elements
Dim d4(25, 1) As Integer 'dimension array with 26 integer elements
Dim d5(25, 1) As Integer 'dimension array with 26 integer elements
Dim d6(27, 1) As Integer 'dimension array with 28 integer elements

```

'-----Allocate Seats-----'

```

For i = 1 To 26
  For j = 1 To 2
    d2(i - 1, j - 1) = s.VariableArrayValue(s.SymbolNumber("RP_Vector1", i, j))
  Next j
Next i

```

```

For i = 1 To 26
  For j = 1 To 2
    d3(i - 1, j - 1) = s.VariableArrayValue(s.SymbolNumber("RP_Vector2", i, j))
  Next j
Next i

```

```

For i = 1 To 26
  For j = 1 To 2
    d4(i - 1, j - 1) = s.VariableArrayValue(s.SymbolNumber("RP_Vector3", i, j))
  Next j
Next i

```

```

Next j
Next i

For i = 1 To 26
  For j = 1 To 2
    d5(i - 1, j - 1) = s.VariableArrayValue(s.SymbolNumber("RP_Vector4", i, j))
  Next j
Next i

For i = 1 To 28
  For j = 1 To 2
    d6(i - 1, j - 1) = s.VariableArrayValue(s.SymbolNumber("RP_Vector5", i, j))
  Next j
Next i

'-----Randomize-----
Randomize Timer

For i = 0 To 25
  rnum = Int(Rnd() * (UBound(d2) - LBound(d2) + 1) + LBound(d2))
  rnum1 = d2(rnum, 1)
  temp = d2(i, 0)
  temp1 = d2(i, 1)
  d2(i, 0) = d2(rnum, 0)
  d2(i, 1) = rnum1
  d2(rnum, 0) = temp
  d2(rnum, 1) = temp1
Next i

For i = 0 To 25
  rnum = Int(Rnd() * (UBound(d3) - LBound(d3) + 1) + LBound(d3))
  rnum1 = d3(rnum, 1)
  temp = d3(i, 0)
  temp1 = d3(i, 1)
  d3(i, 0) = d3(rnum, 0)

```

```
d3(i, 1) = rnum1  
d3(rnum, 0) = temp  
d3(rnum, 1) = temp1
```

```
Next i
```

```
For i = 0 To 25
```

```
  rnum = Int(Rnd() * (UBound(d4) - LBound(d4) + 1) + LBound(d4))  
  rnum1 = d4(rnum, 1)  
  temp = d4(i, 0)  
  temp1 = d4(i, 1)  
  d4(i, 0) = d4(rnum, 0)  
  d4(i, 1) = rnum1  
  d4(rnum, 0) = temp  
  d4(rnum, 1) = temp1
```

```
Next i
```

```
For i = 0 To 25
```

```
  rnum = Int(Rnd() * (UBound(d5) - LBound(d5) + 1) + LBound(d5))  
  rnum1 = d5(rnum, 1)  
  temp = d5(i, 0)  
  temp1 = d5(i, 1)  
  d5(i, 0) = d5(rnum, 0)  
  d5(i, 1) = rnum1  
  d5(rnum, 0) = temp  
  d5(rnum, 1) = temp1
```

```
Next i
```

```
For i = 0 To 27
```

```
  rnum = Int(Rnd() * (UBound(d6) - LBound(d6) + 1) + LBound(d6))  
  rnum1 = d6(rnum, 1)  
  temp = d6(i, 0)  
  temp1 = d6(i, 1)  
  d6(i, 0) = d6(rnum, 0)
```

```

d6(i, 1) = rnum1
d6(rnum, 0) = temp
d6(rnum, 1) = temp1
Next i
'-----Fill the final array elements shuffled-----
For i = 0 To 25
  For j = 0 To 1
    s.VariableArrayValue(s.SymbolNumber("NormalSeats", i + 1, j + 1)) = d2(i, j)
  Next j
Next i

For i = 0 To 25
  For j = 0 To 1
    s.VariableArrayValue(s.SymbolNumber("NormalSeats", i + 1 + 26, j + 1)) = d3(i, j)
  Next j
Next i

For i = 0 To 25
  For j = 0 To 1
    s.VariableArrayValue(s.SymbolNumber("NormalSeats", i + 1 + 26 + 26, j + 1)) = d4(i, j)
  Next j
Next i

For i = 0 To 25
  For j = 0 To 1
    s.VariableArrayValue(s.SymbolNumber("NormalSeats", i + 1 + 26 + 26 + 26, j + 1)) = d5(i, j)
  Next j
Next i

For i = 0 To 27
  For j = 0 To 1
    s.VariableArrayValue(s.SymbolNumber("NormalSeats", i + 1 + 26 + 26 + 26 + 26, j + 1)) = d6(i, j)
  Next j

```

Next i

'-----Blocks Strategy-----'

Case 5

Dim e2(29, 1) As Integer 'dimension array with 30x2 integer elements

Dim e3(23, 1) As Integer 'dimension array with 24x2 integer elements

Dim e4(29, 1) As Integer 'dimension array with 30x2 integer elements

Dim e5(23, 1) As Integer 'dimension array with 24x2 integer elements

Dim e6(23, 1) As Integer 'dimension array with 24x2 integer elements

'-----Allocate Seats-----'

'1st Block

For i = 1 To 30

For j = 1 To 2

e2(i - 1, j - 1) = s.VariableArrayValue(s.SymbolNumber("Bk_Vector1", i, j))

Next j

Next i

'2nd Block

For i = 1 To 24

For j = 1 To 2

e3(i - 1, j - 1) = s.VariableArrayValue(s.SymbolNumber("Bk_Vector5", i, j))

Next j

Next i

'3rd Block

For i = 1 To 30

For j = 1 To 2

e4(i - 1, j - 1) = s.VariableArrayValue(s.SymbolNumber("Bk_Vector2", i, j))

Next j

Next i

'4th Block

```
For i = 1 To 24
  For j = 1 To 2
    e5(i - 1, j - 1) = s.VariableArrayValue(s.SymbolNumber("Bk_Vector4", i, j))
  Next j
Next i
```

'5th Block

```
For i = 1 To 24
  For j = 1 To 2
    e6(i - 1, j - 1) = s.VariableArrayValue(s.SymbolNumber("Bk_Vector3", i, j))
  Next j
Next i
```

'-----Randomize-----

Randomize Timer

```
For i = 0 To 29
  rnum = Int(Rnd() * (UBound(e2) - LBound(e2) + 1) + LBound(e2))
  rnum1 = e2(rnum, 1)
  temp = e2(i, 0)
  temp1 = e2(i, 1)
  e2(i, 0) = e2(rnum, 0)
  e2(i, 1) = rnum1
  e2(rnum, 0) = temp
  e2(rnum, 1) = temp1
Next i
```

For i = 0 To 23

```
  rnum = Int(Rnd() * (UBound(e3) - LBound(e3) + 1) + LBound(e3))
  rnum1 = e3(rnum, 1)
  temp = e3(i, 0)
  temp1 = e3(i, 1)
  e3(i, 0) = e3(rnum, 0)
  e3(i, 1) = rnum1
```

e3(rnum, 0) = temp

e3(rnum, 1) = temp1

Next i

For i = 0 To 29

rnum = Int(Rnd() * (UBound(e4) - LBound(e4) + 1) + LBound(e4))

rnum1 = e4(rnum, 1)

temp = e4(i, 0)

temp1 = e4(i, 1)

e4(i, 0) = e4(rnum, 0)

e4(i, 1) = rnum1

e4(rnum, 0) = temp

e4(rnum, 1) = temp1

Next i

For i = 0 To 23

rnum = Int(Rnd() * (UBound(e5) - LBound(e5) + 1) + LBound(e5))

rnum1 = e5(rnum, 1)

temp = e5(i, 0)

temp1 = e5(i, 1)

e5(i, 0) = e5(rnum, 0)

e5(i, 1) = rnum1

e5(rnum, 0) = temp

e5(rnum, 1) = temp1

Next i

For i = 0 To 23

rnum = Int(Rnd() * (UBound(e6) - LBound(e6) + 1) + LBound(e6))

rnum1 = e6(rnum, 1)

temp = e6(i, 0)

temp1 = e6(i, 1)

e6(i, 0) = e6(rnum, 0)

e6(i, 1) = rnum1

```

e6(rnum, 0) = temp
e6(rnum, 1) = temp1
Next i
'-----Fill the final array elements shuffled-----
For i = 0 To 29
  For j = 0 To 1
    s.VariableArrayValue(s.SymbolNumber("NormalSeats", i + 1, j + 1)) = e2(i, j)
  Next j
Next i

For i = 0 To 23
  For j = 0 To 1
    s.VariableArrayValue(s.SymbolNumber("NormalSeats", i + 1 + 30, j + 1)) = e3(i, j)
  Next j
Next i

For i = 0 To 29
  For j = 0 To 1
    s.VariableArrayValue(s.SymbolNumber("NormalSeats", i + 1 + 30 + 24, j + 1)) = e4(i, j)
  Next j
Next i

For i = 0 To 23
  For j = 0 To 1
    s.VariableArrayValue(s.SymbolNumber("NormalSeats", i + 1 + 30 + 24 + 30, j + 1)) = e5(i, j)
  Next j
Next i

For i = 0 To 23
  For j = 0 To 1
    s.VariableArrayValue(s.SymbolNumber("NormalSeats", i + 1 + 30 + 24 + 30 + 24, j + 1)) = e6(i, j)
  Next j
Next i

```

```
'-----Steffen Strategy-----
```

Case 6

```
Dim f(131, 1) As Integer 'dimension array with 5 integer elements
```

```
'-----Allocate Seats-----
```

```
For i = 1 To 132
```

```
    For j = 1 To 2
```

```
        f(i - 1, j - 1) = s.VariableArrayValue(s.SymbolNumber("Steffen_Vector", i, j))
```

```
    Next j
```

```
Next i
```

```
'-----Fill the final array elements shuffled-----
```

```
'test that array elements were shuffled
```

```
For i = 0 To 131
```

```
    For j = 0 To 1
```

```
        s.VariableArrayValue(s.SymbolNumber("NormalSeats", i + 1, j + 1)) = f(i, j)
```

