



Transporte Robotizado de Bagagens em Aeroportos

MIGUEL MARQUES PEREIRA

novembro de 2021

Transporte Robotizado de Bagagens em Aeroportos

Miguel Marques Pereira



Mestrado em Engenharia Eletrotécnica 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 Unidade Curricular de Tese/Dissertação, do 2º ano, do Mestrado em Engenharia Eletrotécnica e de Computadores

Candidato: Miguel Marques Pereira, Nº 1151397, 1151397@isep.ipp.pt
Orientação científica: Manuel Fernando dos Santos Silva, mss@isep.ipp.pt
Coorientação científica: André Siqueira da Silva, and@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

4 de novembro de 2021

Agradecimentos

Para a realização deste projeto foi necessária a ajuda de algumas pessoas e sem as quais não teria sido fácil consegui-lo. Devo agradecer a todas estas pessoas mas em especial a dois Engenheiros.

Ao Engenheiro Manuel Silva, por ter aceite ser o meu orientador e, de seguida, ter prestado todo o auxílio que foi imprescindível ao longo de toda a tese/dissertação. A todos os conselhos e sugestões dados e o constante cuidado em perceber o avanço do trabalho realizado pois sem o mesmo seria muito mais difícil ter terminado o projeto.

Ao Engenheiro André Siqueira, por ter aceite ser o coorientador deste projeto e, de seguida, ajudado imenso na parte prática ao ter o cuidado de explicar todo o conhecimento que tem sobre o aeroporto.

Resumo

A finalidade deste projeto foi estudar a possibilidade de implementar, recorrendo a ferramentas de simulação, um transporte robotizado de bagagens em aeroportos, ou seja, com robôs móveis.

Para tal foi necessário, primeiramente, verificar que tipos de AGV e robôs móveis existem nesta indústria e verificar qual o que melhor se enquadra para este projeto. Após esta verificação, segue-se a pesquisa sobre possíveis *softwares* de simulação para robôs móveis. Foi necessário analisá-los e avaliar qual o que melhor correspondia às expectativas do projeto, de forma a se tirar o máximo proveito do *software*. De seguida, para terminar a parte de pesquisa, verifica-se alguns dos projetos que já foram realizados com robôs móveis em áreas de logística.

Posto isto, segue-se a parte prática do projeto. Começa-se por apresentar o *layout* do aeroporto, assim como os possíveis trajetos dos aviões e dos AGV dentro do aeroporto. Também o fluxo (quantidades) que entra e sai do aeroporto, tanto dos aviões como dos passageiros e bagagens.

Depois de ser demonstrado como é na realidade, segue-se a implementação do que foi descrito no capítulo anterior, começa-se por explicar como foram criadas as bagagens, a importação de objetos e a implementação dos percursos dos aviões e AGV dentro do aeroporto. Explica-se também como é que ambos os percursos comunicam entre si no projeto implementado.

Por fim, são apresentados os testes realizados e os resultados obtidos; estes testes são vários cenários com diferentes números de AGV em funcionamento de forma a se encontrar o número ideal para minimizar as despesas ao máximo.

Este projeto tem como principais vantagens a automatização de uma tarefa em aeroportos, ou seja, o transporte de bagagens desde o terminal para o avião e vice-versa. É um investimento que a médio-longo prazo será fortemente recompensado. A principal desvantagem acaba por ser que a curto prazo é um grande investimento.

Palavras-Chave: Robôs Móveis, AGV, Transporte de Bagagens, Aeroporto, Flexsim, Simulação.

Abstract

The purpose of this project was to study the possibility of implementing, using simulation tools, a robotic transport of baggage at airports, that is, with mobile robots.

For that, it was necessary, first, to verify what types of AGV and mobile robots exist in this industry and verify which one is the best fit for this project. After this verification, starts the research on possible simulation software for mobile robots. It was necessary to analyze them and evaluate which one correspond better to the project's expectations, in order to get the maximum out of the software. Then, to finish the research part, it is researched some of the projects that have already been carried out with mobile robots in logistics areas.

That said, the practical part of the project follows. It begins by showing the layout of the airport, as well as the possible routes for planes and AGV within the airport. Also the flows (quantities) entering and leaving the airport, from planes and from passengers and luggage.

After being demonstrated how it looks in reality, starts the implementation of what was described in the previous chapter, beginning by explaining how baggage was created in the software, the import of objects and the implementation of airplanes and AGV routes within the airport. It is also explained how both paths communicate with each other in the implemented project.

Finally, the tests performed and the results obtained are presented, these tests are several scenarios with different numbers of AGV in operation in order to find the ideal number to minimize expenses as much as possible.

The main advantages of this project are the automation of a task at airports, that is, the transport of luggage from the terminal to the plane and vice versa. It is an investment that in the medium to long term will be highly rewarded. The main disadvantage turns out to be that in the short term it is a great investment.

Keywords: Mobile Robots, AGV, Baggage Transport, Airport, Flexsim, Simulation.

Índice

Agradecimentos	v
Índice	xi
Índice de Figuras	xv
Índice de Tabelas	xix
Acrónimos	xxi
Glossário	xxiii
1 Introdução	1
1.1 Contextualização	1
1.2 Objetivos	2
1.3 Calendarização	2
1.4 Organização do Relatório	3
2 Revisão Bibliográfica	5
2.1 Robôs Móveis / AGV	5
2.2 Tipos de AGV	6
2.2.1 Automated Guided Cart	6
2.2.2 AGV Rebocador	7
2.2.3 AGV Carga Unitária	8
2.2.4 AGV Empilhador	9
2.2.5 Robôs Móveis Autónomos	11
2.3 Sistemas de Navegação	12
2.4 Sistemas de Direcção	14
2.5 Sistemas de Controlo	16
2.6 Conclusão do Capítulo 2	17

3	Robôs móveis para tarefas logísticas	19
3.1	Robôs Móveis	19
3.2	Aplicações	20
3.2.1	Vantagens e Desvantagens	21
3.3	Aplicações de AGV / AMR	22
3.3.1	Robô Móvel para Processo de Marcação em Tanques de Carga	23
3.3.2	<i>Fetch Robotics</i>	23
3.3.3	Navegação e Localização de AGV num Armazém via Rede de Sensores <i>Wireless</i>	24
3.3.4	Navegação do Robô Via GPS e Desvio de Obstáculos	25
3.3.5	<i>InVia Robotics</i>	26
3.3.6	Robô Móvel Com Manipulador Usado Como Protector De Segurança	27
3.4	Conclusão do Capítulo 3	28
4	Aplicações de Simulação para Robôs Móveis	29
4.1	FlexSim	29
4.2	Visual Components	31
4.3	Simio	33
4.4	Simul8	34
4.5	Anylogic	35
4.6	Escolha do <i>Software</i>	36
4.7	Conclusão do Capítulo 4	37
5	Aeroporto Francisco Sá Carneiro	39
5.1	<i>Layout</i> do Aeroporto	39
5.2	Trajetos dos Aviões e AGV	41
5.3	Fluxo de Aviões	42
5.4	Fluxo de Passageiros e Bagagens	42
5.5	Regras de Segurança	43
5.6	Conclusão do Capítulo 5	44
6	Modelação do Aeroporto	47
6.1	Importação e Criação do <i>Layout</i>	47
6.2	Mapeamento 3D	48
6.3	Criação de Grupos	49
6.4	Características dos AGV	50
6.5	Tipos de Ligação	52
6.6	Atividades Mais Utilizadas	52
6.6.1	<i>Assign Label</i>	52
6.6.2	<i>Wait For Event</i>	52
6.6.3	<i>Decide</i>	53

6.6.4	<i>Pull/Push From/To List</i>	53
6.6.5	<i>Run Sub Flow</i>	55
6.6.6	<i>Sources</i>	55
6.7	Classificação das Bagagens	56
6.8	Arquitetura do projeto	57
6.9	<i>Process Flow</i>	58
6.9.1	<i>Container</i> das Bagagens	59
6.9.1.1	Propriedades das <i>Schedule Source</i>	59
6.9.1.2	Propriedades das <i>Event-Triggered Source</i>	61
6.9.1.3	Recursos do Aeroporto	64
6.9.2	Fluxograma do Fluxo de Aviões	66
6.9.3	<i>Container</i> do Fluxo de Aviões	67
6.9.3.1	Pré Chegada ao Aeroporto	67
6.9.3.2	Carregamento das Bagagens de Chegada	68
6.9.3.3	Deslocação até ao <i>Stand</i> e Descarregamento	70
6.9.3.4	Aviões no Terminal	72
6.9.3.5	Espera Pelas Bagagens de Partida	73
6.9.3.6	Aquisição das Bagagens Através de Listas	74
6.9.3.7	Carregamento das Bagagens de Partida	76
6.9.3.8	Saída do Aeroporto	77
6.9.3.9	Aspeto Final do <i>Container</i> Referente ao Fluxo de Aviões	78
6.9.4	Fluxograma do Fluxo de AGV	78
6.9.5	<i>Container</i> do Fluxo de AGV	79
6.9.5.1	Espera por Bagagens	80
6.9.5.2	Onde Surgiram as Bagagens?	82
6.9.6	Bagagens de Chegada	83
6.9.6.1	Carregamento e Descarregamento das Bagagens de Chegada	85
6.9.6.2	Há Mais Bagagens?	86
6.9.6.3	Último <i>Pull From List</i>	87
6.9.7	Bagagens de Partida	88
6.9.7.1	Carregar Bagagens e Viagem até ao <i>Stand</i>	90
6.9.7.2	Descarregar Bagagens e Viagem até ao Terminal	91
6.9.7.3	Há Mais Bagagens?	93
6.9.7.4	Último <i>Pull From List</i>	93
6.9.7.5	Aspeto Final do <i>Container</i> Referente ao Fluxo de AGV	94
6.9.8	Aspeto Final do <i>Processflow</i>	95
6.10	Conclusão do Capítulo 6	96
7	Testes Realizados e Resultados Obtidos	97

7.1	Valores das <i>Schedule Sources</i>	97
7.2	Testes com Vários Cenários	99
7.2.1	Cenário A: 13 AGV	99
7.2.2	Cenário B: 14 AGV	101
7.2.3	Cenário C: 15 AGV	102
7.2.4	Cenário D: 16 AGV	103
7.2.5	Cenário E: 17 AGV	104
7.3	Tabela Comparativa	106
7.4	Conclusão do Capítulo 7	107
8	Conclusões	109
8.1	Conclusão do Trabalho	109
8.2	Ideias Para Desenvolvimentos Futuros	109
	Referências Bibliográficas	111
A	Schedule Source	119

Índice de Figuras

1.1	Gantt Chart com o plano de trabalhos de TEDI	2
2.1	AGC, a) AGC <i>Under Ride</i> , b) versão com gancho [1, 2]	7
2.2	AGV Rebocador [3]	7
2.3	AGV Carga Unitária [4]	8
2.4	Estabilizadores do AGV Porta Paletes [4]	9
2.5	Aeroporto do Reino Unido, a) AGV Porta Paletes, b) Veículos estacionados [5]	10
2.6	Apoio do AGV Empilhador [6]	11
2.7	Exemplo de um AMR [7]	11
2.8	AGV Filoguiado [8]	13
2.9	AGV Ótico [9]	13
2.10	Velocidade diferencial, a) em frente, b) correcção para a esquerda, c) correcção para a direita [10]	15
2.11	Combinação dos modos de direcção [11]	15
2.12	Implementação do Sistema de Controlo de Zona [12]	16
3.1	AGV num hospital [13]	20
3.2	AGV no Parque Temático [14]	21
3.3	Técnica de localização do robô móvel [15]	23
3.4	Robô colaborativo da Fetch Robotics [16]	24
3.5	Projeto Desenvolvido, a) na Realidade, b) em Simulação [17]	25
3.6	RC Truck com GPS e Sonar [18]	26
3.7	Robô da InVia Robotics [19]	26
3.8	Centro de controlo e alertas [20]	27
4.1	Process Flow Flexsim [21]	30
4.2	Ambiente gráfico do Flexsim [22]	31
4.3	eCatalog no Visual Components [23]	32
4.4	Realidade virtual no Visual Components [23]	32

4.5	Exemplo de um projeto no Simio [24]	33
4.6	Exemplo de um projeto no Simul8 [25]	34
4.7	Métodos de simulação [26]	35
5.1	Vista aérea do Aeroporto Francisco Sá Carneiro	40
5.2	<i>Layout</i> do Aeroporto Francisco Sá Carneiro	40
5.3	Trajetos dos aviões dentro do aeroporto	41
5.4	Trajetos dos AGV dentro do aeroporto	42
5.5	Perímetro de segurança de um avião[27]	43
5.6	Circulação de bagagens e passageiros no avião [27]	44
6.1	Aspeto de uma bagagem	48
6.2	<i>Layout</i> feito no Flexsim	48
6.3	<i>Layout</i> do percurso dos AGV	49
6.4	Propriedades das velocidades dos AGV	51
6.5	Dimensões de uma bagagem de porão	51
6.6	<i>Assign Label</i> , a) Atividade, b) Propriedades <i>default</i>	52
6.7	<i>Wait For Event</i> a) Atividade, b) Propriedades <i>default</i>	53
6.8	<i>Decide</i> , a) Atividade, b) Propriedades <i>default</i>	53
6.9	<i>Pull/Push From/To List</i> , a) Atividade, b) Propriedades <i>default</i>	54
6.10	<i>Run Sub Flow</i> , a) Atividade, b) Propriedades <i>default</i>	55
6.11	<i>Sources</i> , a) Atividades, b) Propriedades <i>default</i>	56
6.12	Diagrama de blocos, a) Bagagens de chegada, b) Bagagens de partida	57
6.13	Fluxograma realizado para as bagagens, a) Bagagens de chegada, b) Bagagens de partida	58
6.14	Propriedades das listas globais	59
6.15	<i>Schedule source</i> para as bagagens de chegada, a) Atividades, b) Propriedades	60
6.16	<i>Schedule source</i> para as bagagens de partida, a) Atividades, b) Propriedades	61
6.17	<i>Event-triggered source</i> para as bagagens de partida, a) Atividades, b) Propriedades	62
6.18	<i>Event-triggered source</i> para as bagagens de chegada, a) Atividades, b) Propriedades	63
6.19	Propriedades do <i>Assign Labels</i> e Código	64
6.20	Recursos do aeroporto, a) Atividades, b) Propriedades	65
6.21	Lista dos aviões, a) Atividades, b) Propriedades	66
6.22	Fluxograma realizado para o Fluxo de Aviões	66
6.23	<i>Schedule Source</i> , a) Atividade, b) Propriedades	67
6.24	<i>Acquire</i> , <i>Wait For Event</i> e <i>Delay</i> , a) Atividades, b) Propriedades	68
6.25	<i>Pull From List</i> , <i>Travel</i> , <i>Move Object</i> e <i>Release</i> , a) Atividades, b) Propriedades	69

6.26	<i>Travel</i> e <i>Run Sub Flow</i> , a) Atividades, b) Propriedades	70
6.27	<i>Unload</i> e <i>Push To List</i> , a) Atividades, b) Propriedades	72
6.28	<i>Push To List</i> , <i>Wait For Event</i> e <i>Travel</i> , a) Atividades, b) Propriedades	73
6.29	<i>Decide</i> , <i>Wait For Event</i> e <i>Delay</i> , a) Atividades, b) Propriedades	74
6.30	<i>Pull From Lists</i> , a) Atividades, b) Propriedades	75
6.31	<i>Travel</i> e <i>Run Sub Flow</i> , a) Atividades, b) Propriedades	76
6.32	<i>Travel</i> e <i>Run Sub Flow</i> , a) Atividades, b) Propriedades	77
6.33	Aspetto final do <i>container</i> do fluxo de aviões	78
6.34	Fluxograma realizado para o Fluxo de AGV	79
6.35	<i>Schedule Source</i> e <i>Assign Labels</i> , a) Atividades, b) Propriedades	80
6.36	<i>Acquire</i> , <i>Wait For Event</i> e <i>Delay</i> , a) Atividades, b) Propriedades	81
6.37	<i>Decide</i> de partida e <i>Decide</i> de chegada e, a) Atividades, b) Propriedades	82
6.38	<i>Increment Label</i> , <i>Decide</i> e <i>Assign Label</i> , a) Atividades, b) Propriedades	83
6.39	<i>Pull From List</i> , <i>Travel</i> e <i>Release</i> , a) Atividades, b) Propriedades	84
6.40	<i>Run Sub Flow</i> , <i>Travel</i> , <i>Run Sub Flow</i> e <i>Travel</i> , a) Atividades, b) Propriedades	85
6.41	<i>Decides</i> bagagens de chegada, a) Atividades, b) Propriedades	87
6.42	<i>Pull From List</i> , a) Atividade, b) Propriedades	88
6.43	<i>Pull From List</i> , <i>Travel</i> e <i>Release</i> , a) Atividades, b) Propriedades	89
6.44	<i>Run Sub Flow</i> e <i>Travel</i> até <i>stand</i> , a) Atividades, b) Propriedades	90
6.45	<i>Run Sub Flow</i> e <i>Travel</i> até terminal, a) Atividades, b) Propriedades	91
6.46	<i>Unload</i> e <i>Push To List</i> , a) Atividades, b) Propriedades	92
6.47	<i>Decides</i> bagagens de partida, a) Atividades, b) Propriedades	93
6.48	<i>Pull From List</i> , a) Atividade, b) Propriedades	94
6.49	Aspetto final do <i>container</i> do fluxo de AGV	95
6.50	Aspetto final do <i>processflow</i>	96
7.1	Planeamento dos aviões em terra	98
7.2	Tempos de espera com 13 AGV para bagagens não prioritárias	100
7.3	Tempos de espera com 14 AGV para bagagens não prioritárias	101
7.4	Tempos de espera com 15 AGV para bagagens não prioritárias	103
7.5	Tempos de espera com 16 AGV para bagagens não prioritárias	104
7.6	Tempos de espera com 17 AGV para bagagens não prioritárias	105
A.1	Valores <i>Schedule Source</i> , a) Bagagens de Chegada, b) Bagagens de Partida	119

Índice de Tabelas

4.1	Tabela comparativa das aplicações de <i>Softwares</i> para simulação analisadas	36
6.1	Tabela referente aos grupos criados no Flexsim	50
7.1	Resultados obtidos com 13 AGV	100
7.2	Resultados obtidos com 14 AGV	101
7.3	Resultados obtidos com 15 AGV	102
7.4	Resultados obtidos com 16 AGV	103
7.5	Resultados obtidos com 17 AGV	105
7.6	Tabela comparativa com os valores de tempo de espera	106

Acrónimos

Acrónimo	Descrição	Página
ISEP	Instituto Superior de Engenharia do Porto	1
DEE	Departamento de Engenharia Eletrotécnica	1
TEDI	Tese/ Dissertação	1
MEEC	Mestrado em Engenharia Eletrotécnica e de Computadores	1
AGV	<i>Automated Guided Vehicle</i>	2
AGC	<i>Automated Guided Cart</i>	6
IPT	<i>Inductive Power Transfer</i>	8
VNA	<i>Very Narrow Aisle</i>	9
AMR	<i>Autonomous Mobile Robots</i>	6
LiDAR	<i>Light Detection and Ranging</i>	12
SLAM	<i>Simultaneous Localization And Mapping</i>	12
GPS	<i>Global Positioning System</i>	12
RF	<i>Radio Frequency</i>	13
FIFO	<i>First In First Out</i>	16
WLAN	<i>Wireless Local Area Network</i>	16
LPS	<i>Local Positioning System</i>	21
RFID	<i>Radio-Frequency Identification</i>	21
CAD	<i>Computer-Aided Design</i>	30
TSA	<i>Transportation Security Administration</i>	30
RC	<i>Radio-controlled</i>	25
PLC	<i>Programmable Logic Controller</i>	31
RaaS	<i>Robots as a Service</i>	26
CEO	<i>Chief Executive Officer</i>	26
AI	<i>Artificial Intelligence</i>	26
CMR	<i>Collaborative Mobile Robot</i>	23
WSCCM	<i>Wireless Static Control Central Mode</i>	24
LNGC	<i>Liquefied Natural Gas Carrier</i>	23
OPO	Aeroporto Francisco Sá Carneiro	39
GPU	<i>Ground Power Unit</i>	44
NAV	Navegação Aérea de Portugal	48

Glossário

Palavra	Descrição	Página
PAX	Número de passageiros transportados.	41
Ponte Telescópica	Ligação existente entre o avião e o aeroporto para a entrada e saída de passageiros.	41
<i>Stand</i>	Local onde os aviões ficam posicionados para assistência.	41
<i>Token</i>	Tokens são objetos que conseguem fluir através das atividades num <i>process flow</i> .	49

Capítulo 1

Introdução

Neste capítulo será feita a contextualização dos temas abordados e serão definidos os objetivos propostos. De seguida, apresentar-se-á um diagrama de Gantt, que mapeia as tarefas desenvolvidas, assim como o tempo que estas ocuparam. Por fim, descrever-se-á a estrutura do relatório e resumir-se-á o conteúdo do mesmo.

Esta dissertação descreve o projeto realizado e desenvolvido na unidade curricular de Tese / Dissertação (TEDI), integrada no 2^o ano do Mestrado em Engenharia Eletrotécnica e de Computadores (MEEC) – Ramo de Automação e Sistemas, no Departamento de Engenharia Eletrotécnica (DEE) do Instituto Superior de Engenharia do Porto (ISEP).

1.1 Contextualização

Este projeto surgiu do interesse em realizar um projeto na área de robótica, neste caso em específico, em realizar o projeto “Transporte Robotizado de Bagagens em Aeroportos”, proposto pelo Eng. Manuel Silva, houve ainda o cuidado para que este esteja enquadrado com a unidade curricular de TEDI no ramo de Automação e Sistemas.

Para tal, foi necessário realizar uma revisão bibliográfica sobre o tema de forma a estar enquadrado com o mesmo e, de seguida, analisar e escolher qual o *software* mais adequado. Será realizada a simulação com dados reais sobre o projeto pretendido de modo a que este se torne um projeto desafiante, interessante e que tenha um nível de dificuldade adequado à unidade curricular a que o mesmo pertence.

1.2 Objetivos

O projeto recai sobre a simulação do tema “Transporte Robotizado de Bagagens em Aeroportos”, assim como a análise da mesma de forma a se verificar as vantagens e o impacto que a sua implementação iria trazer. Para isso, foi necessário superar alguns obstáculos, sendo estes os seguintes:

- Estudar os tipos de AGV existentes no mercado e escolher o que melhor se adapta ao projeto;
- Verificar projetos existentes na mesma área e com robôs móveis;
- Pesquisar e decidir o *software* mais adequado ao projeto;
- Analisar o impacto, vantagens e desvantagens, da utilização de robôs móveis;
- Aprender a trabalhar com o *software* Flexsim;
- Estudar e desenhar o *layout* do aeroporto;
- Implementação do fluxo de aviões dentro do aeroporto;
- Implementação do fluxo de AGV dentro do aeroporto;
- Verificar o tratamento e transporte das bagagens em aeroportos;
- Sincronizar os fluxos através do tratamento de bagagens;
- Analisar os resultados obtidos.

1.3 Calendarização

Sendo o Transporte Robotizado de Bagagens em Aeroportos a principal razão para a qual esta dissertação foi realizada, foi feito um Gantt Chart, ilustrado na Figura 1.1, que sumariza a calendarização de todo o trabalho desenvolvido. Esta figura inclui as tarefas realizadas bem como o tempo despendido para a realização de cada uma destas.



Figura 1.1: Gantt Chart com o plano de trabalhos de TEDI

1.4 Organização do Relatório

Esta dissertação é composta por oito capítulos: Introdução, Revisão Bibliográfica, Robôs Móveis para Tarefas Logísticas, Aplicações de Simulação para Robôs Móveis, Aeroporto Francisco Sá Carneiro, Modelação do Aeroporto, Testes Realizados e Resultados Obtidos e, por fim, Conclusões.

No Capítulo 1 é apresentada uma introdução e, também, explicado, resumidamente, o que foi feito neste projeto, os objetivos e o caminho seguido para se chegar aos mesmos, a calendarização do projeto e a organização do relatório em si.

No Capítulo 2 é feita uma revisão bibliográfica sobre os AGV, começando por explicar e distinguir os diferentes tipos que existem, de seguida a forma como se orientam no espaço, descrevendo os sistemas de navegação e de controlo existentes e, por fim, as suas aplicações, vantagens e desvantagens.

No Capítulo 3 são apresentados alguns robôs móveis para tarefas logísticas, quais os tipos mais utilizados e os exemplos de aplicações e projectos que já estejam em funcionamento nas empresas, ou seja, exemplos já implementados.

No Capítulo 4 é feita uma pesquisa sobre vários *softwares* de simulação e, em cada um, é feita uma descrição e são apresentadas as suas funcionalidades; no fim os mesmos são comparados de forma a se verificar qual é o que melhor se adequa a este projeto em questão.

No Capítulo 5 é apresentado o aeroporto escolhido para realizar a simulação deste projeto. Inicialmente é apresentado o *layout* do aeroporto e, de seguida, são descritos os trajetos dos aviões e dos AGV, assim como os fluxos dos mesmos. Além disso é também estudado o fluxo das bagagens até à entrada do terminal de bagagens de chegada. Por fim, são explicadas regras de segurança a considerar quando chega um avião.

No Capítulo 6 é demonstrado tudo o que implicou a implementação, no *software* escolhido, do projeto, desde o desenho do aeroporto até à implementação dos diferentes fluxos: bagagens, aviões e AGV.

No Capítulo 7 são demonstrados os testes feitos e os resultados obtidos de acordo com cada teste. Por fim, é avaliado qual o melhor cenário para o projeto, ou seja, a quantidade de AGV que melhor se enquadra para um dia de trabalho no aeroporto Francisco Sá Carneiro, de forma a que os passageiros esperem o mínimo possível pelas bagagens.

No Capítulo 8 são apresentadas as devidas conclusões retiradas do projeto desenvolvido, assim como eventuais melhorias que poderão ser aplicadas ao mesmo. Por fim é feita uma apreciação do mesmo.

Capítulo 2

Revisão Bibliográfica

Neste capítulo será feita uma revisão bibliográfica sobre os diferentes tipos de Automated Guided Vehicle (AGV). Serão descritos os aspectos importantes do AGV, tais como os tipos que existem e os diferentes sistemas que existem: de navegação, de controlo e de direção.

2.1 Robôs Móveis / AGV

Em 1913, a *Ford Motor Company* dos Estados Unidos usou veículos guiados de forma orbital, que foi o primeiro desenvolvimento do AGV. Esta tecnologia foi atualizada no Reino Unido, e lá foi desenvolvido pela primeira vez o AGV guiado de forma eletromagnética, este faz com que o AGV se livre do inconveniente ao ser colocado na pista e, assim, facilita a aplicação do AGV. No final da década de 1950, o AGV foi amplamente promovido e aplicado na Europa.

Já na década de 1980, o Japão introduziu o AGV com tecnologia de navegação magnética sob a influência da tecnologia do Reino Unido, o que melhorou muito a estabilidade do AGV e fez com que o mesmo fosse amplamente utilizado na área automóvel.

Com o aprofundamento das trocas entre a China e o Japão, muitas empresas internacionais começaram a perceber que os AGV podem economizar mão de obra, melhorar a eficiência do trabalho e economizar custos. Portanto, nos últimos anos, os AGV também se desenvolveram de forma rápida na China [28].

O AGV é um veículo que consegue mobilizar-se tanto em ambientes internos como externos e que se desloca de forma autónoma após ser realizada a sua programação, geralmente este é utilizado para transportar materiais. Este percorre um caminho definido, orientando-se no espaço graças a sensores e atuadores que

serão explicados na secção sistemas de navegação. Este tipo de veículo foi principalmente desenvolvido e programado para a entrega e recolha de produtos em espaço fabril ou em armazéns.

2.2 Tipos de AGV

Nesta secção são demonstrados os diferentes tipos de AGV que existem na indústria de automação. A sua utilização aumentou drasticamente durante o século XX, devido aos seus diversos benefícios e enorme versatilidade. Existem vários tipos de AGV, porém os principais, e mais utilizados, são: *Automated Guided Cart* (AGC), AGV Rebocador, Carga Unitária e AGV Empilhador; dentro deste último existem ainda vários subtipos, sendo estes, Empilhadora de contra balanço, Porta Paletes e de Longo Alcance [29].

A principal diferença entre o AGV e os robôs móveis autónomos (AMR) resume-se na utilização de caminhos pré definidos, seja por obras do espaço de trabalho (geralmente no solo) ou no planeamento prévio de trajetórias. Por exemplo, caso surja um novo obstáculo no espaço de trabalho, o AGV, caso o mesmo interfira com a sua trajetória irá necessitar de uma revisão para contornar e, assim, evitar este obstáculo enquanto que o robô móvel, de forma autónoma, irá realizar o contorno ou desvio de forma autónoma.

2.2.1 Automated Guided Cart

O AGC é uma variante do AGV e, por isso, são idênticos, porém estes circulam debaixo da carga a transportar, daí também serem chamados de *Under Ride* (ver Figura 2.1.a), são flexíveis e de fácil instalação, geralmente têm um custo menor que os restantes tipos de AGV. É uma das soluções para realizar tarefas repetitivas, como o transporte de cargas de um ponto A até um ponto B [30].

Inicialmente, foram projetados para realizar tarefas fáceis em *loop* porém, ao longo do tempo, foram melhorados a nível de *performance* e, atualmente, conseguem realizar atividades mais complexas. Uma das maiores vantagens é que conseguem carregar e descarregar sem a intervenção do ser humano [31].

A capacidade de carga típica de um AGC é, geralmente, 0,5 toneladas (500 kg), porém é possível encontrar AGC com capacidade de carga até 3 toneladas. Em relação à sua mobilidade, a velocidade *default* de um AGC é de 40 m/min, ou seja, de 2,4 km/h, mas, mais uma vez, há modelos onde se é possível atingir a velocidade de 60 m/min, 3,6 km/h [32]. Uma das versões do AGC é com um gancho e, assim, é possível transportar um *trolley* (ver Figura 2.1.b), logo, neste caso, é possível aumentar a capacidade de carga para 3 toneladas [2].