```
    Next j
```

```
Next i
```

```
'-----Modified Optimal Strategy-----
```

Case 7

```
Dim g2(32, 1) As Integer 'dimension array with 33x2 integer elements
```

```
Dim g3(32, 1) As Integer 'dimension array with 33x2 integer elements
```

```
Dim g4(32, 1) As Integer 'dimension array with 33x2 integer elements
```

```
Dim g5(32, 1) As Integer 'dimension array with 33x2 integer elements
```

```
'-----Allocate Seats-----
```

```
For i = 1 To 33
```

```
    For j = 1 To 2
```

```
        g2(i - 1, j - 1) = s.VariableArrayValue(s.SymbolNumber("MO_Vector1", i, j))
```

```
    Next j
```

```
Next i
```

```
For i = 1 To 33
  For j = 1 To 2
    g3(i - 1, j - 1) = s.VariableArrayValue(s.SymbolNumber("MO_Vector2", i, j))
  Next j
Next i
```

```
For i = 1 To 33
  For j = 1 To 2
    g4(i - 1, j - 1) = s.VariableArrayValue(s.SymbolNumber("MO_Vector3", i, j))
  Next j
Next i
```

```
For i = 1 To 33
  For j = 1 To 2
    g5(i - 1, j - 1) = s.VariableArrayValue(s.SymbolNumber("MO_Vector4", i, j))
  Next j
Next i
```

'-----Randomize-----'

Randomize Timer

```
For i = 0 To 32
  rnum = Int(Rnd() * (UBound(g2) - LBound(g2) + 1) + LBound(g2))
  rnum1 = g2(rnum, 1)
  temp = g2(i, 0)
  temp1 = g2(i, 1)
  g2(i, 0) = g2(rnum, 0)
  g2(i, 1) = rnum1
  g2(rnum, 0) = temp
  g2(rnum, 1) = temp1
Next i
```

For i = 0 To 32

 rnum = Int(Rnd() * (UBound(g3) - LBound(g3) + 1) + LBound(g3))

 rnum1 = g3(rnum, 1)

 temp = g3(i, 0)

 temp1 = g3(i, 1)

 g3(i, 0) = g3(rnum, 0)

 g3(i, 1) = rnum1

 g3(rnum, 0) = temp

 g3(rnum, 1) = temp1

Next i

For i = 0 To 32

 rnum = Int(Rnd() * (UBound(g4) - LBound(g4) + 1) + LBound(g4))

 rnum1 = g4(rnum, 1)

 temp = g4(i, 0)

 temp1 = g4(i, 1)

 g4(i, 0) = g4(rnum, 0)

 g4(i, 1) = rnum1

 g4(rnum, 0) = temp

 g4(rnum, 1) = temp1

Next i

For i = 0 To 32

 rnum = Int(Rnd() * (UBound(g5) - LBound(g5) + 1) + LBound(g5))

 rnum1 = g5(rnum, 1)

 temp = g5(i, 0)

 temp1 = g5(i, 1)

 g5(i, 0) = g5(rnum, 0) '

 g5(i, 1) = rnum1

 g5(rnum, 0) = temp

 g5(rnum, 1) = temp1

Next i

'-----Fill the final array elements shuffled-----'

```

'test that array elements were shuffled

For i = 0 To 32
  For j = 0 To 1
    s.VariableArrayValue(s.SymbolNumber("NormalSeats", i + 1, j + 1)) = g2(i, j)
  Next j
Next i

For i = 0 To 32
  For j = 0 To 1
    s.VariableArrayValue(s.SymbolNumber("NormalSeats", i + 1 + 33, j + 1)) = g3(i, j)
  Next j
Next i

For i = 0 To 32
  For j = 0 To 1
    s.VariableArrayValue(s.SymbolNumber("NormalSeats", i + 1 + 33 + 33, j + 1)) = g4(i, j)
  Next j
Next i

For i = 0 To 32
  For j = 0 To 1
    s.VariableArrayValue(s.SymbolNumber("NormalSeats", i + 1 + 33 + 33 + 33, j + 1)) = g5(i, j)
  Next j
Next i

End Select

'Final disposition of the seats
Dim bf(23, 1) As Integer 'dimension array with 8 integer elements
Dim bn(132, 1) As Integer 'dimension array with 24 integer elements
Dim n As Integer 'number of normal seats (First Class Seats are Fixed
n = 131

'Allocate Seats

```

```

For i = 1 To fsc
  For j = 1 To 2
    bf(i - 1, j - 1) = s.VariableArrayValue(s.SymbolNumber("FSC_Vector", i, j))
  Next j
Next i

```

'Normal Seats

```

For i = 1 To 132
  For j = 1 To 2
    bn(i - 1, j - 1) = s.VariableArrayValue(s.SymbolNumber("NormalSeats", i, j))
  Next j
Next i

```

'-----Check and Fill if plain isn't full-----

If arrival < pmax And UserForm2.OptionButton4.value Then

Randomize Timer

For i = 0 To pmax - arrival - 1

rnum = Int(Rnd() * (UBound(bn) - LBound(bn) + 1 - i) + LBound(bn))

'This loop will shift array position one place to the left. This way the number at your position will be replaced by the next number.

For j = rnum To (n - 1)

bn(j, 0) = bn(j + 1, 0)

bn(j, 1) = bn(j + 1, 1)

Next j

n = n - 1

Next i

End If

'-----Randomize First Class Seats-----

'First Class Seats

Randomize Timer

For i = 0 To fsc - 1

rnum = Int(Rnd() * (UBound(bf) - LBound(bf) + 1) + LBound(bf))

```
rnum1 = bf(rnum, 1)
```

```
temp = bf(i, 0)
```

```
temp1 = bf(i, 1)
```

```
bf(i, 0) = bf(rnum, 0)
```

```
bf(i, 1) = rnum1
```

```
bf(rnum, 0) = temp
```

```
bf(rnum, 1) = temp1
```

```
Next i
```

```
'-----Fill the final array elements shuffled-----
```

```
For i = 0 To fsc - 1
```

```
    For j = 0 To 1
```

```
        s.VariableArrayValue(s.SymbolNumber("SeatArray", i + 1, j + 1)) = bf(i, j)
```

```
    Next j
```

```
Next i
```

```
'-----Check if disease mode its active-----
```

```
If UserForm2.OptionButton3.value Then
```

```
    Dim midlnumber As Integer
```

```
    Dim midlvector(43, 2) As Integer
```

```
    midlnumber = UserForm2.TextBox11.value
```

```
    n = 131
```

```
    ms = 0
```

```
    For i = 0 To n
```

```
        If (bn(i, 1) = 2 Or bn(i, 1) = 5) And n > 87 Then
```

```
            midlvector(ms, 0) = bn(i, 0)
```

```
            midlvector(ms, 1) = bn(i, 1)
```

```
            For j = i To (n - 1)
```

```
                bn(j, 0) = bn(j + 1, 0)
```

```
                bn(j, 1) = bn(j + 1, 1)
```

```

    Next j

    n = n - 1

    i = i - 1

    ms = ms + 1

End If

Next i

Randomize Timer

For i = 0 To midlnumber - 1

    rnum = Int(Rnd() * (UBound(midlvector) - LBound(midlvector) + 1) + LBound(midlvector))

    rnum1 = midlvector(rnum, 1)

    temp = midlvector(i, 0)

    temp1 = midlvector(i, 1)

    midlvector(i, 0) = midlvector(rnum, 0) element rnum

    midlvector(i, 1) = rnum1

    midlvector(rnum, 0) = temp stored as temp

    midlvector(rnum, 1) = temp1

Next i

For i = 0 To midlnumber - 1

    fg = 1

    For j = 0 To (n - 1)

        If bn(j, 0) = midlvector(i, 0) And fg = 1 Then

            If bn(j, 1) + 1 = midlvector(i, 1) Or bn(j, 1) - 1 = midlvector(i, 1) Then

                fg = 0

                rnum = j + 1

            End If

        End If

    End For

    'This loop will find the position a position on the same row and side as the middle number

Next j

n = n + 1

For j = rnum To (n - 1)

    bn(n - j + rnum, 0) = bn(n - j + rnum - 1, 0)

```

```

    bn(n - j + rnum, 1) = bn(n - j + rnum - 1, 1)
Next j
bn(rnum, 0) = midllevector(i, 0)
bn(rnum, 1) = midllevector(i, 1)
Next i

s.VariableArrayValue(s.SymbolNumber("passagersmax")) = n + 1 + fsc

```

End If

'-----Check if has priority-----'

```

If UserForm2.OptionButton2.value Then 'If not
    For i = 0 To n
        For j = 0 To 1
            s.VariableArrayValue(s.SymbolNumber("SeatArray", i + 1 + fsc, j + 1)) = bn(i, j)
        Next j
    Next i
End If

```

If UserForm2.OptionButton1.value Then 'If Yes

```
Dim prioritynum As Integer
```

```
Dim vectorout(131, 1) As Integer
```

```
prioritynum = UserForm2.TextBox8.value
```

```
Randomize Timer
```

```
For i = 0 To prioritynum - 1
```

```
    rnum = Int(Rnd() * (UBound(bn) - LBound(bn) + 1 - i - (pmax - 24 - n)) + LBound(bn)) 'generate
random number and store as variable rnum
```

```
    vectorout(i, 0) = bn(rnum, 0)
```

```
    vectorout(i, 1) = bn(rnum, 1)
```

```
    For j = rnum To (n - 1)
```

```

        bn(j, 0) = bn(j + 1, 0)
        bn(j, 1) = bn(j + 1, 1)
    Next j
    n = n - 1
Next i

For i = 0 To n
    For j = 0 To 1
        s.VariableArrayValue(s.SymbolNumber("SeatArray", i + 1 + fsc + prioritynum, j + 1)) = bn(i, j)
    Next j
Next i

For i = 0 To prioritynum - 1
    For j = 0 To 1
        s.VariableArrayValue(s.SymbolNumber("SeatArray", i + 1 + fsc, j + 1)) = vectorout(i, j)
    Next j
Next i

End If

End Sub

Private Sub ModelLogic_RunBeginSimulation()