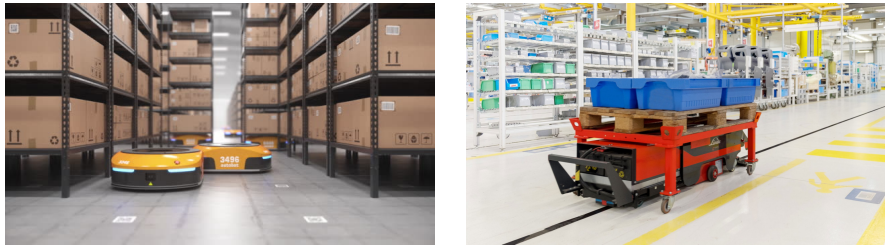


Figura 2.1: AGC, a) AGC *Under Ride*, b) versão com gancho [1, 2]

2.2.2 AGV Rebocador

O AGV Rebocador consegue transportar várias cargas em simultâneo, ou seja, como se fosse uma locomotiva a transportar os diversos vagões (ver Figura 2.2).

Foi o primeiro a ser introduzido no mundo dos AGV e é o mais popular, é designado, e mais conhecido, por *Tugger* e é frequentemente utilizado para transportar cargas pesadas em distâncias mais longas visto que este possui uma capacidade de carga de até 4 toneladas, dependendo do AGV tanto o engate como o desengate da carga pode ser manual ou automático [33]. Sendo assim, pode-se destacar que as empresas que utilizam este veículo não estão a retirar totalmente a mão de obra dos seus processos, mas sim, a optimizá-los e a torná-los eficientes que é o principal objectivo de um AGV: tornar uma determinada tarefa eficiente.

A grande vantagem deste AGV em específico, é que este consegue movimentar mais cargas, em apenas uma viagem, do que uma empilhadora convencional e pode ter várias paragens de entrega e recolha ao longo do percurso definido pelo espaço onde se encontra, seja em armazém ou fábrica. Desta forma, a aplicação do sistema no abastecimento logístico das linhas de produção torna-se muito mais rápida, fácil e eficiente [34].



Figura 2.2: AGV Rebocador [3]

De forma a conseguir obter especificações/informações sobre este tipo de AGV, foi consultado um exemplo de uma empresa, denominado por AGV LINCE. Este tem vários tipos de processos de carga das baterias: Carregador a bordo, Troca Manual, *Opportunity Charging* e *Inductive Power Transfer* (IPT), ou seja, Carga por Indução, tem também diferentes sistemas de navegação, sendo que são todos através de sensores: indutivo, óptico, laser e magnético. A sua velocidade pode ir até 60 m/min, ou seja, 3,6 km/h [3].

2.2.3 AGV Carga Unitária

Os AGV de carga unitária são projetados para transportar, tal como o seu nome indica, um único produto, este pode ser um motor ou bobina, ou até uma paleta que contenha vários produtos. Podem ser equipados com plataformas elevatórias, correntes ou correias transportadoras. Deste modo, podem ser personalizados para uma tarefa em específico [35].

Quando este é equipado com correntes ou correias transportadoras tem rolos que permitem transferir as cargas para os transportadores e pode carregar, ou descarregar, em ambas as direções, o que melhora a eficiência operacional. Uma das maiores desvantagens destes AGV é que não podem pegar em cargas que se encontrem no chão [36].

A capacidade de carga deste AGV varia consoante o modelo a utilizar. Por exemplo, o modelo com plataforma elevatória (ver Figura 2.3) pode ter cargas que tenham até 13600 kg. Consegue transportar vários tipos de paletes com o mesmo AGV e consegue posicionar as cargas com boa precisão independentemente do tamanho da carga (por exemplo, uma paleta). Devido à sua plataforma elevatória consegue uma melhor adaptação às exigências da tarefa a realizar e é ideal para manusear cargas que não possam ser carregadas através de uma empilhadora, à semelhança de quando são equipados com correntes ou correias transportadoras não podem recolher cargas do chão [4].



Figura 2.3: AGV Carga Unitária [4]

2.2.4 AGV Empilhador

O AGV Empilhador é, geralmente, utilizado para realizar diversas tarefas, entre elas: mover cargas entre a produção e o armazém, recolher cargas dos transportadores, cargas em paletes, embalagens com invólucros de estiramento, etc. Existem vários tipos de AGV Empilhador, sendo os mais importantes e os que serão apresentados a seguir, AGV Porta Paletes, o AGV Empilhador de contrabalanço e o AGV Empilhador de Longo Alcance; além destes, existem outros ainda conhecidos tais como: AGV Empilhador para corredores muito estreitos ou *Very Narrow Aisle* (VNA) e AGV *Outrigger* (ou *Straddle*).

O primeiro exemplo de AGV Empilhador é o AGV Porta Paletes. Este é um dos tipos de AGV Empilhador que é concebido para o transporte de cargas (sobre paletes) em instalações fabris, este é bastante parecido a um porta-paletes convencional, porém automatizado. Tem um movimento vertical limitado consoante a carga, conseguindo, assim, deixar a carga em paletes empilhadas ou prateleiras que estejam perto do chão ou, ainda, em tarefas que se concretizem ao nível no do chão [37]. Os modelos do AGV Porta Paletes têm estabilizadores debaixo dos garfos (ver Figura 2.4), estes permitem que o AGV tenha uma dimensão reduzida e menos peso garantindo, assim, que este realize boas manobras, ou seja, fazem curvas mais fechadas e têm maior capacidade de manobra em corredores estreitos quando comparados com outros tipos de AGV. Realiza as funções típicas de uma empilhadora porém de forma mais eficiente [38].

A capacidade de carga típica deste AGV é de 1 tonelada e a amplitude dos garfos é, geralmente, de 1,5 metros, daí ser o ideal para tarefas que estejam ao nível do chão ou em prateleiras baixas. A velocidade máxima deste tipo de AGV Empilhador é de 2 m/s [39].

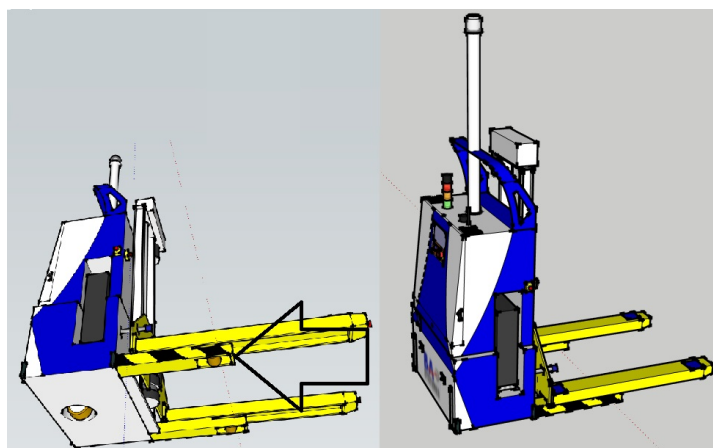


Figura 2.4: Estabilizadores do AGV Porta Paletes [4]

Um exemplo prático e real, já está a ser implementado, bastante interessante neste tipo de AGV é o que existe no Aeroporto de *Gatwick*, no Reino Unido, que juntamente com a *Stanley Robotics* (ver Figura 2.5.a)) criaram um sistema de estacionamento eficiente. Através do AGV Porta Paletes, sempre que detecta um carro, este desliza sob o veículo, levanta-o suavemente e move-o para a área de armazenamento (zona de parque). Além disso, o robô também regista os detalhes do itinerário do tripulante (carro) para que o mesmo esteja pronto para entrega quando o condutor voltar. Desta forma espera-se que os actuais 6000 locais de estacionamento sejam o suficiente para estacionar 8500 veículos, visto que estacionados com o AGV não precisam de espaço para os motoristas entrarem e saírem do veículo (ver Figura 2.5.b)) [5].



Figura 2.5: Aeroporto do Reino Unido, a) AGV Porta Paletes, b) Veículos estacionados [5]

Outro exemplo dos AGV Empilhadores é o AGV de Contra Balanço; tal como o seu nome indica, contra balança o peso da carga com um peso, geralmente ferro fundido ou uma bateria, na parte traseira do AGV. Ele é usado sempre que não é possível usar AGV de Porta Paletes, mas principalmente para envio e recolha de cargas. Têm velocidade típica de 1,7 m/s, são maiores, mas mais estreitos, e precisam de mais espaço para curvar que os AGV Porta Paletes, porém podem lidar com cargas mais pesadas e atingir alturas de empilhamento mais altas. Possuem um mastro de elevação posicionado na extremidade do AGV que proporciona maior versatilidade ao mesmo, quando se trata de lidar com vários tipos de carga [40, 41].

Por fim, o último exemplo do AGV Empilhador; é o AGV de Longo Alcance, são projetados para trabalhar em espaços de armazenamento com corredores estreitos e têm dois apoios, por baixo dos garfos, que distribuem o peso da carga (ver Figura 2.6). São os mais adequados para tarefas em que haja uma grande profundidade ou prateleiras muito altas, visto que apresenta um elevador hidráulico e consegue estender os garfos montados nos veículos até 7,62 m [42], tanto pode guardar como recolher as cargas que estejam desde o nível do solo até à altura alcançada. Uma das principais vantagens deste AGV é a velocidade de

levantamento e o alto desempenho [43].



Figura 2.6: Apoio do AGV Empilhador [6]

2.2.5 Robôs Móveis Autônomos

Os AMR são normalmente mais avançados tecnologicamente do que os restantes tipos de AGV (ver Figura 2.7). Embora muitos AGV usem sistemas de navegação fixos, como fios ou fita magnética, vários AMR são equipados com recursos de navegação inteligente, ou seja, sensores e sistemas de câmaras, que permitem detectar e navegar pelo espaço de trabalho e, até, realizar o desvio e contorno de obstáculos.



Figura 2.7: Exemplo de um AMR [7]

Assim, graças à sua tecnologia mais sofisticada, em vez de seguirem percursos fixos, os AMR são capazes de criar dinamicamente os seus próprios caminhos eficientes de um Ponto A até ao Ponto B numa instalação e não necessitam de ser supervisionados pelo operador mas sim trabalhar de forma colaborativa com o mesmo, tendo assim maior flexibilidade e segurança. Porém, apesar de já existirem modelos para lidarem com cargas pesadas, geralmente os AMR lidam com cargas úteis menores e têm recursos de transporte vertical limitados [44, 45].

Um *Light Detection and Ranging* (LiDAR) consiste principalmente num laser, um *scanner* e um receptor *Global Positioning System* (GPS) especializado. O LiDAR é um sensor geralmente utilizado no sistema *Simultaneous Localization And Mapping* (SLAM). Estes sistemas, além do LiDAR, utilizam sensores ultrasónicos que transmitem pulsos de ultrasons para medir a distância entre o robô e os objetos ao seu redor. Estes, combinados com os dados provenientes de outros sensores, são compilados e geram informações tridimensionais, criando um mapa do espaço à sua volta, permitindo que os robôs naveguem pelas instalações em segurança e evitem obstáculos sem a necessidade de qualquer infraestrutura adicional. Com este mapeamento permitem que, por exemplo, cientistas examinem ambientes naturais e artificiais com exatidão, precisão e flexibilidade [46].

Existem duas principais formas de utilizar o LiDAR, o modo topográfico que normalmente usa um laser infravermelho (com comprimento de onda à volta dos 1,064 nm) para mapear o meio ambiente, enquanto que o *bathymetric* LiDAR usa um comprimento de onda verde na ordem dos 532 nm, que penetra a água e permite medir o fundo do mar ou elevações do leito do rio [47].

2.3 Sistemas de Navegação

Como já é habitual com o avanço tecnológico, o AGV também traz mais vantagens do que desvantagens e é, por isso, bastante utilizado pelas indústrias. Contudo, há aspectos importantes que devem ser analisados e estudados, sendo estes, qual o sistema de orientação recomendado a cada situação e como é que o AGV realiza as suas manobras, ou seja, o sistema de direcção e o tipo de controlo mais apropriado consoante o espaço de trabalho que se encontra. O sistema de navegação, ou seja, a forma como o AGV se move dentro de um espaço definido pode ser realizado de formas bastante distintas, porém sempre através de sensores, sendo algumas as seguintes: filo guiado, laser, seguidor de linha (navegação ótica) e, ainda ser controlado, ou por GPS ou por *Bluetooth*.

Ao escolher a deslocação filoguiada, a empresa necessita de realizar obras de forma a instalar o percurso desejado para o AGV (ver Figura 2.8). Esta é a forma mais simples de navegação visto que estabelece um percurso fisicamente pré-definido.

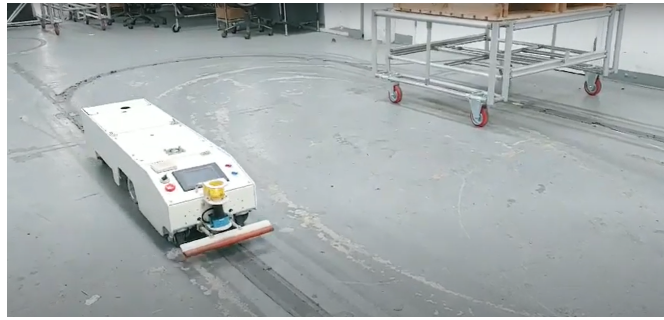


Figura 2.8: AGV Filoguiado [8]

Baseia-se num fio eléctrico embutido no pavimento e num sistema de detecção do campo magnético do AGV, basicamente um sensor para detectar o sinal de Radiofrequência (RF) proveniente do fio embutido (entre 2 a 3 centímetros abaixo da superfície). Visto que obriga a realizar a abertura de um corte no pavimento, estes sistemas apenas são utilizados quando os percursos não são susceptíveis a alterações, o que os torna muito pouco flexíveis e inadequados a indústrias onde existe a necessidade de reconfigurar o *layout* da fábrica [8, 48].

Ao utilizar o AGV com sistema de orientação óptico, o mais conhecido é o modo seguidor de linha, este é semelhante ao sistema anterior porém não é necessário realizar alterações a nível de construções no pavimento. Com este tipo de tecnologia o veículo segue uma fita que está colada no chão e que define o caminho que o AGV tem de percorrer (ver Figura 2.9). Esta fita é colorida no caso de uso dos sensores infravermelhos. Desloca-se através da informação obtida dos sensores infravermelhos que se encontram por baixo do mesmo, porém tem a desvantagem de não poder ser incorporada em áreas com elevado tráfego, pois a fita colorida pode ficar danificada ou suja [49].



Figura 2.9: AGV Ótico [9]

Este têm a vantagem de ter um custo de instalação inferior ao do sistema filoguiado visto que não é necessário realizar obras e, também, é possível alterar o percurso do AGV pois a fita está colada e pode ser colocada de outra forma. Tanto o sistema ótico como o sistema filoguiado, são ideais para os AGV rebocadores, uma vez que estes se deslocam sempre no mesmo circuito [50].

O desvio entre a fita e a superfície pode ser detetado para criar um sinal de erro e direccionar o AGV na direcção correcta. Apesar de ser fácil alterar o percurso usando este método, requer constante manutenção para garantir que a fita está em boas condições.

Por fim, no caso do sistema de orientação do AGV ser localizado via GPS e/ou ser controlado manualmente por *Bluetooth* ou outro meio de comunicação *wireless*, a área é mapeada e guardada na memória do AGV, o que lhe permite mover-se de forma autónoma, sem que haja necessidade de realizar alterações na infraestrutura do edifício. Esta solução é bastante flexível e ideal para percursos dinâmicos, uma vez que permite mudar facilmente a trajetória do AGV ao introduzir as alterações no *software* de mapeamento. Os dados guardados na memória, juntamente com outros sensores instalados no AGV melhoram a precisão do mesmo e é, por isso, a tecnologia que oferece maior fiabilidade [51].

2.4 Sistemas de Direcção

Os AGV têm diferentes configurações para a direcção, variando consoante a aplicação ou projeto que é desejado realizar, a sua complexidade é influenciada pela quantidade de componentes electrónicos necessários para o controlo da velocidade.

O controlo por velocidade diferencial consiste em ter duas rodas motrizes (uma roda de cada lado) ou então dois conjuntos de duas rodas motrizes (duas de cada lado) em que a direcção é dada pela diferença entre as velocidades aplicadas a cada um dos lados (ver Figura 2.10). Nesta figura é possível verificar as correcções que o AGV terá de fazer para continuar o percurso na linha.

No primeiro caso da figura ambos os lados têm a mesma velocidade, porém no segundo e terceiro caso o mesmo não acontece pois o AGV não responde da mesma forma nos dois casos, apesar de ambos terem duas soluções cada. No segundo caso, como é necessário virar à esquerda, ou se aumenta a velocidade do lado direito ou se reduz a velocidade do lado esquerdo ou, até, ambas as anteriores, quanto maior for esta diferença mais fechada será a curva que o AGV realiza. No terceiro caso acontece exatamente o contrário, ou seja, ou se aumenta a velocidade do lado esquerdo ou se reduz a velocidade do lado direito ou, à semelhança do anterior, ambos [52, 53].

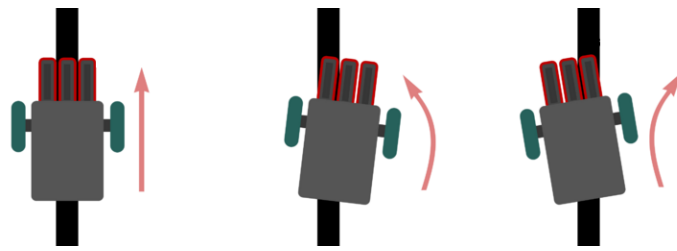


Figura 2.10: Velocidade diferencial, a) em frente, b) correcção para a esquerda, c) correcção para a direita [10]

O controlo por velocidade diferencial não deve ser utilizado em AGV com reboque pois o AGV faria com que o reboque se desviasse ao virar, sendo o mais adequado o controlo seguinte.

O controlo por volante, também conhecido por *Ackerman Steering Technique* [54], é outra das formas de controlar o sistema de direcção do AGV. Este tipo é mais utilizado em AGV que tenham apenas uma roda na frente. Resumidamente neste tipo é típico que as rodas dianteiras forneçam a direcção e as rodas traseiras forneçam a tracção (semelhante a um carro); apesar de este ser mais encontrado em AGV com 3 rodas (configuração em triciclo) também pode ser encontrado em AGV com 4 rodas. Tem maior precisão do que o de controlo de velocidade diferencial e oferece curvas mais suaves. Este controlo é geralmente utilizado no AGV rebocador [52, 53].

Por fim, existe ainda um terceiro tipo (ver Figura 2.11) que acaba por ser uma combinação dos anteriores onde dois motores independentes (estes realizam a direcção e tracção) estão em dois cantos (diagonais) e os outros dois cantos têm apenas rodas giratórias. Assim, o AGV que utiliza esta direcção combinada pode girar como um carro (em arco) em qualquer direcção, pode deslocar-se lateralmente e dirigir no modo diferencial em qualquer direcção [53, 49].

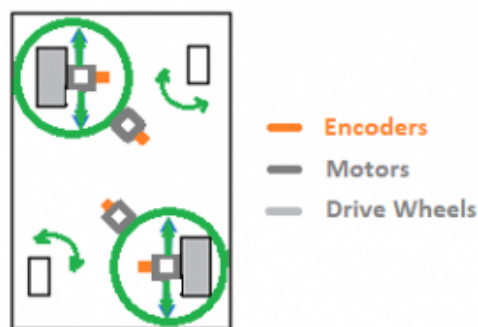


Figura 2.11: Combinação dos modos de direcção [11]

2.5 Sistemas de Controlo

Os sistemas de controlo de um espaço de trabalho, onde circulam um ou mais AGV em simultâneo, são importantes para evitar colisões entre os mesmos e, em simultâneo, tornar os seus percursos eficientes (efectuarem a menor distância possível). Há várias formas de o fazer, uma delas é o controlo da zona, este caso é semelhante a uma fila de mensagens que segue uma ordem de *First In First Out* (FIFO) [55, 12].

Este método é simples de instalar, consiste num transmissor sem fios que transmite sinais em áreas definidas. Este pode ser, por exemplo, via *Wireless Local Area Network* (WLAN), além disso o AGV terá de conter um sensor capaz de receber o sinal e remetê-lo de volta ao transmissor/emissor consoante a situação.

Caso a área esteja livre, um sinal para prosseguir é enviado, permitindo que o AGV continue o seu caminho. Se outro AGV estiver na área de trabalho, este recebe o sinal para o mandar parar e, além disso, este sinal alerta os restantes AGV, caso também estejam a tentar entrar no determinado local, de que a área não está limpa. Neste caso, os AGV em espera irão parar e esperar até que o AGV saia da zona; de seguida, o transmissor envia um sinal para que o próximo possa prosseguir. Ou seja, é exatamente igual ao que acontece numa fila de mensagens (o primeiro a chegar é o primeiro a sair), porém há excepções onde o sistema pode verificar a carga a transportar e a necessidade (no espaço de trabalho) da mesma, os mais urgentes deverão passar primeiro (ver Figura 2.12) [49].

Nesta figura é possível verificar a existência de três AGV, dois azuis e um amarelo. Existem dois pontos de decisão e, nas junções deve passar primeiro o que tiver mais prioridade, por exemplo: na junção que aparece mais acima na figura, caso o azul seja prioritário ao amarelo, o amarelo deverá esperar que o azul passe no ponto de junção e, apenas de seguida, continuar o seu percurso.

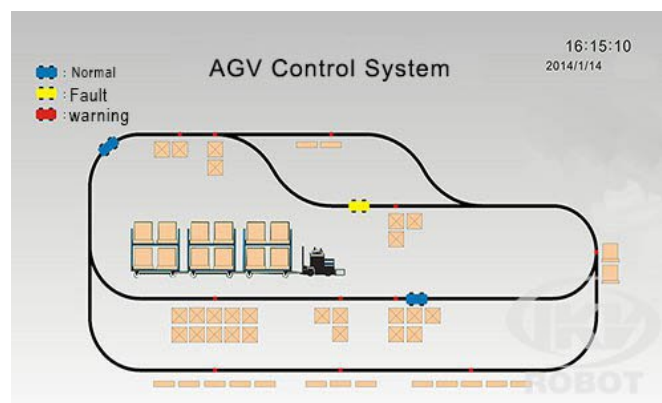


Figura 2.12: Implementação do Sistema de Controlo de Zona [12]

Há, ainda, outra forma de usar este sistema de controlo: consiste em equipar cada AGV com o seu próprio transmissor, permitindo que este envie um sinal de bloqueio para outros AGV que se aproximem da zona. Quando o AGV se encontra livre este deve estar na sua posição de início, ou seja, onde geralmente se encontram possíveis cargas a transportar, esta posição de início é, geralmente, utilizada para os AGV carregarem as suas baterias.

É, ainda, possível definir percursos independentes para cada AGV. É uma opção que dá mais trabalho e, geralmente, obriga a que o AGV faça um percurso mais longo do que poderia fazer caso utilizasse o sistema de controlar a zona [53].

Um outro método, muitas vezes conciliado com o Controlo da Zona, que é usado como sistema de controlo e, assim, evita colisões com objetos ou entre os AGV é equipar os robôs móveis com sensores que transmitem um sinal de frequência e esperam pela resposta, através da qual o AGV determina se está um objecto à sua frente e toma as medidas necessárias para evitar colisões. Geralmente este sensor é o ultrassónico e funciona como um radar ou então sensores infravermelhos, ambos funcionam de maneira semelhante e cumprem o objectivo [49].

2.6 Conclusão do Capítulo 2

Após a realização deste capítulo foi possível conhecer a principal diferença entre um AGV e um robô móvel, quais os tipos de AGV existentes no mercado e ter uma ideia de qual será o AGV a usar no projeto, tendo por base este capítulo, o AGV escolhido será entre o AGV Rebocador e o AGC *Under Ride*. Com as informações da capacidade de carga e a velocidade com que estes se deslocam, é possível ter uma noção da quantidade de AGV necessários e o tempo que irão levar para carregar e/ou descarregar as bagagens, o deslocamento dos AGV também ficou consolidado com as secções dos sistemas que os mesmos adoptam, sejam estes os sistemas de navegação, controlo ou de deslocamento.

Capítulo 3

Robôs móveis para tarefas logísticas

Neste capítulo, primeiramente, é revista a definição de robôs móveis e a sua funcionalidade. São apresentadas as aplicações, as vantagens e desvantagens da utilização dos robôs móveis. Por fim, serão apresentados alguns projetos sobre o mundo dos robôs móveis para tarefas logísticas, uns são de empresas que trabalham com robôs móveis e outros são retirados de artigos do IEEE.

3.1 Robôs Móveis

Um dos tipos de robôs com capacidade para realizar tarefas logísticas são os robôs móveis autônomos (AMR), mencionados no Capítulo 2.

Os AMR são capazes de se mover, tanto em ambientes externos como internos, de forma autônoma, ou seja, são robôs que devem ser capazes de perceber o ambiente à sua volta, tomar decisões sobre a melhor ação a ser executada e realizá-la, com ou sem intervenção humana, com o mínimo de erro. Isto requer, entre outras funcionalidades, a capacidade de definir uma sequência de ações a serem tomadas através de um conjunto de sensores (um exemplo é o sensor ultrassônico, com este é possível detetar obstáculos ou objetos que estejam perto do AMR).

Navegar de forma autônoma de um ponto a outro de um cenário é o essencial para qualquer robô concluir uma missão. Para isso, este deve ser capaz de planejar e seguir uma rota a fim de alcançar o ponto determinado como objetivo, detetando e desviando-se de obstáculos imprevistos [56].

3.2 Aplicações

Como os AGV podem fornecer um transporte eficiente e de forma econômica e, visto que isto é um elemento importante e comum na melhoria das tarefas de fábricas, podem ser aplicados a vários setores em projetos padrão ou personalizados para melhor atender aos requisitos de um setor. Geralmente têm como objetivo o transporte de diversos materiais, incluindo paletes, rolos, carrinhos, contentores, entre outros. Também são, muitas vezes, utilizados para o transporte de materiais perigosos ou prejudiciais para um ser humano, tais como, materiais radioativos, produtos químicos, entre outros [57].

As aplicações típicas dos AGV, além do transporte de matérias-primas, também incluem transporte de produtos em processamento e/ou acabados, apoio às linhas de produção, apoio à separação de materiais em tarefas de armazenamento e distribuição, entre outros.

Na indústria farmacêutica e química e locais que necessitam de requisitos especiais de limpeza, segurança e em que não seja suposto haver poluição são utilizados os AGV para o transporte de materiais para depósitos de armazenamento e para outras células ou estações de processamento.

Nos hospitais, podem ser usados no transporte de medicamentos, transporte de macas, roupas, refeições de pacientes, lixo, entre outros (ver Figura 3.1).



Figura 3.1: AGV num hospital [13]

Na indústria alimentar (bens de consumo), o AGV é, principalmente, utilizado para transportar matéria prima desde o *stock* até às máquinas e os produtos finais

até aos locais de expedição [58, 59].

Na logística, o AGV é utilizado para organização de armazéns, transporte de matérias primas ou produtos acabados, transporte de cargas específicas que não podem ser transportadas através de equipamentos comuns, seja por serem produtos de elevado volume/peso ou produtos muito delicados.

Na indústria automóvel também se podem encontrar vários exemplos, sendo alguns os seguintes: reboque de carrinhos, dispositivo auxiliar de montagem, fornecer ferramentas especializadas que devem ser trocadas e também transportar os materiais entre áreas dentro do processo produtivo.

Na indústria do papel e impressão o AGV pode mover rolos de papel, paletes, ou seja, fornecer toda a movimentação de material de rotina na produção e armazenamento [60, 61].

Recentemente, os parques temáticos têm utilizado o AGV, chamado sistema de passeio *trackless*. São passeios com AGV que usam sistemas *Local Positioning System* (LPS), Wi-Fi ou *Radio-frequency identification* (RFID) para se mover. A vantagem deste sistema é que o passeio pode executar movimentos aparentemente aleatórios, proporcionando uma experiência de passeio diferente sempre que o mesmo é realizado (ver Figura 3.2) [14, 62].



Figura 3.2: AGV no Parque Temático [14]

3.2.1 Vantagens e Desvantagens

Após o que foi apresentado anteriormente, é necessário avaliar os AGV em relação às suas vantagens e desvantagens para se perceber o quão óbvio e importante é

a sua utilização. Com a sua introdução no ambiente fabril é, automaticamente, reduzida a mão de obra no local. Porém a empresa terá de realizar um investimento inicial bastante alto para adquirir o AGV, ou seja, ao invés de realizar um pagamento contínuo com o pagamento de salário, seguros, impostos, entre outros, a empresa iria realizar um único pagamento que seria o custo do AGV [63, 64]. Como é óbvio, há custos acrescidos no futuro em certos aspectos, custos de energia, manutenção, entre outros. O custo de um AGV varia bastante consoante o tipo do mesmo, porém, em média estão entre os 50000 a 150000 euros [65].

Além deste, outro ponto importante de referir é a segurança que os mesmos oferecem: são programados de forma a serem seguros no espaço de trabalho com a ajuda de sensores que estão nele incorporados. Além disso, estes não correm o risco de cansaço e/ou cometerem erros por distração, com isto aspectos como precisão, eficiência e produtividade são elevados ao máximo o que torna os processos bastante rentáveis e a redução de custo/tempo de inactividade. Além disso, podem trabalhar em condições menos satisfatórias, seja por serem ambientes sujos, barulhentos, perigosos, etc.

Uma desvantagem que raramente é referida é o facto de que maior parte dos AGV comunica com os servidores via *wireless*; caso a empresa tenha problemas de ligação ou momentaneamente sem acesso não será possível obter os dados provenientes do AGV [66].

O investimento que é feito inicialmente também acaba por ser uma desvantagem se for visto a curto/médio prazo, pois a longo prazo acaba por não o ser se for obtido o retorno do investimento. As pequenas empresas, onde não haja tanto capital, irão sofrer mais com este problema do que propriamente as médias ou grandes empresas.

Em suma, no que diz respeito a vantagens, pode-se dizer que as mais relevantes e notórias para a empresa serão: a redução de mão de obra e redução dos custos de operação, maior segurança e maior produtividade. Em relação às desvantagens acabam por ser ligadas ao elevado investimento inicial, assim como ocasionais custos de manutenção, e há ainda pequenos pormenores onde o AGV necessita de ser melhorado, tais como: adaptação para tarefas que não são repetitivas (versatilidade), pois há um decréscimo de flexibilidade de operações; e também não são adequados para terrenos irregulares pois estes iriam dificultar a deslocação do veículo [67].