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

UserForm1.Show

End Sub

Private Sub ModelLogic_RunEndReplication()

```

'Dimensionamento objetos gerais do Arena/SIMAN

Dim m As Model

Dim s As SIMAN

'Atribui objetos às variáveis

Set m = ThisDocument.Model

Set s = m.SIMAN

Dim boarding(99) As Integer 'dimension array with 100 integer elements

var_rep = UserForm3.Label12.Caption

UserForm3.Label12.Caption = var_rep + 1

If UserForm3.Label12.Caption > 1 Then

var_boa = s.VariableArrayValue(s.SymbolNumber("Boarding_Time"))

boarding(var_rep) = s.VariableArrayValue(s.SymbolNumber("Boarding_Time"))

var_ava = s.VariableArrayValue(s.SymbolNumber("Ind_Avarage"))

var_max = s.VariableArrayValue(s.SymbolNumber("Ind_Max"))

var_min = s.VariableArrayValue(s.SymbolNumber("Ind_Min"))

'Atualizar medias boarding, total / individual

UserForm3.Label5.Caption = ((UserForm3.Label5.Caption * var_rep) + var_boa) /
UserForm3.Label12.Caption

UserForm3.Label6.Caption = ((UserForm3.Label6.Caption * var_rep) + var_ava) /
UserForm3.Label12.Caption

'atualizar boarding máximo

If var_boa > Val(UserForm3.Label16.Caption) Then

UserForm3.Label16.Caption = var_boa

End If

'atualizar boarding mínimo

If var_boa < Val(UserForm3.Label15.Caption) Then

UserForm3.Label15.Caption = var_boa

End If

'Passageiro mais rápido

If var_max > Val(UserForm3.Label8.Caption) Then

UserForm3.Label8.Caption = var_max

End If

'Passageiro mais lento

If var_min < Val(UserForm3.Label7.Caption) Then

UserForm3.Label7.Caption = var_min

End If

Else

boarding(0) = s.VariableArrayValue(s.SymbolNumber("Boarding_Time"))

UserForm3.Label5.Caption = s.VariableArrayValue(s.SymbolNumber("Boarding_Time"))

UserForm3.Label15.Caption = s.VariableArrayValue(s.SymbolNumber("Boarding_Time"))

UserForm3.Label16.Caption = s.VariableArrayValue(s.SymbolNumber("Boarding_Time"))

UserForm3.Label6.Caption = s.VariableArrayValue(s.SymbolNumber("Ind_Avarage"))

UserForm3.Label7.Caption = s.VariableArrayValue(s.SymbolNumber("Ind_Min"))

UserForm3.Label8.Caption = s.VariableArrayValue(s.SymbolNumber("Ind_Max"))

End If

End Sub

Private Sub ModelLogic_RunEndSimulation()

'Dimensionamento objetos gerais do Arena/SIMAN

Dim m As Model

Dim s As SIMAN

'Atribui objetos às variáveis

Set m = ThisDocument.Model

Set s = m.SIMAN

```
var_sta = UserForm3.Label10.Caption
```

```
Select Case var_sta
```

```
Case 1
```

```
    UserForm3.Label10.Caption = "Random"
```

```
Case 2
```

```
    UserForm3.Label10.Caption = "Back to Front"
```

```
Case 3
```

```
    UserForm3.Label10.Caption = "Outside In"
```

```
Case 4
```

```
    UserForm3.Label10.Caption = "Reversed Pyramid"
```

```
Case 5
```

```
    UserForm3.Label10.Caption = "Blocks"
```

```
Case 6
```

```
    UserForm3.Label10.Caption = "Steffen"
```

```
Case 7
```

```
    UserForm3.Label10.Caption = "Modified Optimal"
```

```
End Select
```

```
UserForm3.Show
```

```
End Sub
```

```
Private Sub CommandButton1_Click()
```

```
Set m = ThisDocument.Model
```

```
Set s = m.SIMAN
```

```
UserForm1.Hide
```

```
End Sub
```

```
Private Sub CommandButton2_Click()
```

```
UserForm2.MultiPage1.value = 0 'é a posição relativa do separador, começa em 0
```

```
UserForm2.Show
```

```
End Sub
```

```
Private Sub ComboBox1_Change()
```

```
'Em caso de alteração da combobox
```

```
Var_Combobox1Alterada = UserForm2.ComboBox1.ListIndex
```

```
Select Case Var_Combobox1Alterada
```

```
Case 1 'Exponencial
```

```
'Esconde as labels e textboxes
```

```
UserForm2.Label59.Visible = False
```

```
UserForm2.Label60.Visible = False
```

```
UserForm2.TextBox25.Visible = False
```

```
UserForm2.TextBox26.Visible = False
```

```
'Apresentam as labels e textboxes relevantes
```

```
UserForm2.Label58.Caption = "Mean"
```

```
UserForm2.Label58.Visible = True
```

```
UserForm2.TextBox24.Visible = True
```

```
'Mostra botão para alteração das variáveis
```

```
UserForm2.CommandButton20.Visible = True
```

```
UserForm2.Label57.Visible = True
```

```
Case 2 'Poisson
```

```
'Esconde as labels e textboxes
```

```
UserForm2.Label59.Visible = False
```

```
UserForm2.Label60.Visible = False
```

```
UserForm2.TextBox25.Visible = False
```

```
UserForm2.TextBox26.Visible = False
```

```
'Apresentam as labels e textboxes relevantes
```

```
UserForm2.Label58.Caption = "Mean"  
UserForm2.Label58.Visible = True  
UserForm2.TextBox24.Visible = True  
'Mostra botão para alteração das variaveis  
UserForm2.CommandButton20.Visible = True  
UserForm2.Label57.Visible = True
```

Case 3 'Normal

```
'Esconde as labels e textboxes  
UserForm2.Label60.Visible = False  
UserForm2.TextBox26.Visible = False  
'Apresentam as labels e textboxes relevantes  
UserForm2.Label58.Caption = "Mean"  
UserForm2.Label58.Visible = True  
UserForm2.Label59.Caption = "StdDev"  
UserForm2.Label59.Visible = True  
UserForm2.TextBox24.Visible = True  
UserForm2.TextBox25.Visible = True  
'Mostra botão para alteração das variaveis  
UserForm2.CommandButton20.Visible = True  
UserForm2.Label57.Visible = True
```

Case 4 'LogNormal

```
'Esconde as labels e textboxes  
UserForm2.Label60.Visible = False  
UserForm2.TextBox26.Visible = False  
'Apresentam as labels e textboxes relevantes  
UserForm2.Label58.Caption = "LogMean"  
UserForm2.Label58.Visible = True  
UserForm2.Label59.Caption = "LogStd"  
UserForm2.Label59.Visible = True  
UserForm2.TextBox24.Visible = True  
UserForm2.TextBox25.Visible = True  
'Mostra botão para alteração das variaveis  
UserForm2.CommandButton20.Visible = True
```

UserForm2.Label57.Visible = True

Case 5 'Uniform

'Esconde as labels e textboxes

UserForm2.Label60.Visible = False

UserForm2.TextBox26.Visible = False

'Apresentam as labels e textboxes relevantes

UserForm2.Label58.Caption = "Minimum"

UserForm2.Label58.Visible = True

UserForm2.Label59.Caption = "Maximum"

UserForm2.Label59.Visible = True

UserForm2.TextBox24.Visible = True

UserForm2.TextBox25.Visible = True

'Mostra botão para alteração das variaveis

UserForm2.CommandButton20.Visible = True

UserForm2.Label57.Visible = True

Case 6 'Triangular

UserForm2.Label58.Caption = "Minimum"

UserForm2.Label58.Visible = True

UserForm2.Label59.Caption = "Mode"

UserForm2.Label59.Visible = True

UserForm2.Label60.Caption = "Maximum"

UserForm2.Label60.Visible = True

UserForm2.TextBox24.Visible = True

UserForm2.TextBox25.Visible = True

UserForm2.TextBox26.Visible = True

'Mostra botão para alteração das variaveis

UserForm2.CommandButton20.Visible = True

UserForm2.Label57.Visible = True

End Select

End Sub

Private Sub ComboBox2_Change()

'Em caso de alteração da combobox

Var_Combobox2Alterada = UserForm2.ComboBox2.ListIndex

Select Case Var_Combobox2Alterada

Case 1 'Exponencial

'Esconde as labels e textboxes

UserForm2.Label71.Visible = False

UserForm2.Label72.Visible = False

UserForm2.TextBox35.Visible = False

UserForm2.TextBox36.Visible = False

'Apresentam as labels e textboxes relevantes

UserForm2.Label70.Caption = "Mean"

UserForm2.Label70.Visible = True

UserForm2.TextBox34.Visible = True

'Mostra botão para alteração das variaveis

UserForm2.CommandButton21.Visible = True

UserForm2.Label69.Visible = True

Case 2 'Poisson

'Esconde as labels e textboxes

UserForm2.Label71.Visible = False

UserForm2.Label72.Visible = False

UserForm2.TextBox35.Visible = False

UserForm2.TextBox36.Visible = False

'Apresentam as labels e textboxes relevantes

UserForm2.Label70.Caption = "Mean"

UserForm2.Label70.Visible = True

UserForm2.TextBox34.Visible = True

'Mostra botão para alteração das variaveis

UserForm2.CommandButton21.Visible = True

UserForm2.Label69.Visible = True

Case 3 'Normal

'Esconde as labels e textboxes

UserForm2.Label72.Visible = False

UserForm2.TextBox36.Visible = False

'Apresentam as labels e textboxes relevantes

UserForm2.Label70.Caption = "Mean"

UserForm2.Label70.Visible = True

UserForm2.Label71.Caption = "StdDev"

UserForm2.Label71.Visible = True

UserForm2.TextBox34.Visible = True

UserForm2.TextBox35.Visible = True

'Mostra botão para alteração das variáveis

UserForm2.CommandButton21.Visible = True

UserForm2.Label69.Visible = True

Case 4 'LogNormal

'Esconde as labels e textboxes

UserForm2.Label72.Visible = False

UserForm2.TextBox36.Visible = False

'Apresentam as labels e textboxes relevantes

UserForm2.Label70.Caption = "LogMean"

UserForm2.Label70.Visible = True

UserForm2.Label71.Caption = "LogStd"

UserForm2.Label71.Visible = True

UserForm2.TextBox34.Visible = True

UserForm2.TextBox35.Visible = True

'Mostra botão para alteração das variáveis

UserForm2.CommandButton21.Visible = True

UserForm2.Label69.Visible = True

Case 5 'Uniform

'Esconde as labels e textboxes

UserForm2.Label72.Visible = False

UserForm2.TextBox36.Visible = False

'Apresentam as labels e textboxes relevantes

UserForm2.Label70.Caption = "Minimum"

UserForm2.Label70.Visible = True

```
UserForm2.Label71.Caption = "Maximum"  
UserForm2.Label71.Visible = True  
UserForm2.TextBox34.Visible = True  
UserForm2.TextBox35.Visible = True  
'Mostra botão para alteração das variaveis  
UserForm2.CommandButton21.Visible = True  
UserForm2.Label69.Visible = True
```

Case 6 'Triangular

```
UserForm2.Label70.Caption = "Minimum"  
UserForm2.Label70.Visible = True  
UserForm2.Label71.Caption = "Mode"  
UserForm2.Label71.Visible = True  
UserForm2.Label72.Caption = "Maximum"  
UserForm2.Label72.Visible = True  
UserForm2.TextBox34.Visible = True  
UserForm2.TextBox35.Visible = True  
UserForm2.TextBox36.Visible = True  
'Mostra botão para alteração das variaveis  
UserForm2.CommandButton21.Visible = True  
UserForm2.Label69.Visible = True
```

End Select

End Sub

```
Private Sub ComboBox3_Change()
```

```
'Em caso de alteração da combobox
```

```
Var_Combobox3Alterada = UserForm2.ComboBox3.ListIndex
```

```
Select Case Var_Combobox3Alterada
```

```
Case 1 'Exponencial
```

```
'Esconde as labels e textboxes
```

```
UserForm2.Label91.Visible = False
```

UserForm2.Label92.Visible = False

UserForm2.TextBox45.Visible = False

UserForm2.TextBox46.Visible = False

'Apresentam as labels e textboxes relevantes

UserForm2.Label90.Caption = "Mean"

UserForm2.Label90.Visible = True

UserForm2.TextBox44.Visible = True

'Mostra botão para alteração das variáveis

UserForm2.CommandButton22.Visible = True

UserForm2.Label89.Visible = True

Case 2 'Poisson

'Esconde as labels e textboxes

UserForm2.Label91.Visible = False

UserForm2.Label92.Visible = False

UserForm2.TextBox45.Visible = False

UserForm2.TextBox46.Visible = False

'Apresentam as labels e textboxes relevantes

UserForm2.Label90.Caption = "Mean"

UserForm2.Label90.Visible = True

UserForm2.TextBox44.Visible = True

'Mostra botão para alteração das variáveis

UserForm2.CommandButton22.Visible = True

UserForm2.Label89.Visible = True

Case 3 'Normal

'Esconde as labels e textboxes

UserForm2.Label92.Visible = False

UserForm2.TextBox46.Visible = False

'Apresentam as labels e textboxes relevantes

UserForm2.Label90.Caption = "Mean"

UserForm2.Label90.Visible = True

UserForm2.Label91.Caption = "StdDev"

UserForm2.Label91.Visible = True

UserForm2.TextBox44.Visible = True

UserForm2.TextBox45.Visible = True

'Mostra botão para alteração das variáveis

UserForm2.CommandButton22.Visible = True

UserForm2.Label89.Visible = True

Case 4 'LogNormal

'Esconde as labels e textboxes

UserForm2.Label92.Visible = False

UserForm2.TextBox46.Visible = False

'Apresentam as labels e textboxes relevantes

UserForm2.Label90.Caption = "LogMean"

UserForm2.Label90.Visible = True

UserForm2.Label91.Caption = "LogStd"

UserForm2.Label91.Visible = True

UserForm2.TextBox44.Visible = True

UserForm2.TextBox45.Visible = True

'Mostra botão para alteração das variáveis

UserForm2.CommandButton22.Visible = True

UserForm2.Label89.Visible = True

Case 5 'Uniform

'Esconde as labels e textboxes

UserForm2.Label92.Visible = False

UserForm2.TextBox46.Visible = False

'Apresentam as labels e textboxes relevantes

UserForm2.Label90.Caption = "Minimum"

UserForm2.Label90.Visible = True

UserForm2.Label91.Caption = "Maximum"

UserForm2.Label91.Visible = True

UserForm2.TextBox44.Visible = True

UserForm2.TextBox45.Visible = True

'Mostra botão para alteração das variáveis

UserForm2.CommandButton22.Visible = True

UserForm2.Label89.Visible = True

Case 6 'Triangular

```
UserForm2.Label90.Caption = "Minimum"
UserForm2.Label90.Visible = True
UserForm2.Label91.Caption = "Mode"
UserForm2.Label91.Visible = True
UserForm2.Label92.Caption = "Maximum"
UserForm2.Label92.Visible = True
UserForm2.TextBox44.Visible = True
UserForm2.TextBox45.Visible = True
UserForm2.TextBox46.Visible = True
'Mostra botão para alteração das variaveis
UserForm2.CommandButton22.Visible = True
UserForm2.Label89.Visible = True
End Select
End Sub

Private Sub ComboBox4_Change()
'Em caso de alteração da combobox
Var_Combobox4Alterada = UserForm2.ComboBox4.ListIndex

Select Case Var_Combobox4Alterada
Case 1 'Exponencial
'Esconde as labels e textboxes
UserForm2.Label111.Visible = False
UserForm2.Label112.Visible = False
UserForm2.TextBox55.Visible = False
UserForm2.TextBox56.Visible = False
'Apresentam as labels e textboxes relevantes
UserForm2.Label110.Caption = "Mean"
UserForm2.Label110.Visible = True
UserForm2.TextBox54.Visible = True
'Mostra botão para alteração das variaveis
UserForm2.CommandButton23.Visible = True
UserForm2.Label109.Visible = True
```

Case 2 'Poisson

'Esconde as labels e textboxes que não interessam

UserForm2.Label111.Visible = False

UserForm2.Label112.Visible = False

UserForm2.TextBox55.Visible = False

UserForm2.TextBox56.Visible = False

'Apresentam as labels e textboxes relevantes

UserForm2.Label110.Caption = "Mean"

UserForm2.Label110.Visible = True

UserForm2.TextBox54.Visible = True

'Mostra botão para alteração das variáveis

UserForm2.CommandButton23.Visible = True

UserForm2.Label109.Visible = True

Case 3 'Normal

'Esconde as labels e textboxes

UserForm2.Label112.Visible = False

UserForm2.TextBox56.Visible = False

'Apresentam as labels e textboxes relevantes

UserForm2.Label110.Caption = "Mean"

UserForm2.Label110.Visible = True

UserForm2.Label111.Caption = "StdDev"

UserForm2.Label111.Visible = True

UserForm2.TextBox54.Visible = True

UserForm2.TextBox55.Visible = True

'Mostra botão para alteração das variáveis

UserForm2.CommandButton23.Visible = True

UserForm2.Label109.Visible = True

Case 4 'LogNormal

'Esconde as labels e textboxes

UserForm2.Label112.Visible = False

UserForm2.TextBox56.Visible = False

'Apresentam as labels e textboxes relevantes

UserForm2.Label110.Caption = "LogMean"

```
UserForm2.Label110.Visible = True
UserForm2.Label111.Caption = "LogStd"
UserForm2.Label111.Visible = True
UserForm2.TextBox54.Visible = True
UserForm2.TextBox55.Visible = True
'Mostra botão para alteração das variaveis
UserForm2.CommandButton23.Visible = True
UserForm2.Label109.Visible = True
```

Case 5 'Uniform

```
'Esconde as labels e textboxes
UserForm2.Label112.Visible = False
UserForm2.TextBox56.Visible = False
'Apresentam as labels e textboxes relevantes
UserForm2.Label110.Caption = "Minimum"
UserForm2.Label110.Visible = True
UserForm2.Label111.Caption = "Maximum"
UserForm2.Label111.Visible = True
UserForm2.TextBox54.Visible = True
UserForm2.TextBox55.Visible = True
'Mostra botão para alteração das variaveis
UserForm2.CommandButton23.Visible = True
UserForm2.Label109.Visible = True
```

Case 6 'Triangular

```
UserForm2.Label110.Caption = "Minimum"
UserForm2.Label110.Visible = True
UserForm2.Label111.Caption = "Mode"
UserForm2.Label111.Visible = True
UserForm2.Label112.Caption = "Maximum"
UserForm2.Label112.Visible = True
UserForm2.TextBox54.Visible = True
UserForm2.TextBox55.Visible = True
UserForm2.TextBox56.Visible = True
'Mostra botão para alteração das variaveis
```

```

    UserForm2.CommandButton23.Visible = True

    UserForm2.Label109.Visible = True

End Select

End Sub

Private Sub CommandButton1_Click()
'Carrega para o modelo os valores para a estratégia 1 - aleatorio (default)

Dim m As Model

Dim s As SIMAN

Set m = ThisDocument.Model

Set s = m.SIMAN

s.VariableArrayValue(s.SymbolNumber("S_Strategy")) = 1

End Sub

Private Sub CommandButton11_Click()
'Carrega para o modelo os valores para a estratégia 5 - blocos

Dim m As Model

Dim s As SIMAN

Set m = ThisDocument.Model

Set s = m.SIMAN

s.VariableArrayValue(s.SymbolNumber("S_Strategy")) = 5

End Sub

Private Sub CommandButton13_Click()
'Carrega para o modelo os valores para a estratégia 6 - Steffen