3.3 Aplicações de AGV / AMR

Nesta secção serão apresentados alguns projetos com AGV e robôs móveis. Visto que os robôs móveis são utilizados em áreas bastante distintas, foram seleccionados os que melhor se enquadram na área de logística. Com esta secção será

possível perceber que tipos de AGV se enquadram melhor nos projetos a desenvolver e quais os mais escolhidos.

3.3.1 Robô Móvel para Processo de Marcação em Tanques de Carga

Foi desenvolvido um robô móvel para Processo de Marcação em Tanques de Carga *Liquefied Natural Gas Carrier* (LNGC) que realiza as suas funções na parede. Estas funções são: marcação de linha, pré-marcação de parafusos e medição do nivelamento do solo. Tem navegação autónoma através de um GPS interno. O processo de marcação do tanque LNGC que não era 100% automatizado passou a ter uma precisão melhorada, reduzindo todos os possíveis erros humanos. Em relação à pré-marcação dos parafusos e medição do nivelamento, este é feito através de sensores laser instalados no robô móvel [15].

A distância entre o sensor e o alvo é medida por três sensores integrados no robô (ver Figura 3.3). Os dois sensores com melhor precisão serão os usados para calcular a posição central do robô, os dados das coordenadas são dadas através dos *softwares* RobotStudio e IGPS 3D. O robô segue com velocidade de 5 m/min e precisão de 15 mm até ao local de destino.

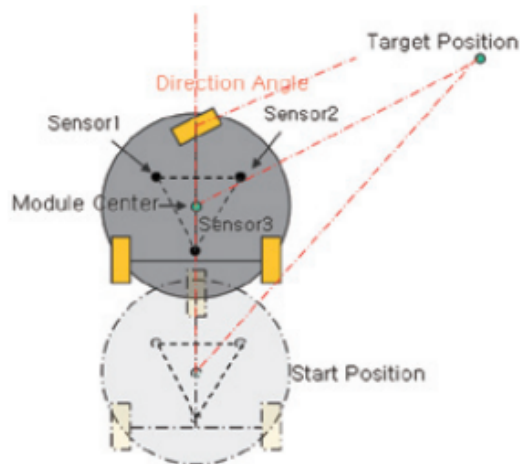


Figura 3.3: Técnica de localização do robô móvel [15]

3.3.2 Fetch Robotics

Um dos pioneiros a utilizar robôs móveis nesta área é a Fetch Robotics, que desenvolveu um AMR colaborativo, ou seja, um *Collaborative Mobile Robot* (CMR). É um robô móvel inteligente que auxilia humanos num espaço de trabalho compartilhado, para localizar, “seguir” e movimentar *stock* entre depósitos e instalações

de logística. O sistema inclui uma base de robô, anexos modulares e um sistema de base de dados.

Têm modelos no centro de distribuição da *DHL* (empresa de transporte) na Holanda onde os robôs ajudam a gerir toda a cadeia logística de peças de reposição, desde o recebimento do pedido até à entrega ao cliente. O uso dos robôs simplifica o transporte de materiais nos casos de ser ponto a ponto. Estes robôs utilizam a base *Freight* da própria Fetch (ver Figura 3.4) e foram projetados para atender ao setor de logística, mas são aplicáveis a outras áreas ou setores [16].



Figura 3.4: Robô colaborativo da Fetch Robotics [16]

Outro exemplo ainda da Fetch Robotics é na empresa RK Logistics onde a mesma implantou robôs para lidar com 30% a 50% dos itens nas instalações e usa robôs com sensores *RFID* para o rastreamento de *stock*.

3.3.3 Navegação e Localização de AGV num Armazém via Rede de Sensores Wireless

O objetivo deste trabalho é controlar todos os AGV através de sistema de comunicação *wireless*, o veículo será capaz de se comunicar via *Wireless Static Control Central Mode* (WSCCM). Neste método os veículos são controlados, bem como conduzidos por um centro de unidade de controlo principal (ver Figura 3.5), porém não podem comunicar uns com os outros.

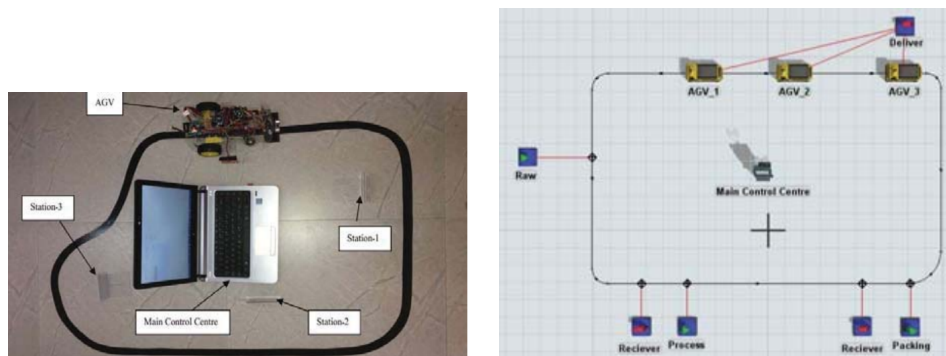


Figura 3.5: Projeto Desenvolvido, a) na Realidade, b) em Simulação [17]

Os AGV são usados em armazéns automáticos pelas suas características de orientação automática, posicionamento preciso, rota ideal e segurança. O principal objetivo do AGV é viajar entre as várias estações de trabalho manual para melhorar a produtividade, aumentar a automação e reduzir o custo da mão de obra nos armazéns. Os testes comprovaram que, após testar em cada parte do programa, o AGV consegue obter a informação para o próximo destino, encontrar a rota certa, detectar a estação, paralisar e informar a sua localização.

O principal foco deste projeto foi utilizar apenas um AGV solitário para transportar o material necessário. Tiveram em consideração que ao utilizar vários AGV ao mesmo tempo num processo de produção com muitas estações torna o projeto muito mais complicado, visto que no caso deste projeto não iriam ter comunicação entre eles mas apenas com o centro de unidade de controlo [17].

3.3.4 Navegação do Robô Via GPS e Desvio de Obstáculos

Este projeto é composto por um robô móvel cuja a sua navegação é controlada pelo sistema de posicionamento global (GPS) e um sistema de prevenção de obstáculos.

É um robô móvel (ver Figura 3.6) equipado com sistema de navegação GPS e sistema de prevenção de obstáculos com estrutura móvel de baixo custo, ou seja, módulo GPS e sensor sonar - esta combinação entre o GPS e o sonar irá determinar a posição e evitar os obstáculos. Assim o robô móvel pode navegar pelo *waypoint* (ponto intermediário do percurso) desejado e ao mesmo tempo aplicar as regras para evitar obstáculos. Para este projeto usaram um carro *Radio-controlled* (RC) como robô móvel [18].



Figura 3.6: RC Truck com GPS e Sonar [18]

Este projeto mostra que a navegação do robô móvel usando o módulo GPS pode ser construída através da combinação entre um carro/camião RC e um sensor sonar. O conceito do sistema pode ser aplicado a robôs móveis de maior escala com melhor precisão do sistema GPS e melhor sistema de detecção de obstáculos, como visão e telémetro (instrumento para medir distâncias) a laser [18].

3.3.5 InVia Robotics

A inVia Robotics, sediada na Califórnia, desenvolve uma variedade de AMR (ver Figura 3.7) que operam juntos para várias tarefas de depósito, reabastecimento de *stock*, contagem cíclica e verificações de itens. O processo é controlado via *software Robots as a Service* (RaaS), permitindo assim que as empresas otimizem os depósitos sem interromper o ecossistema operacional. Lior Elazary, *Chief Executive Officer* (CEO) da inVia, afirmou que os “robôs usam um sistema *Artificial intelligence* (AI) muito sofisticado com algoritmos de detecção o que faz com que os mesmos sejam capazes de se adaptarem ao ambiente do armazém existente e espera aumentar a produtividade de um armazém desde 300% a 500%” [68, 19].



Figura 3.7: Robô da InVia Robotics [19]

3.3.6 Robô Móvel Com Manipulador Usado Como Protector De Segurança

Este projeto consiste num robô móvel com um manipulador que foi desenvolvido para trabalhar como segurança. O robô patrulha o prédio de forma autónoma e verifica emergências como incêndios, intrusos, etc. Quando este detecta uma emergência, o robô informa um operador remoto que lida com a emergência operando remotamente o robô. O robô tem diversas funcionalidades, entre as quais: navegação autónoma, sistema de manipulação com realimentação de força (*force feedback*), sistema de visão por câmara operada remotamente com visão ampla. A velocidade deste robô é de 1 m/s. Em relação ao operador remoto, este pode, por exemplo, abrir portas através do manipulador e seguir a visão do robô na direção que o mesmo entender.

De forma a se entender melhor o problema, foi retirado o diagrama do próprio projeto que dá uma melhor percepção da forma como o robô comunica (ver Figura 3.8) [20].

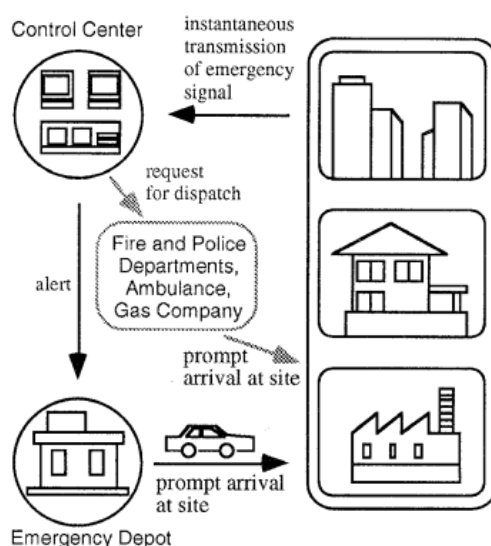


Figura 3.8: Centro de controlo e alertas [20]

Este projeto tem alguns problemas que são referidos no artigo: o elevado tempo que demora para que o robô consiga abrir uma porta; o campo de visão vertical que é de apenas 30° e, por fim, o robô apenas se consegue mover de forma autónoma se não tiver obstáculos pois não faz o desvio dos mesmos - sendo assim, apenas consegue mover-se em ambientes simples, como corredores ou espaços vazios[20].

3.4 Conclusão do Capítulo 3

Neste capítulo foi possível verificar em que principais áreas, e em que condições, é que os robôs móveis são utilizados, assim como as vantagens e desvantagens que estes trazem ao ambiente onde são inseridos. Como foi possível verificar os robôs móveis podem ser utilizados tanto em ambientes de empresas como fora das mesmas.

Esta solução manteve-se eficiente durante a pandemia, num momento em que muitos segmentos precisaram de continuar a produzir com menos funcionários nas fábricas em função da Covid-19 e a gerir processos de forma remota, porém os benefícios vão muito além da simples substituição da mão de obra humana.

Capítulo 4

Aplicações de Simulação para Robôs Móveis

Após o estudo de robôs móveis e de AGV é necessário decidir qual o software no qual será feito o projeto em questão. Para tal, foram pesquisados vários de forma a compará-los e decidir qual é o que tem maior viabilidade e oferece mais vantagens em relação aos restantes. Há várias vantagens que os softwares de simulação oferecem que são óbvias, tais como: o poder de planejar e prever riscos que possam aparecer na vida real ao testar ideias, poupar tempo, dinheiro e ter uma ideia prévia do impacto que o projeto poderá causar.

Neste capítulo serão analisados algumas aplicações de softwares de simulação de processos industriais e que suportam robôs móveis de forma a ser possível escolher o software que melhor corresponde às necessidades do projeto em questão, visto que será necessário realizar a simulação do transporte de bagagens num aeroporto.

4.1 FlexSim

O software FlexSim, lançado em 2003, desenvolvido pela *FlexSim Software Products, Inc.* intitula-se como o melhor na área de simulação 3D. Na página inicial pode-se verificar a seguinte frase *The most powerful, capable, and easy-to-use 3D simulation software. FlexSim lets you model, and improve, existing and proposed systems.*

É um software orientado ao objeto, ou seja, cada objeto tem uma lógica pré-construída pela execução de várias tarefas, de modo a que estes sejam semelhantes ao mundo real. É um software que utiliza pouco, ou nenhum, código dependendo do projeto. Visto que é um simulador, é possível criar vários cená-

rios “*What If*” com a possibilidade de importar desenhos *Computer-aided design* (CAD) personalizados e sem a necessidade de realizar alterações no mundo real. Além disso, contém uma ferramenta de fluxograma, fluxo de processos (*process flow*), para criar o modelo do projeto com base em diagramas de blocos (ver Figura 4.1), sendo assim um *software* que permite analisar, visualizar e melhorar processos do mundo real [21, 69]. Este é meramente um exemplo para se verificar o aspecto de um *process flow* no Flexsim.

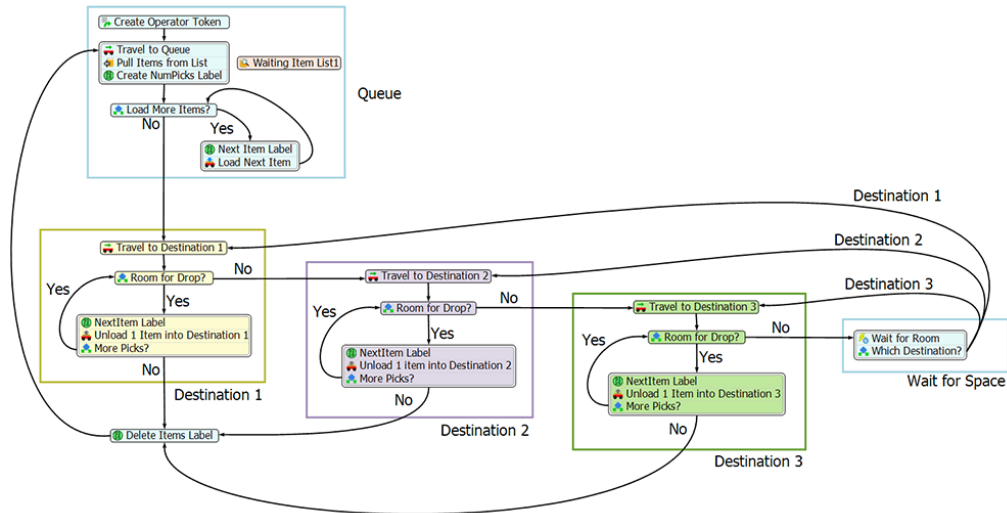


Figura 4.1: Process Flow Flexsim [21]

Os utilizadores deste programa podem aceder à vista do modelo (ver Figura 4.2) e arrastar os objetos 3D para onde desejam, de forma a realizar um diagrama de blocos com mais ou menos complexidade. Os utilizadores mais experientes têm opção de especificar parâmetros via programação em C++ ou no próprio FlexScript (linguagem de programação do Flexsim).

O Flexsim tem ainda uma ferramenta, *ExpertFit*, que consegue determinar precisamente qual a melhor probabilidade de distribuição que modela os dados que foram retirados do mundo real.