Dim m As Model

Dim s As SIMAN

```

```
Set m = ThisDocument.Model
```

```
Set s = m.SIMAN
```

```
s.VariableArrayValue(s.SymbolNumber("S_Strategy")) = 6
```

```
End Sub
```

```
Private Sub CommandButton17_Click()
```

```
'Botão para leitura dos parametros de simulação
```

```
Dim m As Model
```

```
Dim s As SIMAN
```

```
Set m = ThisDocument.Model
```

```
Set s = m.SIMAN
```

```
' Atenção ao numero de ordem das textboxes
```

```
UserForm2.TextBox99.value = s.InitializeSystemBetweenReplications
```

```
UserForm2.TextBox100.value = s.InitializeStatisticsBetweenReplications
```

```
UserForm2.TextBox102.value = m.NumberOfReplications
```

```
UserForm2.TextBox99.Visible = True
```

```
UserForm2.TextBox100.Visible = True
```

```
UserForm2.TextBox102.Visible = True
```

```
UserForm2.CommandButton18.Visible = True
```

```
' Apresentação dos campos para alterações pré-preenchidos
```

```
If (UserForm2.TextBox99.value = True) Then
```

```
    UserForm2.ComboBox12.ListIndex = 0
```

```
Else
```

```
    UserForm2.ComboBox12.ListIndex = 1
```

End If

UserForm2.ComboBox12.Visible = True

If (UserForm2.TextBox100.value = True) Then

 UserForm2.ComboBox13.ListIndex = 0

 Else

 UserForm2.ComboBox13.ListIndex = 1

End If

UserForm2.ComboBox13.Visible = True

UserForm2.TextBox107.value = UserForm2.TextBox102.value

UserForm2.TextBox107.Visible = True

End Sub

Private Sub CommandButton18_Click()

'Botão de alteração dos parametros da simulação

'Dimensionamento objetos gerais do Arena/SIMAN

Dim m As Model

Dim s As SIMAN

'Atribui objetos às variáveis

Set m = ThisDocument.Model

Set s = m.SIMAN

Var_ComboBox12Index = UserForm2.ComboBox12.ListIndex

Select Case Var_ComboBox12Index

 Case 0

 s.InitializeSystemBetweenReplications = True

 UserForm2.TextBox99.value = "TRUE"

 Case 1

```

        s.InitializeSystemBetweenReplications = False
        UserForm2.TextBox99.value = "FALSE"
    End Select

    Var_Combobox13Index = UserForm2.Combobox13.ListIndex
    Select Case Var_Combobox13Index
        Case 0
            s.InitializeStatisticsBetweenReplications = True
            UserForm2.TextBox100.value = "TRUE"
        Case 1
            s.InitializeStatisticsBetweenReplications = False
            UserForm2.TextBox100.value = "FALSE"
    End Select

    If UserForm2.TextBox107.value <> "" Then
        Var_TempTextBox107 = CStr(UserForm2.TextBox107.value)
        'm.NumberOfReplications = Var_TempTextBox107
        s.RunMaximumReplications = Var_TempTextBox107
        UserForm2.TextBox102.value = Var_TempTextBox107
    Else
        MsgBox "Please insert values in all text boxes"
    End If

End Sub

Private Sub CommandButton2_Click()
'Carrega para o modelo os valores para a estratégia 2 - tras para frente
Dim m As Model
Dim s As SIMAN

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

```

```
s.VariableArrayValue(s.SymbolNumber("S_Strategy")) = 2
```

```
End Sub
```

```
Private Sub CommandButton20_Click()
```

```
'Botão de alteração das variáveis de Passagers Arrivals
```

```
'Dimensionamento objetos gerais do Arena/SIMAN
```

```
Dim m As Model
```

```
Dim s As SIMAN
```

```
'Atribui objetos às variáveis
```

```
Set m = ThisDocument.Model
```

```
Set s = m.SIMAN
```

```
Var_Combobox1Index = UserForm2.Combobox1.ListIndex
```

```
Select Case Var_Combobox1Index
```

```
Case 1 'Selecionada distribuição Exponencial
```

```
'Carrega valores das textboxes pra variáveis
```

```
Var_TempTextBox24 = Val(UserForm2.TextBox24.value) 'Média da distribuição escolhida
```

```
'Valida existência de valores e em caso afirmativo procede à alteração
```

```
If Var_TempTextBox24 <> 0 Then
```

```
'Cria string para substituição da expressão e carrega valores nas variáveis
```

```
Var_TempConstructGR = "Expo" & "(" & Var_TempTextBox24 & ")"
```

```
s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity")) = Var_TempConstructGR
```

```
Else
```

```
MsgBox "Please insert values in all text boxes"
```

```
End If
```

```
Case 2 'Selecionada distribuição de Poisson
```

```
'Carrega valores das textboxes pra variáveis
```

```
Var_TempTextBox24 = Val(UserForm2.TextBox24.value) 'Média da distribuição escolhida
```

```

If Var_TempTextBox24 <> 0 Then
    Var_TempConstructGR = "Pois" & "(" & Var_TempTextBox24 & ")"
    s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity")) = Var_TempConstructGR
Else
    MsgBox "Please insert values in all text boxes"
End If

Case 3 'Selecionada distribuição Normal
'Carrega valores das textboxes pra variaveis
Var_TempTextBox24 = Val(UserForm2.TextBox24.value) 'Média da distribuição escolhida
Var_TempTextBox25 = Val(UserForm2.TextBox25.value) 'Desvio standard
If Var_TempTextBox24 <> 0 And Var_TempTextBox25 <> 0 Then
    Var_TempConstructGR = "Norm" & "(" & Var_TempTextBox24 & "," & Var_TempTextBox25 & ")"
    s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity")) = Var_TempConstructGR
Else
    MsgBox "Please insert values in all text boxes"
End If

Case 4 'Selecionada distribuição LogNormal
'Carrega valores das textboxes pra variaveis
Var_TempTextBox24 = Val(UserForm2.TextBox24.value) 'LogMédia da distribuição escolhida
Var_TempTextBox25 = Val(UserForm2.TextBox25.value) 'Log Desvio standard
If Var_TempTextBox24 <> 0 And Var_TempTextBox25 <> 0 Then
    Var_TempConstructGR = "Logn" & "(" & Var_TempTextBox24 & "," & Var_TempTextBox25 & ")"
    s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity")) = Var_TempConstructGR
Else
    MsgBox "Please insert values in all text boxes"
End If

Case 5 'Selecionada distribuição Uniform
'Carrega valores das textboxes pra variaveis
Var_TempTextBox24 = Val(UserForm2.TextBox24.value) 'Minimum
Var_TempTextBox25 = Val(UserForm2.TextBox25.value) 'Maximum
If Var_TempTextBox24 <> 0 And Var_TempTextBox25 <> 0 Then
    'Verifica se o valor minimo é inferior ao máximo
    If (Var_TempTextBox24 < Var_TempTextBox25) Then

```

```

        Var_TempConstructGR = "Unif" & "(" & Var_TempTextBox24 & "," & Var_TempTextBox25 & ")"
        s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity")) =
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 'Selecionada distribuição Triangular
'Carrega valores das textboxes pra variaveis
Var_TempTextBox24 = Val(UserForm2.TextBox24.value) 'Minimum
Var_TempTextBox25 = Val(UserForm2.TextBox25.value) 'Mode
Var_TempTextBox26 = Val(UserForm2.TextBox26.value) 'Maximum
If Var_TempTextBox24 <> 0 And Var_TempTextBox25 <> 0 And Var_TempTextBox26 <> 0 Then
    If (Var_TempTextBox24 < Var_TempTextBox25) And (Var_TempTextBox25 <
Var_TempTextBox26) Then
        Var_TempConstructGR = "Tria" & "(" & Var_TempTextBox24 & "," & Var_TempTextBox25 & ","
& Var_TempTextBox26 & ")"
        s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity")) =
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

'Carregamento da expressão da distribuição de tempo
Var_ExpressionGR = s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Entity"))
Var_DistributionGR = Left(Var_ExpressionGR, 4)

```

'Tratamento e apresentação da distribuição

Select Case Var_DistributionGR

Case "Expo", "EXPO"

'Esconde as labels e textboxes

UserForm2.Label53.Visible = False

UserForm2.Label54.Visible = False

UserForm2.TextBox22.Visible = False

UserForm2.TextBox23.Visible = False

'Trata dados e carrega-os para a form para apresentação

UserForm2.TextBox20.value = "Exponencial"

UserForm2.TextBox21.value = Val(Mid(Var_ExpressionGR, 6))

'Apresenta os campos com os valores carregados

UserForm2.Label52.Caption = "Mean"

UserForm2.Label52.Visible = True

UserForm2.TextBox20.Visible = True

UserForm2.TextBox21.Visible = True

Case "Pois", "POIS"

'Esconde as labels e textboxes

UserForm2.Label53.Visible = False

UserForm2.Label54.Visible = False

UserForm2.TextBox22.Visible = False

UserForm2.TextBox23.Visible = False

'Trata dados e carrega-os para a form para apresentação

UserForm2.TextBox20.value = "Poisson"

UserForm2.TextBox21.value = Val(Mid(Var_ExpressionGR, 6))

'Apresenta os campos com os valores carregados

UserForm2.Label52.Caption = "Mean"

UserForm2.Label52.Visible = True

UserForm2.TextBox20.Visible = True

UserForm2.TextBox21.Visible = True

Case "Norm", "NORM"

'Esconde as labels e textboxes

```
UserForm2.Label54.Visible = False
UserForm2.TextBox23.Visible = False
'Trata dados e carrega-os para a form para apresentação
UserForm2.TextBox20.value = "Normal"
UserForm2.TextBox21.value = Val(Mid(Var_ExpressionGR, 6))
UserForm2.TextBox22.value = Val(Mid(Var_ExpressionGR, InStr(Var_ExpressionGR, ",") + 1))
'Apresenta os campos com os valores carregados
UserForm2.Label52.Caption = "Mean"
UserForm2.Label52.Visible = True
UserForm2.Label53.Caption = "StdDev"
UserForm2.Label53.Visible = True
UserForm2.TextBox20.Visible = True
UserForm2.TextBox21.Visible = True
UserForm2.TextBox22.Visible = True
Case "Logn", "LOGN"
'Esconde as labels e textboxes
UserForm2.Label54.Visible = False
UserForm2.TextBox23.Visible = False
'Trata dados e carrega-os para a form para apresentação
UserForm2.TextBox20.value = "LogNormal"
UserForm2.TextBox21.value = Val(Mid(Var_ExpressionGR, 6))
UserForm2.TextBox22.value = Val(Mid(Var_ExpressionGR, InStr(Var_ExpressionGR, ",") + 1))
'Apresenta os campos com os valores carregados
UserForm2.Label52.Caption = "LogMean"
UserForm2.Label52.Visible = True
UserForm2.Label53.Caption = "LogStd"
UserForm2.Label53.Visible = True
UserForm2.TextBox20.Visible = True
UserForm2.TextBox21.Visible = True
UserForm2.TextBox22.Visible = True
Case "Unif", "UNIF"
'Esconde as labels e textboxes
UserForm2.Label54.Visible = False
```

```
UserForm2.TextBox23.Visible = False

'Trata dados e carrega-os para a form para apresentação
UserForm2.TextBox20.value = "Uniform"
UserForm2.TextBox21.value = Val(Mid(Var_ExpressionGR, 6))
UserForm2.TextBox22.value = Val(Mid(Var_ExpressionGR, InStr(Var_ExpressionGR, ",") + 1))

'Apresenta os campos com os valores carregados
UserForm2.Label52.Caption = "Minimum"
UserForm2.Label52.Visible = True
UserForm2.Label53.Caption = "Maximum"
UserForm2.Label53.Visible = True
UserForm2.TextBox20.Visible = True
UserForm2.TextBox21.Visible = True
UserForm2.TextBox22.Visible = True

Case "Tria", "TRIA"

'Trata dados e carrega-os para a form para apresentação
UserForm2.TextBox20.value = "Triangular"
UserForm2.TextBox21.value = Val(Mid(Var_ExpressionGR, 6))
UserForm2.TextBox22.value = Val(Mid(Var_ExpressionGR, InStr(Var_ExpressionGR, ",") + 1))
UserForm2.TextBox23.value = Val(Mid(Var_ExpressionGR, InStr(InStr(Var_ExpressionGR, ",") + 1,
Var_ExpressionGR, ",") + 1))

'Apresenta os campos com os valores carregados
UserForm2.Label52.Caption = "Minimum"
UserForm2.Label52.Visible = True
UserForm2.Label53.Caption = "Mode"
UserForm2.Label53.Visible = True
UserForm2.Label54.Caption = "Maximum"
UserForm2.Label54.Visible = True
UserForm2.TextBox20.Visible = True
UserForm2.TextBox21.Visible = True
UserForm2.TextBox22.Visible = True
UserForm2.TextBox23.Visible = True

End Select
```

End Sub

Private Sub CommandButton22_Click()

'Botão de alteração das variáveis de Hand Luggage

'Dimensionamento objetos gerais do Arena/SIMAN

Dim m As Model

Dim s As SIMAN

'Atribui objetos às variáveis

Set m = ThisDocument.Model

Set s = m.SIMAN

Var_ComboBox3Index = UserForm2.ComboBox3.ListIndex

Select Case Var_ComboBox3Index

Case 1 'Selecionada distribuição Exponencial

'Carrega valores das textboxes pra variáveis

Var_TempTextBox44 = Val(UserForm2.TextBox44.value) 'Média da distribuição escolhida

If Var_TempTextBox44 <> 0 Then

Var_TempConstructGR = "Expo" & "(" & Var_TempTextBox44 & ")"

s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Luggage")) =
Var_TempConstructGR

Else

MsgBox "Please insert values in all text boxes"

End If

Case 2 'Selecionada distribuição de Poisson

'Carrega valores das textboxes pra variáveis

Var_TempTextBox44 = Val(UserForm2.TextBox44.value) 'Média da distribuição escolhida

If Var_TempTextBox44 <> 0 Then

Var_TempConstructGR = "Pois" & "(" & Var_TempTextBox44 & ")"

s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Luggage")) =
Var_TempConstructGR

```

Else
    MsgBox "Please insert values in all text boxes"
End If

Case 3 'Selecionada distribuição Normal
    'Carrega valores das textboxes pra variaveis
    Var_TempTextBox44 = Val(UserForm2.TextBox44.value) 'Média da distribuição escolhida
    Var_TempTextBox45 = Val(UserForm2.TextBox45.value) 'Desvio standard
    If Var_TempTextBox44 <> 0 And Var_TempTextBox45 <> 0 Then
        Var_TempConstructGR = "Norm" & "(" & Var_TempTextBox44 & "," & Var_TempTextBox45 & ")"
        s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Luggage")) =
Var_TempConstructGR
    Else
        MsgBox "Please insert values in all text boxes"
    End If

Case 4 'Selecionada distribuição LogNormal
    'Carrega valores das textboxes pra variaveis
    Var_TempTextBox44 = Val(UserForm2.TextBox44.value) 'LogMédia da distribuição escolhida
    Var_TempTextBox45 = Val(UserForm2.TextBox45.value) 'Log Desvio standard
    If Var_TempTextBox44 <> 0 And Var_TempTextBox45 <> 0 Then
        Var_TempConstructGR = "Logn" & "(" & Var_TempTextBox44 & "," & Var_TempTextBox45 & ")"
        s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Velocity_Entity")) =
Var_TempConstructGR
    Else
        MsgBox "Please insert values in all text boxes"
    End If

Case 5 'Selecionada distribuição Uniform
    'Carrega valores das textboxes pra variaveis
    Var_TempTextBox44 = Val(UserForm2.TextBox44.value) 'Minimum
    Var_TempTextBox45 = Val(UserForm2.TextBox45.value) 'Maximum
    If Var_TempTextBox44 <> 0 And Var_TempTextBox45 <> 0 Then
        'Verifica se o valor minimo é inferior ao máximo
        If (Var_TempTextBox44 < Var_TempTextBox45) Then
            Var_TempConstructGR = "Unif" & "(" & Var_TempTextBox44 & "," & Var_TempTextBox45 & ")"

```

```

        s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Luggage")) =
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 'Selecionada distribuição Triangular
    'Carrega valores das textboxes pra variaveis
    Var_TempTextBox44 = Val(UserForm2.TextBox44.value) 'Minimum
    Var_TempTextBox45 = Val(UserForm2.TextBox45.value) 'Mode
    Var_TempTextBox46 = Val(UserForm2.TextBox46.value) 'Maximum
    If Var_TempTextBox44 <> 0 And Var_TempTextBox45 <> 0 And Var_TempTextBox46 <> 0 Then
        'Verifica se o valor minimo é inferior ao medio e por sua vez ao maximo
        If (Var_TempTextBox44 < Var_TempTextBox45) And (Var_TempTextBox45 <
Var_TempTextBox46) Then
            Var_TempConstructGR = "Tria" & "(" & Var_TempTextBox44 & "," & Var_TempTextBox45 & ","
& Var_TempTextBox46 & ")"
            s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Luggage")) =
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

'Carregamento da expressão da distribuição de tempo
Var_ExpressionGR = s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Luggage"))
Var_DistributionGR = Left(Var_ExpressionGR, 4)

```

'Tratamento e apresentação da distribuição

Select Case Var_DistributionGR

Case "Expo", "EXPO"

'Esconde as labels e textboxes

UserForm2.Label85.Visible = False

UserForm2.Label86.Visible = False

UserForm2.TextBox42.Visible = False

UserForm2.TextBox43.Visible = False

'Trata dados e carrega-os para a form para apresentação

UserForm2.TextBox40.value = "Exponencial"

UserForm2.TextBox41.value = Val(Mid(Var_ExpressionGR, 6))

'Apresenta os campos com os valores carregados

UserForm2.Label84.Caption = "Mean"

UserForm2.Label84.Visible = True

UserForm2.TextBox40.Visible = True

UserForm2.TextBox41.Visible = True

Case "Pois", "POIS"

'Esconde as labels e textboxes

UserForm2.Label85.Visible = False

UserForm2.Label86.Visible = False

UserForm2.TextBox42.Visible = False

UserForm2.TextBox43.Visible = False

'Trata dados e carrega-os para a form para apresentação

UserForm2.TextBox40.value = "Poisson"

UserForm2.TextBox41.value = Val(Mid(Var_ExpressionGR, 6))

'Apresenta os campos com os valores carregados

UserForm2.Label84.Caption = "Mean"

UserForm2.Label84.Visible = True

UserForm2.TextBox40.Visible = True

UserForm2.TextBox41.Visible = True

Case "Norm", "NORM"

'Esconde as labels e textboxes

```
UserForm2.Label86.Visible = False
UserForm2.TextBox43.Visible = False
'Trata dados e carrega-os para a form para apresentação
UserForm2.TextBox40.value = "Normal"
UserForm2.TextBox41.value = Val(Mid(Var_ExpressionGR, 6))
UserForm2.TextBox42.value = Val(Mid(Var_ExpressionGR, InStr(Var_ExpressionGR, ",") + 1))
'Apresenta os campos com os valores carregados
UserForm2.Label84.Caption = "Mean"
UserForm2.Label84.Visible = True
UserForm2.Label85.Caption = "StdDev"
UserForm2.Label85.Visible = True
UserForm2.TextBox40.Visible = True
UserForm2.TextBox41.Visible = True
UserForm2.TextBox42.Visible = True
Case "Logn", "LOGN"
'Esconde as labels e textboxes
UserForm2.Label86.Visible = False
UserForm2.TextBox43.Visible = False
'Trata dados e carrega-os para a form para apresentação
UserForm2.TextBox40.value = "LogNormal"
UserForm2.TextBox41.value = Val(Mid(Var_ExpressionGR, 6))
UserForm2.TextBox42.value = Val(Mid(Var_ExpressionGR, InStr(Var_ExpressionGR, ",") + 1))
'Apresenta os campos com os valores carregados
UserForm2.Label84.Caption = "LogMean"
UserForm2.Label84.Visible = True
UserForm2.Label85.Caption = "LogStd"
UserForm2.Label85.Visible = True
UserForm2.TextBox40.Visible = True
UserForm2.TextBox41.Visible = True
UserForm2.TextBox42.Visible = True
Case "Unif", "UNIF"
'Esconde as labels e textboxes
UserForm2.Label86.Visible = False
```