Curiosamente, no *website* deste *software*, é possível verificar que existe um separador com o nome “*Airport Simulation*” o que indica que, à partida, é um *software* preparado para o projeto a ser realizado, sendo assim uma vantagem [70]. Neste separador o caso de estudo incidiu sobre um transporte seguro, “*Transportation Security Administration*” (TSA), que permitiu chegar a soluções para configurar a linha ideal que incorporasse o processo “*TSA PreCheck*”. Este transporte em conjunto com o processo referido devem, de forma eficiente, acomodar vários tipos de passageiros, incluindo passageiros com deficiência e funcionários

/ tripulantes.

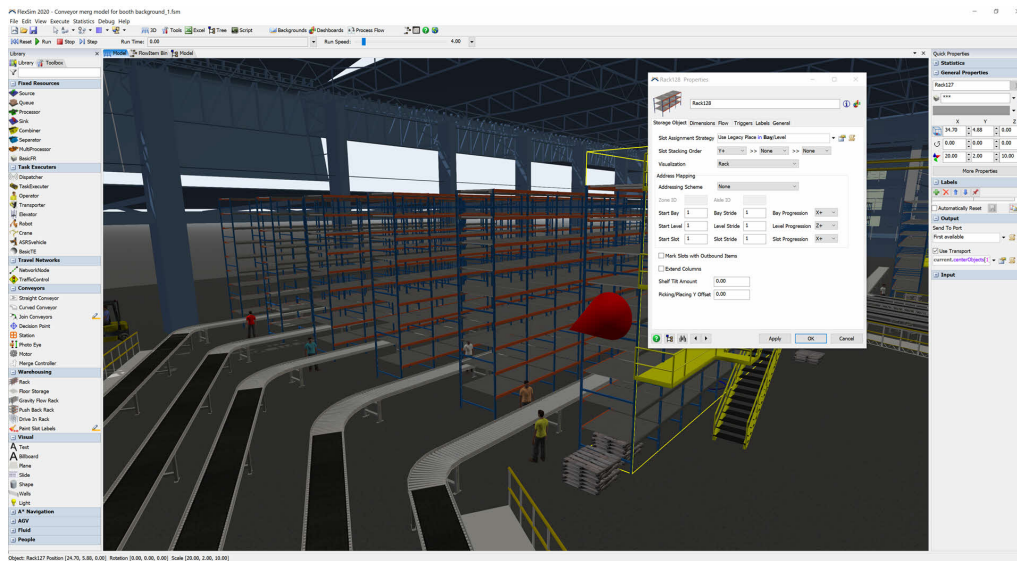


Figura 4.2: Ambiente gráfico do Flexsim [22]

4.2 Visual Components

Este *software*, lançado em 1999, é desenhado para suportar aplicações com simulação 3D, tem como principal objetivo tornar o projeto de fábrica e a tecnologia de simulação fáceis de usar e acessíveis a organizações de todos os tamanhos. Em dezembro de 2017, a empresa KUKA anunciou a aquisição da *Visual Components*. Os comportamentos físicos, desenvolvidos pelo mecanismo da *Nvidia PhysX*, podem ser simulados e visualizados, por exemplo: colisões e gravidade [71]. É um programa para projetos que incluem planeamento de *layout*, simulação de sistemas de produção, programação *off-line* e verificação de *Programmable Logic Controller (PLC)*.

Existe, ainda, um catálogo virtual, *eCatalog* (ver Figura 4.3), que permite colocar os objetos via *plug-and-play*, estes objetos encontram-se em blocos com propriedades pré definidas, porém é possível alterar algumas propriedades, tais como, dimensão, velocidade, cor, entre outras...

Em semelhança ao Flexsim, também é possível importar desenhos CAD através do conversor do próprio *software* e, de forma a reduzir a necessidade de se programar todos os componentes, existe um separador que permite definir, editar e manusear produtos e processos de produção.

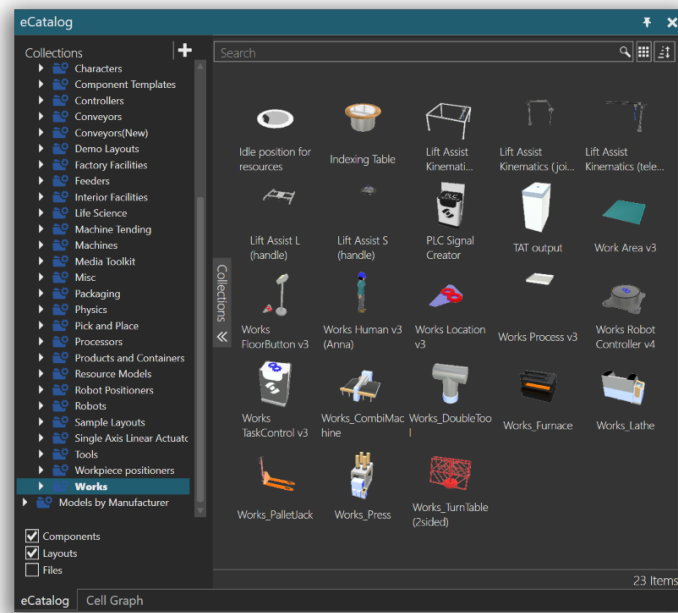


Figura 4.3: eCatalog no Visual Components [23]

Além disso, existe a possibilidade de editar o espaço em 3D de forma a facilitar a visualização e análise. Uma curiosidade sobre o *Visual Components* é que, caso o utilizador tenha a versão *Premium*, tem a possibilidade de ver o projeto em realidade virtual (ver Figura 4.4) [23].

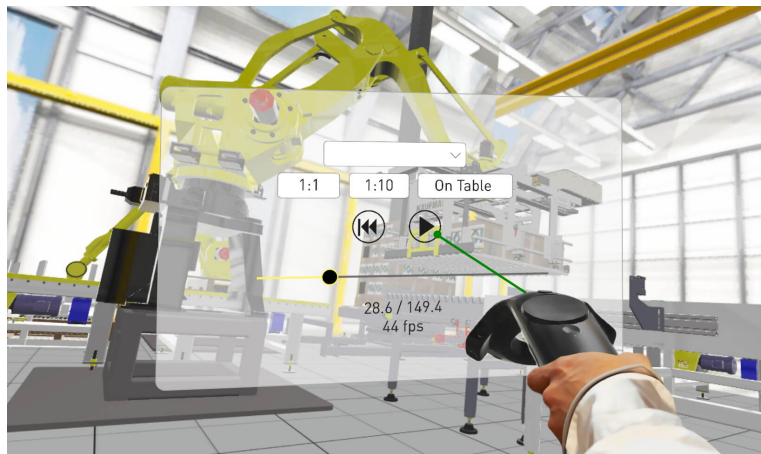


Figura 4.4: Realidade virtual no Visual Components [23]

4.3 Simio

O *software* Simio, lançado em 2008, permite realizar simulações em 2D e/ou 3D (ver Figura 4.5). Os objetos podem ser colocados no ambiente através de “*drag and place*”. Devido à “*inteligência*” do *software*, os objetos podem interagir entre eles não sendo necessária qualquer programação. Tem capacidade para gerar num só trabalho um projeto complexo ou de larga escala.

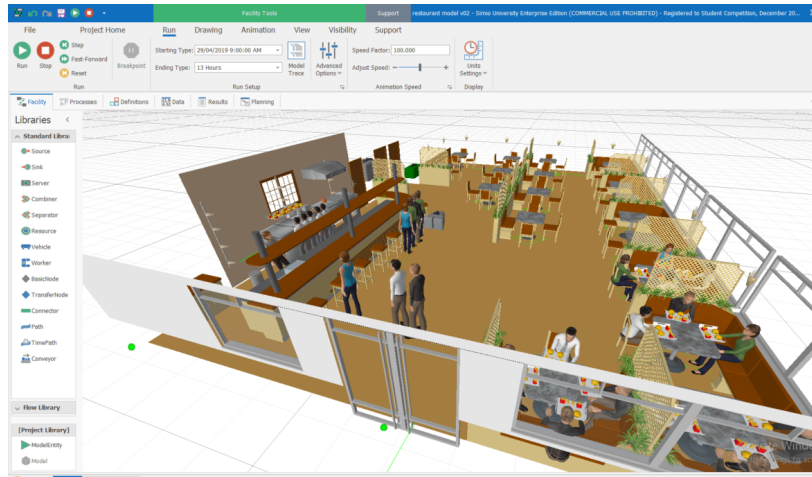


Figura 4.5: Exemplo de um projeto no Simio [24]

Tem uma precisão capaz de disponibilizar um modelo exato ao da vida real, isto deve-se ao facto de que os produtos deste *software* fazem parte do *Google Warehouse*, assim permite que haja um acesso rápido a uma enorme biblioteca de símbolos 3D disponíveis que podem adicionar realismo aos modelos de forma fácil e rápida. Desta forma, o Simio é uma ótima ferramenta para projeto nas seguintes áreas: ambiente fabril, saúde, aeroespacial, exploração mineira e engenharia industrial [72].

Em semelhança aos restantes, usa objetos inteligentes, que combinam uma série de lógicas e animações, que podem ser de bibliotecas de objetos pré-construídos ou personalizados, para representar, por exemplo máquinas, robôs, aviões, clientes, navios, médicos, etc. O ambiente em que é realizada a simulação do projeto pode ser feito em condições perfeitas ou em condições do mundo real [73].

Na página deste *software* é possível verificar um setor de “simulação e agendamento de transporte e logística” e, no *software* é possível testar vários cenários e, assim, ajudar a prever riscos que advêm de cada um dos cenários e permitir que as empresas estejam preparadas para qualquer acontecimento que possam inviabilizar os objetivos. Assim como os restantes, com a ajuda do *software* de si-

mulação, as empresas economizam tempo e dinheiro criando sistemas e processos mais eficientes [74].

4.4 Simul8

À semelhança dos *softwares* anteriores, o Simul8, lançado em 2012, é um *software* de simulação 2D que permite editar projetos com uma interface de *drag and place* totalmente focada na velocidade para construir e executar simulações, usa blocos de construção para simular e experimentar, com pouca ou mesmo sem necessidade de programação. É possível guardar blocos feitos pelo utilizador de forma a ser possível usar em vários projetos sem a necessidade de os construir várias vezes. Uma curiosidade acerca desta ferramenta é que permite editar, guardar e partilhar simulações num *browser*, permitindo que qualquer pessoa, dentro das que têm acesso ao projeto, execute e experimente sem a necessidade de fazer *download* [75].

Uma funcionalidade bastante interessante neste *software* é que permite que haja partilha com outras ferramentas, ou seja, todos os resultados podem ser exportados para análise ou partilha de dados para outras aplicações, incluindo Excel, *Google Sheets*, etc. A programação nesta ferramenta é à base de Visual Basic, porém é adaptado pelo Simul8 e tem o nome de Visual Logic.

A grande desvantagem deste *software* em relação aos anteriores é que este apenas permite a visualização 2D, como poder ser visto na Figura 4.6. No caso da simulação da figura é um projeto de engarrafamento porém o objetivo é mostrar qual é o ambiente gráfico da aplicação [76].

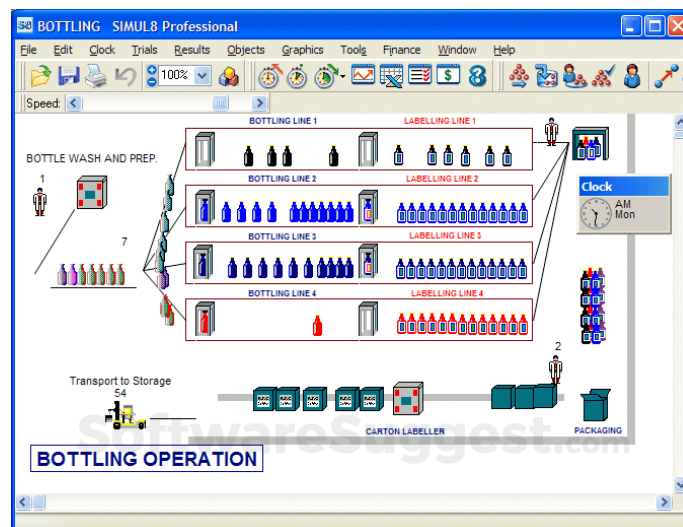


Figura 4.6: Exemplo de um projeto no Simul8 [25]

4.5 Anylogic

O *software* Anylogic, lançado no ano 2000, tem edição em 3D, que pode ser usado quer em Windows, como em Linux ou em macOS. É utilizado em várias indústrias, sendo algumas as seguintes: transporte, abastecimento, organização de armazéns, etc.

Por exemplo, dentro da indústria de transporte é possível verificar se a simulação do projeto tem viabilidade e avaliar os custos, transporte operacional e controlo com base na monitorização de dados em tempo real, estatísticas acumuladas e previsões baseadas em simulação [77, 78].

Esta ferramenta combina três métodos de simulação (ver Figura 4.7), sendo estes: método de dinâmica do sistema, baseados em agentes (*agent-based*) e eventos discretos (*discrete event*).

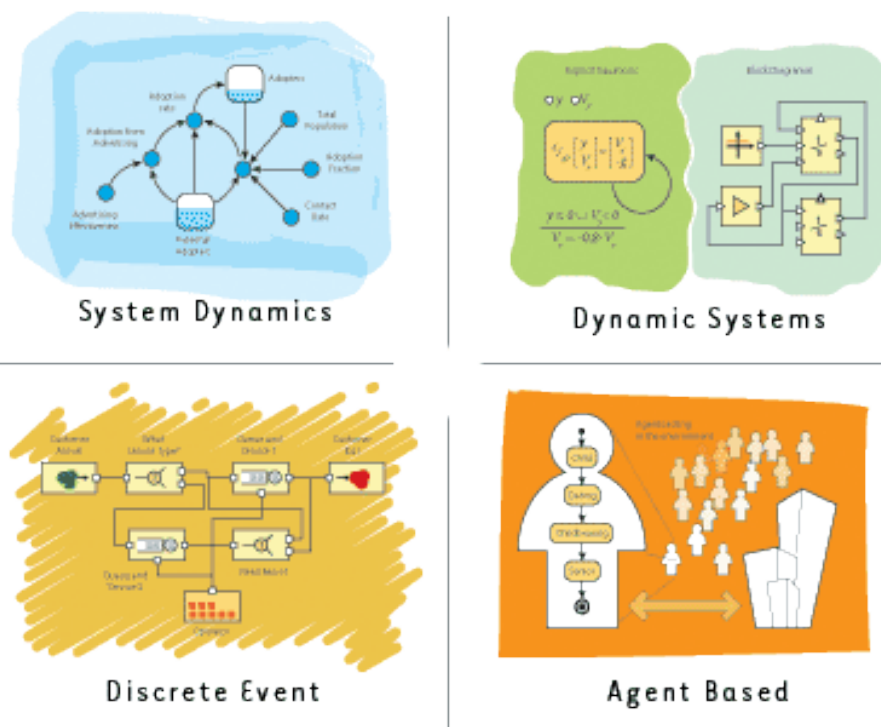


Figura 4.7: Métodos de simulação [26]

Os métodos de dinâmica do sistema e eventos discretos são abordagens de simulação mais antigas enquanto que a *agent-based* é uma abordagem mais recente. Enquanto que os métodos de dinâmica do sistema e eventos discretos têm uma abordagem *top-down* que descreve um sistema de uma perspectiva macro, ou seja vê o sistema de forma global e de seguida vê individual, o método *agent-based* é

uma abordagem inversa, ou seja, *bottom-up* que verifica intervenientes que atuam apenas uma vez no sistema assim como as interações dos agentes durante a simulação, com o objetivo de determinar o comportamento macro do sistema [79, 80]. Normalmente estes métodos são usados em separado porém, com o *software* Anylogic é possível combinar ambos os métodos e, assim, tornar a simulação mais completa, eficiente e, desta forma, facilitar a sua utilização.

De forma resumida, o método de eventos discretos adequa-se quando o objetivo é a tomada de decisões e previsões, ou seja é usado a nível tático ou estratégico, o método dinâmico é mais usado para estudar o comportamento do sistema a longo prazo, logo de forma estratégica, e o método *agent-based* é mais usado para identificar a interação e operação entre entidades de maneiras mais realistas e flexíveis, ou seja, é focado nas interações que ocorrem nos sistemas [81].

4.6 Escolha do Software

Nesta secção foi realizada uma tabela, ver Tabela 4.1, com as funcionalidades dos diferentes *softwares* descritos anteriormente.

Tabela 4.1: Tabela comparativa das aplicações de *Softwares* para simulação analisadas

Funcionalidades	Flexsim	VisualComponents	Simio	Simul8	Anylogic
Edição 3D	x	x	x		x
Importar objetos CAD	x	x		x	
Utilização de robôs móveis	x	x	x		x
Três métodos de simulação	x		x	x	x
Análise de dados em gráficos	x	x	x	x	x
Planeamento de trajetórias de AGV	x		x	x	x
Versão grátis	x		x		x
Manipulação Direta	x	x	x	x	x

Dentro das aplicações de *software* analisadas e descritas, é possível verificar que, à exceção do *software* Simul8, todos permitem realizar a simulação em 3D. Porém, após algumas pesquisas, foi possível verificar que o *software* Anylogic é configurado em 2D porém para ter edição 3D teria de se estar sempre a introduzir vistas ou câmeras em diversos locais da edição, o que seria um atraso e dificuldade acrescida na aprendizagem e utilização do mesmo. Sendo assim, estes *softwares* não são os mais adequados.

Com a possibilidade de introdução de desenhos/objetos CAD, os *softwares* Flexsim e VisualComponents estão um passo à frente em relação aos restantes, em específico ao Simio, pois este não tem qualquer outra desvantagem aparente a não ser a mencionada anteriormente; porém, visto que será necessário utilizar

objetos específicos, como bagagens, aviões e até os AGV, esta será uma ótima ajuda na construção do *layout* do projeto.

Entre os *softwares* Flexsim e VisualComponents não é possível verificar-se qualquer diferença a nível de funcionalidades, além do planeamento de trajetórias de AGV e do facto do VisualComponents apenas utilizar dois dos três métodos de simulação descritos anteriormente: dinâmica do sistema, eventos discretos e *agent-based*. O VisualComponents não utiliza a abordagem *agent-based*. Também se percebeu que o VisualComponents não oferece versão gratuita a não ser um *free trial*, como o objetivo não é a utilização de uma única vez teria de se pagar pelo *software*. Enquanto que o *software* FlexSim, apesar de ter a versão gratuita apenas disponibiliza um limite de 100 objetos por projeto.

Após todas as comparações feitas, seguiu-se com o *software* FlexSim visto que além de ser o que melhor se enquadra no projeto, o ISEP também adquiriu licenças com o uso de objetos ilimitados do mesmo, o que resolveu o problema do limite de objetos imposto pela versão gratuita.

4.7 Conclusão do Capítulo 4

Após a descrição dos *softwares* pesquisados e estudados, é possível verificar que, apesar de todos serem de simulação, há várias formas de o fazer e foi necessário decidir qual é o que melhor se enquadra neste projeto. Além do ponto tecnológico, também foram tidas em conta as condições necessárias para obter o mesmo, tanto a nível financeiro como os diferentes “pacotes” dentro de cada um, ou seja, versão gratuita, premium, etc...

Assim, tal como mencionado na última secção deste capítulo, foram comparados e analisados todos os *softwares* descritos nos pontos fundamentais para o projeto e decidiu-se utilizar o *software* Flexsim.

Capítulo 5

Aeroporto Francisco Sá Carneiro

Neste capítulo será apresentado o Aeroporto Francisco Sá Carneiro, aeroporto este que será posteriormente modelado e utilizado para a simulação. Primeiramente é apresentado o layout do aeroporto, de seguida os trajectos, tanto dos aviões como dos AGV e, por fim, os fluxos dos mesmos.

5.1 Layout do Aeroporto

O Aeroporto Francisco Sá Carneiro (OPO) (ver Figura 5.1) localiza-se no norte de Portugal, no distrito do Porto, mais especificamente, nos concelhos de Matosinhos, Maia e Vila do Conde. É o segundo maior aeroporto de Portugal em questões de tráfego e a principal porta de entrada, na região norte e na Eurorregião Galiza-Norte de Portugal.

O aeroporto foi fundado no final dos anos 1940, mas foi, várias vezes, renovado desde então e atualmente é um edifício moderno com capacidade para, até, 16 milhões de passageiros por ano [82, 83].

Empresas como a Iberia, Lufthansa, Transavia e TAP voam regularmente de e para este aeroporto, bem como companhias aéreas de baixo custo, como a Ryanair, Clickair, Transavia e Easyjet.

Apesar da situação atual ser de 20 movimentos/hora e com 40 lugares de estacionamento (35 aeronaves de passageiros mais 5 de carga), o Aeroporto está a desenvolver um plano de expansão, de modo a garantir uma capacidade de processamento de 32 movimentos/hora. Além disso, a infraestrutura estará preparada para receber a estação de transporte ferroviário [84].



Figura 5.1: Vista aérea do Aeroporto Francisco Sá Carneiro

Na Figura 5.2 é possível observar as pistas (17/35) onde os aviões aterram e/ou descolam e os 4 caminhos de circulação de aeronaves ao local onde é feita a carga e descarga dos passageiros e das bagagens - da esquerda para a direita, estas ligações são denominadas por: B, C, D e F. Na parte inferior da imagem também é possível ver o terminal.

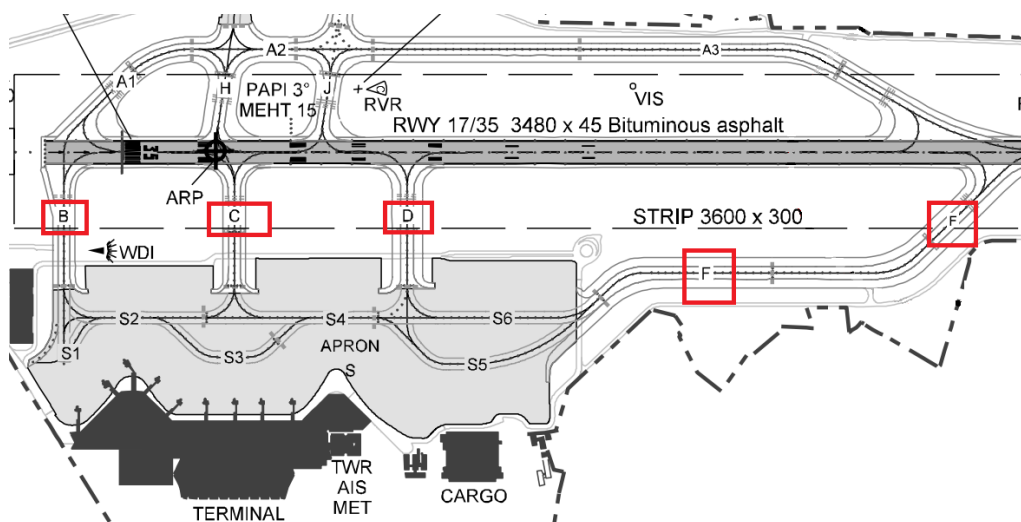


Figura 5.2: *Layout* do Aeroporto Francisco Sá Carneiro

5.2 Trajetos dos Aviões e AGV

O Aeroporto Francisco Sá Carneiro tem uma pista com quase 3,5 km, 3480 metros para ser exato, a esta pista existem várias ligações aos locais onde são embarcadas/desembarcadas os passageiros (PAX) e bagagens, ou seja os *stands*. Existem 43 locais onde os aviões podem estar estacionados e 9 “mangas” onde os aviões podem encostar para carregar e descarregar.

As “mangas” têm o termo técnico de pontes telescópicas. O local onde os aviões ficam posicionados para assistência chama-se *stand* (ou posição) e são identificados por letra + número. Podem ser posições de contacto com a aerogare (através das pontes telescópicas) ou remotas (acessíveis através de transportes - autocarros/ *minibus*, transporte VIP ou aviação executiva).

Os aviões, dentro do aeroporto, devem aceder a um dos *stand*: para assistência, caso estejam para realizar uma partida (saída do aeroporto); para descarregar, caso estejam a realizar uma chegada (entrada no aeroporto). Estes acessos (zona S da Figura 5.2) podem ser através de qualquer um dos canais (ver Figura 5.3) B, C ou D (este último não aparece na figura mencionada).

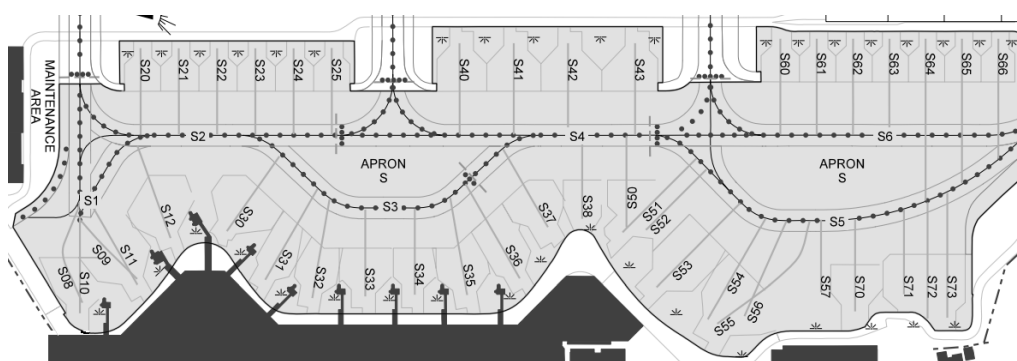


Figura 5.3: Trajetos dos aviões dentro do aeroporto

Em relação aos AGV, estes devem fazer as suas viagens entre o terminal do aeroporto e os respectivos *stands* de forma a deixar/recolher as bagagens. Estes devem deixar as bagagens que chegam no quadrado onde está pintado, a laranja, na Figura 5.4 e carregar as bagagens para o avião no quadrado pintado a azul. O percurso desenhado a vermelho é o percurso dos AGV, estes só entram nos *stands* caso haja bagagens para recolher ou entregar, senão circulam no percurso principal. Caso estes recolham as bagagens no ponto do terminal, devem transportar até ao *stand* pretendido para o avião descolar e deixá-las lá. Caso o AGV vá recolher as bagagens ao avião que esteja num dos *stand* deve transportar até ao local do terminal e deixá-las lá.

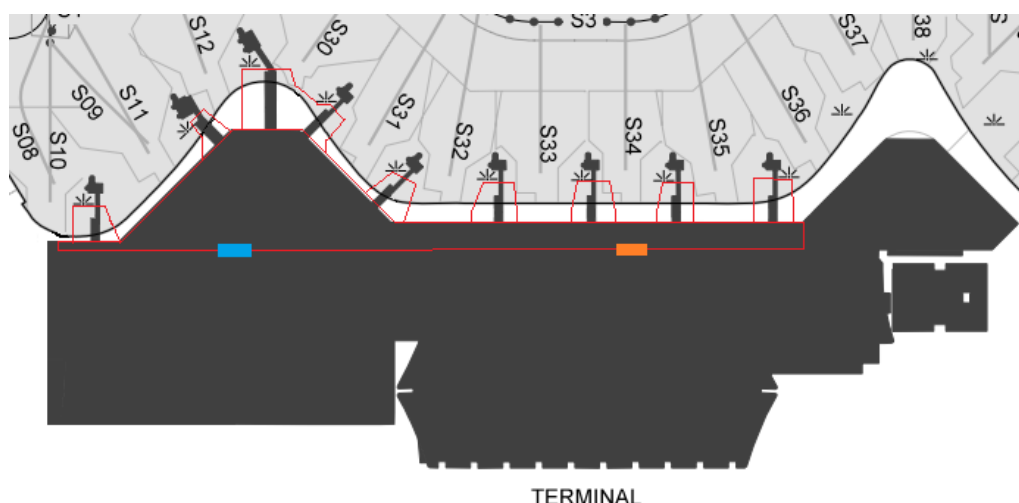


Figura 5.4: Trajetos dos AGV dentro do aeroporto

5.3 Fluxo de Aviões

O número de voos que passam no aeroporto não é uma variável fixa, ou seja, varia todos os dias e ao longo do dia.

Há dois tipos de voos: voos de carga e voos de passageiros. Houve acesso a uma informação confidencial com as estatísticas, durante o mês de setembro de 2016, no Aeroporto, e foi possível constatar que, nesse mês, houve entrada de 114 voos de carga e 943 voos de passageiros e saída de 115 voos de carga e 944 voos de passageiros. Logo, apesar de haver voos de carga, estes são em menores quantidades. No total, houveram 1057 voos de entrada no aeroporto e 1059 voos de saída do aeroporto.

Por mera curiosidade, a companhia com mais voos de passageiros no mês em análise foi a Easyjet que realizou 404 voos de entrada e mais 404 voos de saída e, de seguida, a Transavia que realizou 246 voos de entrada e 246 voos de saída.

5.4 Fluxo de Passageiros e Bagagens

No mês referente a este documento é possível verificar que foram transportadas 89 486 bagagens. Importante referir que, apenas por parte da EasyJet foram transportadas 32 403 bagagens, ou seja, mais de 1/3 do número de bagagens que foram transportadas naquele mês, no Aeroporto Francisco Sá Carneiro.

Deste número total de bagagens é, também, possível verificar que houve o transporte, em média, de 79 bagagens por voo e de 0,5 bagagens por pessoa. Em relação ao número de pessoas por voo este é, em média, de 152 pessoas.

Como é óbvio, estes valores variam todos os meses e consoante a companhia e a capacidade do meio de transporte, neste caso, o avião.

5.5 Regras de Segurança

Como em todas as situações há regras e, nos aeroportos, não é exceção logo devem ser respeitadas. Primeiramente, nada deve, em momento algum, passar por baixo do avião, aliás é feito um perímetro imaginário (ver Figura 5.5) onde os AGV não devem passar. Outra regra que os AGV têm de respeitar é a não obstrução dos camiões de abastecimento de combustível (cilindro laranja), estes são prioritários aos AGV.