```
UserForm2.TextBox43.Visible = False

'Trata dados e carrega-os para a form para apresentação
UserForm2.TextBox40.value = "Uniform"
UserForm2.TextBox41.value = Val(Mid(Var_ExpressionGR, 6))
UserForm2.TextBox42.value = Val(Mid(Var_ExpressionGR, InStr(Var_ExpressionGR, ",") + 1))

'Apresenta os campos com os valores carregados
UserForm2.Label84.Caption = "Minimum"
UserForm2.Label84.Visible = True
UserForm2.Label85.Caption = "Maximum"
UserForm2.Label85.Visible = True
UserForm2.TextBox40.Visible = True
UserForm2.TextBox41.Visible = True
UserForm2.TextBox42.Visible = True

Case "Tria", "TRIA"

'Trata dados e carrega-os para a form para apresentação
UserForm2.TextBox40.value = "Triangular"
UserForm2.TextBox41.value = Val(Mid(Var_ExpressionGR, 6))
UserForm2.TextBox42.value = Val(Mid(Var_ExpressionGR, InStr(Var_ExpressionGR, ",") + 1))
UserForm2.TextBox43.value = Val(Mid(Var_ExpressionGR, InStr(InStr(Var_ExpressionGR, ",") + 1,
Var_ExpressionGR, ",") + 1))

'Apresenta os campos com os valores carregados
UserForm2.Label84.Caption = "Minimum"
UserForm2.Label84.Visible = True
UserForm2.Label85.Caption = "Mode"
UserForm2.Label85.Visible = True
UserForm2.Label86.Caption = "Maximum"
UserForm2.Label86.Visible = True
UserForm2.TextBox40.Visible = True
UserForm2.TextBox41.Visible = True
UserForm2.TextBox42.Visible = True
UserForm2.TextBox43.Visible = True

End Select

End Sub
```

```
Private Sub CommandButton23_Click()
```

```
'Botão de alteração das variáveis de Hand Luggage
```

```
'Dimensionamento objetos gerais do Arena/SIMAN
```

```
Dim m As Model
```

```
Dim s As SIMAN
```

```
'Atribui objetos às variáveis
```

```
Set m = ThisDocument.Model
```

```
Set s = m.SIMAN
```

```
Var_ComboBox4Index = UserForm2.ComboBox4.ListIndex
```

```
Select Case Var_ComboBox4Index
```

```
Case 1 'Selecionada distribuição Exponencial
```

```
'Carrega valores das textboxes pra variáveis
```

```
Var_TempTextBox54 = Val(UserForm2.TextBox54.value) 'Média da distribuição escolhida
```

```
If Var_TempTextBox54 <> 0 Then
```

```
Var_TempConstructGR = "Expo" & "(" & Var_TempTextBox54 & ")"
```

```
s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Seat")) = Var_TempConstructGR
```

```
Else
```

```
MsgBox "Please insert values in all text boxes"
```

```
End If
```

```
Case 2 'Selecionada distribuição de Poisson
```

```
'Carrega valores das textboxes pra variáveis
```

```
Var_TempTextBox54 = Val(UserForm2.TextBox54.value) 'Média da distribuição escolhida
```

```
If Var_TempTextBox54 <> 0 Then
```

```
Var_TempConstructGR = "Pois" & "(" & Var_TempTextBox54 & ")"
```

```
s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Seat")) = Var_TempConstructGR
```

```
Else
```

```
MsgBox "Please insert values in all text boxes"
```

```
End If
```

Case 3 'Selecionada distribuição Normal

'Carrega valores das textboxes pra variaveis

Var_TempTextBox54 = Val(UserForm2.TextBox54.value) 'Média da distribuição escolhida

Var_TempTextBox55 = Val(UserForm2.TextBox55.value) 'Desvio standard

If Var_TempTextBox54 <> 0 And Var_TempTextBox55 <> 0 Then

Var_TempConstructGR = "Norm" & "(" & Var_TempTextBox54 & "," & Var_TempTextBox55 & ")"

s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Seat")) = Var_TempConstructGR

Else

MsgBox "Please insert values in all text boxes"

End If

Case 4 'Selecionada distribuição LogNormal

'Carrega valores das textboxes pra variaveis

Var_TempTextBox54 = Val(UserForm2.TextBox54.value) 'LogMédia da distribuição escolhida

Var_TempTextBox55 = Val(UserForm2.TextBox55.value) 'Log Desvio standard

If Var_TempTextBox54 <> 0 And Var_TempTextBox55 <> 0 Then

Var_TempConstructGR = "Logn" & "(" & Var_TempTextBox54 & "," & Var_TempTextBox55 & ")"

s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Seat")) = Var_TempConstructGR

Else

MsgBox "Please insert values in all text boxes"

End If

Case 5 'Selecionada distribuição Uniform

'Carrega valores das textboxes pra variaveis

Var_TempTextBox54 = Val(UserForm2.TextBox54.value) 'Minimum

Var_TempTextBox55 = Val(UserForm2.TextBox55.value) 'Maximum

If Var_TempTextBox54 <> 0 And Var_TempTextBox55 <> 0 Then

'Verifica se o valor minimo é inferior ao máximo

If (Var_TempTextBox54 < Var_TempTextBox55) Then

Var_TempConstructGR = "Unif" & "(" & Var_TempTextBox54 & "," & Var_TempTextBox55 & ")"

s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Seat")) = 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 'Selecionada distribuição Triangular

    'Carrega valores das textboxes pra variaveis

    Var_TempTextBox54 = Val(UserForm2.TextBox54.value) 'Minimum

    Var_TempTextBox55 = Val(UserForm2.TextBox55.value) 'Mode

    Var_TempTextBox56 = Val(UserForm2.TextBox56.value) 'Maximum

    If Var_TempTextBox54 <> 0 And Var_TempTextBox55 <> 0 And Var_TempTextBox56 <> 0 Then

        'Verifica se o valor minimo é inferior ao medio e por sua vez ao maximo

        If (Var_TempTextBox54 < Var_TempTextBox55) And (Var_TempTextBox55 <
Var_TempTextBox56) Then

            Var_TempConstructGR = "Tria" & "(" & Var_TempTextBox54 & "," & Var_TempTextBox55 & ","
& Var_TempTextBox56 & ")"

            s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Seat")) = 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

```

'Carregamento da expressão da distribuição de tempo

```
Var_ExpressionGR = s.VariableArrayValueAsVariant(s.SymbolNumber("Alter_Time_Seat"))
```

```
Var_DistributionGR = Left(Var_ExpressionGR, 4)
```

'Tratamento e apresentação da distribuição

```
Select Case Var_DistributionGR
```

```
Case "Expo", "EXPO"
```

```
'Esconde as labels e textboxes
```

```
UserForm2.Label105.Visible = False
```

```
UserForm2.Label106.Visible = False
UserForm2.TextBox52.Visible = False
UserForm2.TextBox53.Visible = False
'Trata dados e carrega-os para a form para apresentação
UserForm2.TextBox50.value = "Exponencial"
UserForm2.TextBox51.value = Val(Mid(Var_ExpressionGR, 6))
'Apresenta os campos com os valores carregados
UserForm2.Label104.Caption = "Mean"
UserForm2.Label104.Visible = True
UserForm2.TextBox50.Visible = True
UserForm2.TextBox51.Visible = True
Case "Pois", "POIS"
'Esconde as labels e textboxes
UserForm2.Label105.Visible = False
UserForm2.Label106.Visible = False
UserForm2.TextBox52.Visible = False
UserForm2.TextBox53.Visible = False
'Trata dados e carrega-os para a form para apresentação
UserForm2.TextBox50.value = "Poisson"
UserForm2.TextBox51.value = Val(Mid(Var_ExpressionGR, 6))
'Apresenta os campos com os valores carregados
UserForm2.Label104.Caption = "Mean"
UserForm2.Label104.Visible = True
UserForm2.TextBox50.Visible = True
UserForm2.TextBox51.Visible = True
Case "Norm", "NORM"
'Esconde as labels e textboxes
UserForm2.Label106.Visible = False
UserForm2.TextBox53.Visible = False
'Trata dados e carrega-os para a form para apresentação
UserForm2.TextBox50.value = "Normal"
UserForm2.TextBox51.value = Val(Mid(Var_ExpressionGR, 6))
UserForm2.TextBox52.value = Val(Mid(Var_ExpressionGR, InStr(Var_ExpressionGR, ",") + 1))
```

```
'Apresenta os campos com os valores carregados
UserForm2.Label104.Caption = "Mean"
UserForm2.Label104.Visible = True
UserForm2.Label105.Caption = "StdDev"
UserForm2.Label105.Visible = True
UserForm2.TextBox50.Visible = True
UserForm2.TextBox51.Visible = True
UserForm2.TextBox52.Visible = True
Case "Logn", "LOGN"
'Esconde as labels e textboxes
UserForm2.Label106.Visible = False
UserForm2.TextBox53.Visible = False
'Trata dados e carrega-os para a form para apresentação
UserForm2.TextBox50.value = "LogNormal"
UserForm2.TextBox51.value = Val(Mid(Var_ExpressionGR, 6))
UserForm2.TextBox52.value = Val(Mid(Var_ExpressionGR, InStr(Var_ExpressionGR, ",") + 1))
'Apresenta os campos com os valores carregados
UserForm2.Label104.Caption = "LogMean"
UserForm2.Label104.Visible = True
UserForm2.Label105.Caption = "LogStd"
UserForm2.Label105.Visible = True
UserForm2.TextBox50.Visible = True
UserForm2.TextBox51.Visible = True
UserForm2.TextBox52.Visible = True
Case "Unif", "UNIF"
'Esconde as labels e textboxes
UserForm2.Label106.Visible = False
UserForm2.TextBox53.Visible = False
'Trata dados e carrega-os para a form para apresentação
UserForm2.TextBox50.value = "Uniform"
UserForm2.TextBox51.value = Val(Mid(Var_ExpressionGR, 6))
UserForm2.TextBox52.value = Val(Mid(Var_ExpressionGR, InStr(Var_ExpressionGR, ",") + 1))
'Apresenta os campos com os valores carregados
```

```

UserForm2.Label104.Caption = "Minimum"

UserForm2.Label104.Visible = True

UserForm2.Label105.Caption = "Maximum"

UserForm2.Label85.Visible = True

UserForm2.TextBox50.Visible = True

UserForm2.TextBox51.Visible = True

UserForm2.TextBox52.Visible = True

Case "Tria", "TRIA"

'Trata dados e carrega-os para a form para apresentação

UserForm2.TextBox50.value = "Triangular"

UserForm2.TextBox51.value = Val(Mid(Var_ExpressionGR, 6))

UserForm2.TextBox52.value = Val(Mid(Var_ExpressionGR, InStr(Var_ExpressionGR, ",") + 1))

UserForm2.TextBox53.value = Val(Mid(Var_ExpressionGR, InStr(InStr(Var_ExpressionGR, ",") + 1,
Var_ExpressionGR, ",") + 1))

'Apresenta os campos com os valores carregados

UserForm2.Label104.Caption = "Minimum"

UserForm2.Label104.Visible = True

UserForm2.Label105.Caption = "Mode"

UserForm2.Label105.Visible = True

UserForm2.Label106.Caption = "Maximum"

UserForm2.Label106.Visible = True

UserForm2.TextBox50.Visible = True

UserForm2.TextBox51.Visible = True

UserForm2.TextBox52.Visible = True

UserForm2.TextBox53.Visible = True

End Select

End Sub

Private Sub CommandButton26_Click()

'Carrega para o modelo os valores para a estratégia 6 - Steffen

Dim m As Model

Dim s As SIMAN

```

```
Set m = ThisDocument.Model
```

```
Set s = m.SIMAN
```

```
s.VariableArrayValue(s.SymbolNumber("S_Strategy")) = 7
```

```
End Sub
```

```
Private Sub CommandButton5_Click()
```

```
'Carrega para o modelo os valores para a estratégia 3 - fora para dentro
```

```
Dim m As Model
```

```
Dim s As SIMAN
```

```
Set m = ThisDocument.Model
```

```
Set s = m.SIMAN
```

```
s.VariableArrayValue(s.SymbolNumber("S_Strategy")) = 3
```

```
End Sub
```

```
Private Sub CommandButton7_Click()
```

```
Set m = ThisDocument.Model
```

```
Set s = m.SIMAN
```

```
Dim flag As Boolean
```

```
flag = False
```

```
UserForm3.Label10.Caption = s.VariableArrayValue(s.SymbolNumber("S_Strategy"))
```

```
'luggage
```

```
Dim luggage_per As Integer
```

```
luggage_per = UserForm2.TextBox2.value
```

```

If luggage_per > 100 Then
    MsgBox "Please review percentage of passagers with luggage"
    flag = True
End If

's.VariableArrayValue(s.SymbolNumber("hasluggage")) = luggage_per

'arrival
Dim arrival As Integer 'Number of passagers
Dim pmax As Integer 'Number max of passagers
pmax = 156
arrival = UserForm2.TextBox1.value

If arrival <> 0 Then
    If arrival > pmax Then
        MsgBox "Please review Maximum value of passagers"
        flag = True
    End If
Else
    MsgBox "Please insert values in all text boxes"
    flag = True
End If

'-----Check if has priority-----
If UserForm2.OptionButton1.value Then 'If Yes

    Dim prioritynum As Integer
    prioritynum = UserForm2.TextBox8.value

    If prioritynum <> 0 Then
        If prioritynum > arrival Then
            MsgBox "Please review Maximum value of passagers with priority"
            flag = True
        End If
    End If
End If

```

```
        End If
    Else
        MsgBox "Please insert values in all text boxes"
        flag = True
    End If
End If

If flag = False Then
    UserForm2.Hide
End If

End Sub

Private Sub CommandButton8_Click()

    UserForm2.Hide

End Sub

Private Sub CommandButton9_Click()
'Carrega para o modelo os valores para a estratégia 4 - piramide reversa
Dim m As Model
Dim s As SIMAN

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

s.VariableArrayValue(s.SymbolNumber("S_Strategy")) = 4

End Sub
```

```
Private Sub MultiPage1_Change()
```

```
'
```

```
' Boarding Parameters
```

```
'
```

```
'Esconde botões
```

```
UserForm2.CommandButton20.Visible = False
```

```
UserForm2.CommandButton22.Visible = False
```

```
UserForm2.CommandButton23.Visible = False
```

```
'Verifica/criar listas para as comboboxs dos Passagers Arrivals
```

```
'Marcador de combobox1 já criada
```

```
Var_Combobox1 = UserForm2.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

'Verifica/criar listas para as comboboxs dos Hand Luggage

'Marcador de combobox3 já criada

Var_Combobox3 = UserForm2.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

'Verifica/criar listas para as comboboxs do Seat Interference

'Marcador de combobox4 já criada

Var_Combobox4 = UserForm2.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

'Cria lista de funções a aparecer nas ComBoxes dos Passagers Arrivals

' Expo(Mean)

' Pois(Mean)

' Norm(Mean,StdDev)

' Logn(LogMean,LogStd)

' Unif(Min,Max)

' Tria(Min,Mode,Max)

If Var_ComboBox1Existe = 0 Then

UserForm2.ComboBox1.AddItem "" 'ListIndex = 0

UserForm2.ComboBox1.AddItem "Exponencial" 'ListIndex = 1

UserForm2.ComboBox1.AddItem "Poisson" 'ListIndex = 2

UserForm2.ComboBox1.AddItem "Normal" 'ListIndex = 3

UserForm2.ComboBox1.AddItem "LogNormal" 'ListIndex = 4

```
UserForm2.ComboBox1.AddItem "Uniform" 'ListIndex = 5
UserForm2.ComboBox1.AddItem "Triangular" 'ListIndex = 6
'Set combo box to first entry
UserForm2.ComboBox1.ListIndex = 0
Var_ComboBox1Existe = 1
End If
```

'Cria lista de funções a aparecer nas ComBoxes dos Hand Luggage

```
' Expo(Mean)
' Pois(Mean)
' Norm(Mean,StdDev)
' Logn(LogMean,LogStd)
' Unif(Min,Max)
' Tria(Min,Mode,Max)
```

If Var_ComboBox3Existe = 0 Then

```
UserForm2.ComboBox3.AddItem "" 'ListIndex = 0
UserForm2.ComboBox3.AddItem "Exponencial" 'ListIndex = 1
UserForm2.ComboBox3.AddItem "Poisson" 'ListIndex = 2
UserForm2.ComboBox3.AddItem "Normal" 'ListIndex = 3
UserForm2.ComboBox3.AddItem "LogNormal" 'ListIndex = 4
UserForm2.ComboBox3.AddItem "Uniform" 'ListIndex = 5
UserForm2.ComboBox3.AddItem "Triangular" 'ListIndex = 6
'Set combo box to first entry
UserForm2.ComboBox3.ListIndex = 0
Var_ComboBox3Existe = 1
```

End If

'Cria lista de funções a aparecer nas ComBoxes do Seat Inteference

```
' Expo(Mean)
' Pois(Mean)
' Norm(Mean,StdDev)
' Logn(LogMean,LogStd)
' Unif(Min,Max)
```

```
' Tria(Min,Mode,Max)
If Var_Combobox4Existe = 0 Then
    UserForm2.ComboBox4.AddItem "" 'ListIndex = 0
    UserForm2.ComboBox4.AddItem "Exponencial" 'ListIndex = 1
    UserForm2.ComboBox4.AddItem "Poisson" 'ListIndex = 2
    UserForm2.ComboBox4.AddItem "Normal" 'ListIndex = 3
    UserForm2.ComboBox4.AddItem "LogNormal" 'ListIndex = 4
    UserForm2.ComboBox4.AddItem "Uniform" 'ListIndex = 5
    UserForm2.ComboBox4.AddItem "Triangular" 'ListIndex = 6
    'Set combo box to first entry
    UserForm2.ComboBox4.ListIndex = 0
    Var_Combobox4Existe = 1
End If

'
```

```
' Simulation Parameters
'
```

```
' Esconde caixas dos futuros valores
UserForm2.TextBox107.Visible = False
UserForm2.ComboBox12.Visible = False
UserForm2.ComboBox13.Visible = False
```

```
' Esconde botão de comando
UserForm2.CommandButton18.Visible = False
```

```
'Verifica existencia de listas das comboboxs
'Marcador de combobox12 já criada (apenas true e false)
Var_Combobox12 = UserForm2.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
```

'Marcador de combobox13 já criada (apenas true e false)

```
Var_ComboBox13 = UserForm2.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
```

'Cria lista de opções a aparecer nas ComBoxes dos 'Simulation Parameters

```
If Var_ComboBox12Existe = 0 Then
```

```
    UserForm2.ComboBox12.AddItem "True" 'ListIndex = 0
```

```
    UserForm2.ComboBox12.AddItem "False" 'ListIndex = 1
```

```
    'Set combo box to first entry
```

```
    UserForm2.ComboBox12.ListIndex = 0
```

```
    Var_ComboBox12Existe = 1
```

```
End If
```

```
If Var_ComboBox13Existe = 0 Then
```

```
    UserForm2.ComboBox13.AddItem "True" 'ListIndex = 0
```

```
    UserForm2.ComboBox13.AddItem "False" 'ListIndex = 1
```

```
    'Set combo box to first entry
```

```
UserForm2.ComboBox13.ListIndex = 0
Var_ComboBox13Existe = 1
End If

End Sub

Private Sub OptionButton1_Click()
    TextBox8.Enabled = True

End Sub

Private Sub OptionButton2_Click()
    TextBox8.Enabled = False

End Sub

Private Sub OptionButton3_Click()
    TextBox11.Enabled = True
    TextBox1.Enabled = False

End Sub

Private Sub OptionButton4_Click()
    TextBox11.Enabled = False
    TextBox1.Enabled = True

End Sub

Private Sub TextBox1_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 TextBox102_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 TextBox11_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 TextBox2_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 TextBox25_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 TextBox34_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 TextBox35_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 TextBox36_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 TextBox44_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 TextBox45_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 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 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 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 TextBox8_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
```