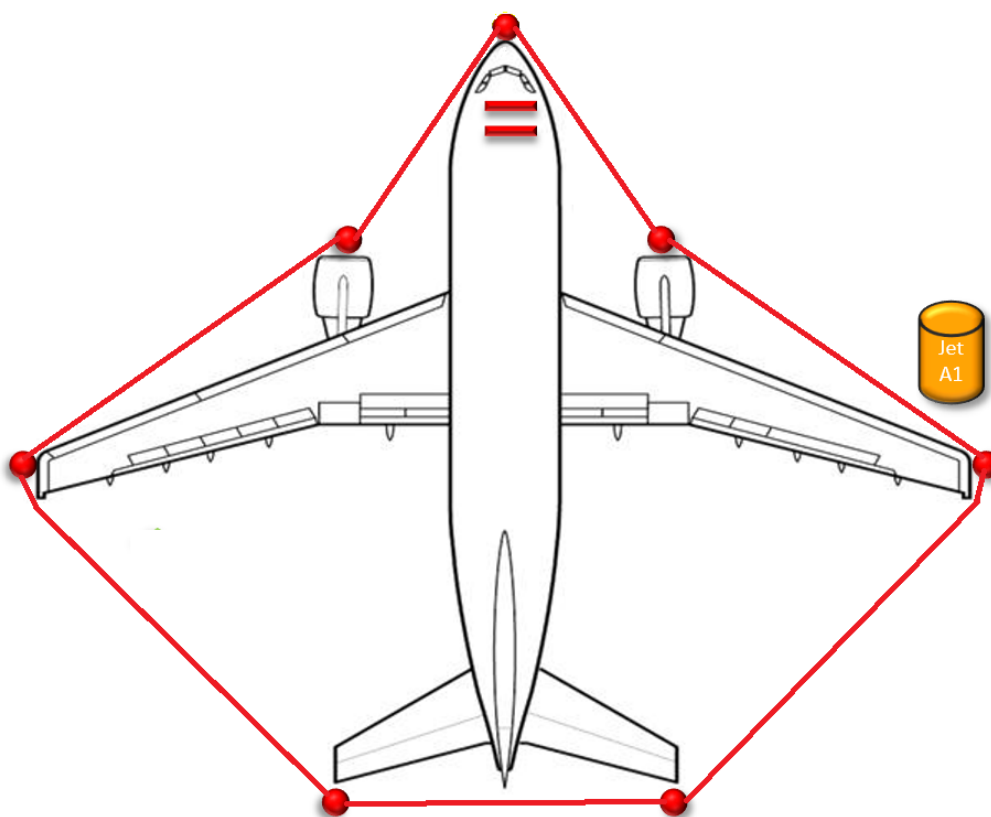


Figura 5.5: Perímetro de segurança de um avião[27]

O processo para a chegada de um avião é o seguinte, primeiro chega um avião e após a chegada deste é que os AGV se deslocam até ao *stand* onde o avião se encontra. Esta regra é muito importante para a implementação visto que vai ser avaliado o tempo de espera que os passageiros esperam pelas bagagens, logo este tempo que os AGV irão levar até chegar ao *stand* do avião é algo incontornável.

Quando chegam ao local, primeiro é feita a descarga das bagagens e transportadas até ao terminal e apenas após terminada a descarga é que se inicia o processo de carregamento de bagagens, desde o terminal até ao avião.

Pela perspectiva da Figura 5.5, a entrada e saída de bagagens é sempre feita pelo lado direito do avião e pelo lado esquerdo é feita a circulação dos passageiros. Na Figura 5.6 é possível verificar as regras de circulação. Do lado direito do avião será, também, feito o carregamento de combustível. Além disso é possível verificar a *Ground Power Unit* (GPU), equipamento que serve para fornecer energia aos aviões quando estes se encontram estacionados no aeroporto.

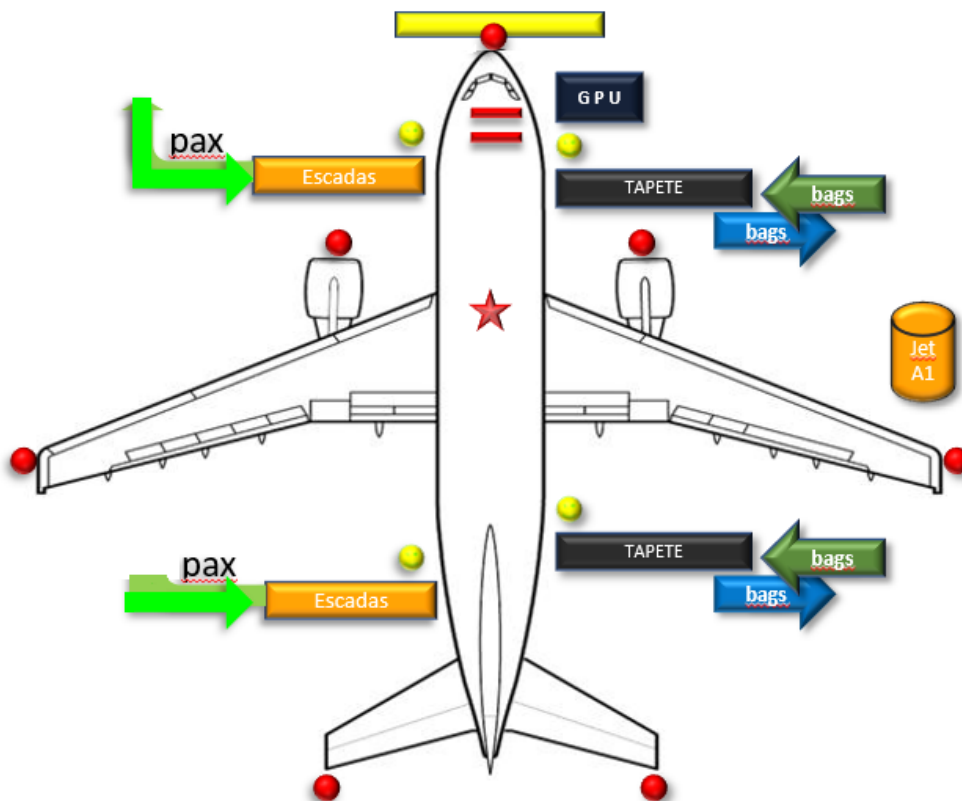


Figura 5.6: Circulação de bagagens e passageiros no avião [27]

5.6 Conclusão do Capítulo 5

Neste capítulo foi possível aprender sobre o Aeroporto Francisco Sá Carneiro, em especial, sobre os dados mais relevantes para o projeto: a sua capacidade, locais de embarque/desembarque e principais caminhos de circulação dos aviões e AGV dentro do aeroporto.

Além disso, foi possível ficar a saber valores concretos, durante um mês de funcionamento no aeroporto, que serão muito importantes para a simulação deste projeto, tais como: bagagens por voo, bagagens por pessoa, média de voos por dia.

Por fim, foram expostas algumas regras de segurança que devem, sempre, ser respeitadas no processo da chegada de um avião.

Com estes dados é possível construir um modelo de simulação que seja o mais próximo possível da realidade e trabalhar com valores reais de forma a se ter uma simulação muito próxima da realidade.

Capítulo 6

Modelação do Aeroporto

Neste capítulo é explicada a forma como foi desenhado o layout e, de seguida, programada a simulação no Flexsim, de forma que fosse obtido o resultado esperado para este projeto. É explicado passo a passo o process flow criado.

6.1 Importação e Criação do Layout

Para que este projeto tenha um aspecto mais realista foi necessário arranjar objetos relativos a um aeroporto, tais como: aviões, bagagens, escadas para embarque ou desembarque de PAX e os próprios caminhos para que estes não se desloquem dentro do aeroporto sobre o vazio. Para tal, recorreu-se a vários *Websites* de objetos 3D [85, 86, 87, 88], o objetivo foi encontrar os objetos pretendidos dentro dos formatos suportados pelo *software*, ou seja, .3ds, .dae, e .dwg.

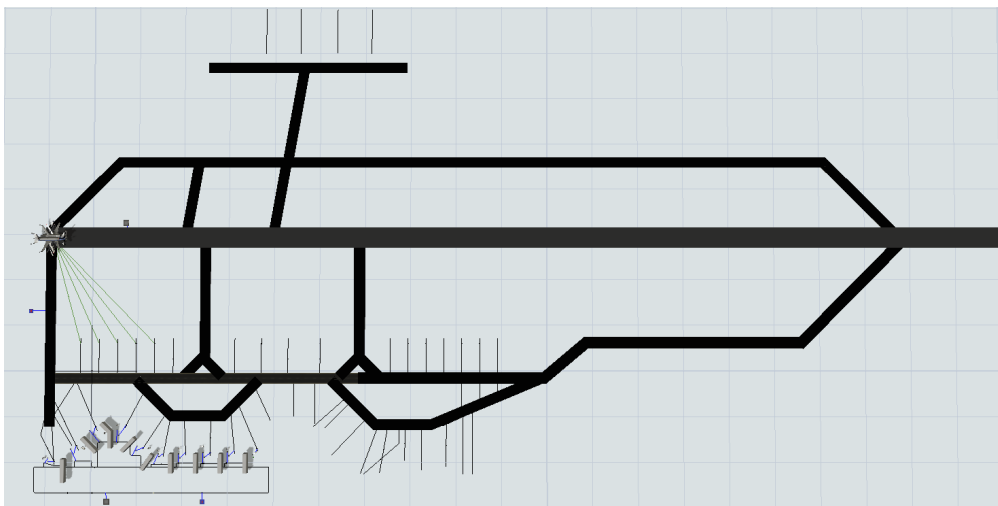
Enquanto que nos objetos relativos aos aviões (escadas de acesso ao avião e os caminhos do aeroporto) foi apenas associar ao objeto pretendido a mudança de visual, para as bagagens (ver Figura 6.1) foi necessário criar um “*Flow Item Bin*” para que estas pudessem ser criadas e usadas como um “*FlowItem*”. Ou seja, um objeto que será usado desde o início da simulação até ao fim da mesma.



Figura 6.1: Aspeto de uma bagagem

6.2 Mapeamento 3D

Para a criação do *layout* do Aeroporto Francisco Sá Carneiro foi visto um documento da Navegação Aérea de Portugal (NAV). Desta forma, conseguiram-se medidas mais próximas das reais (desde as pistas até ao terminal e a distância entre *stands*). Este documento ajudou a ter uma melhor perceção dos caminhos onde os aviões poderiam circular, sendo possível verificar o resultado do *layout* na Figura 6.2.

Figura 6.2: *Layout* feito no Flexsim

Ao longo dos percursos, tanto dos aviões como dos AGV, são necessários diversos pontos de controlo (tanto para os aviões como para os AGV). Com estes pontos evitam-se colisões e obtêm-se pontos de referência para as viagens. A Figura 6.3 (ampliação do *layout* feito no Flexim) mostra em detalhe o percurso que os AGV realizam para os *stands* (ida e volta).

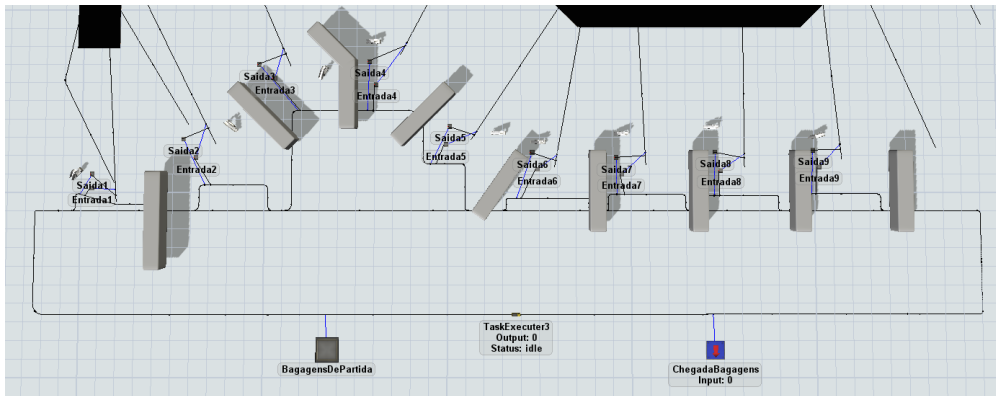


Figura 6.3: *Layout* do percurso dos AGV

Existe um percurso principal e os devidos “desvios” que os AGV necessitam de realizar para chegar aos *stands* dos aviões caso seja para carregar ou descarregar bagagens nos mesmos.

Ambas as figuras permitem ver *queues*. Um dos exemplos é o objeto “BagagensDePartida” situada na parte inferior esquerda da Figura 6.3. Uma *queue* é um objeto que permite armazenar itens - no caso deste projeto irão armazenar bagagens.

Para simular as saídas de bagagens (entrada no terminal do aeroporto ou saída de bagagens do aeroporto) foram utilizadas *sinks* que não é mais do que uma atividade que destrói *tokens*, removendo todos os dados que tenham armazenados até lá chegar. Um *token* é um objeto que consegue fluir através das atividades num *process flow* enquanto está a ser feita a simulação.

6.3 Criação de Grupos

De forma a ser possível utilizar vários pontos de controlo em simultâneo, é necessário associá-los a grupos. Portanto, foram criados 3 grupos principais: Grupos de Localização, Grupos de Transportes e Grupos de Equipamentos. Foi criada uma tabela (Tabela 6.1) onde é possível verificar como está disposta esta criação dos grupos e a que se refere cada um deles.

Tabela 6.1: Tabela referente aos grupos criados no Flexsim

Localização	CargasAvião	Locais onde os aviões podem carregar as bagagens
	DescargasAvião	Locais onde os aviões podem descarregar as bagagens
	CargasAGV	Locais onde os AGV podem carregar as bagagens
	DescargasAGV	Locais onde os AGV podem descarregar as bagagens
Transportes	Plane	Grupo referente aos aviões
	AGV	Grupo referente aos AGV
Equipamentos	ChegadaBags	Locais onde os Aviões deixam as bagagens que chegam ao Aeroporto (BagsIn)
	PartidasBags	Locais onde os AGV deixam as bagagens para saírem deste Aeroporto (BagsOut)

6.4 Características dos AGV

Os AGV serão os responsáveis pelos carregamentos e descarregamentos de bagagens entre o terminal e os aviões. Para tal, foi necessário configurar as suas propriedades, tais como, a velocidade a que este se desloca em diferentes situações (carregado ou vazio), a capacidade de carga e o tempo para carregar/ descarregar bagagens.

Em relação à velocidade do AGV, esta é apresentada em metros por segundo (m/s) e foi necessário configurar a mesma para estas situações: velocidade em reta, velocidade nas curvas, velocidade quando está carregado e velocidade quando está descarregado (ver Figura 6.4). O tempo que um AGV irá demorar para carregar ou descarregar uma bagagem é de 10 segundos.

DefaultAGV		
	Empty	Loaded
Acceleration	2	1
Deceleration	2	1
Forward Speed		
Straight	2	1
Curved	1	0.50
Spur	2	2
Reverse Speed		
Straight	2	2
Curved	1	0.50
Spur	2	2

Figura 6.4: Propriedades das velocidades dos AGV

A capacidade de carga do AGV é definida dentro de uma atividade, pelo que será explicado no fluxo de AGV. Porém o limite máximo, imposto no projeto, que um AGV pode transportar é de 20 bagagens, por questões de volume, visto que uma bagagem de porão tem como dimensão máxima total ($a+b+c$) até 158 cm (ver Figura 6.5), ou até 300 cm se os PAX pagarem extras.

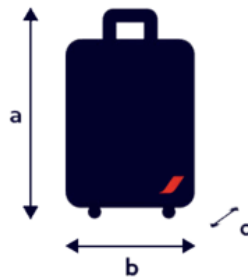


Figura 6.5: Dimensões de uma bagagem de porão

Considerou-se, para o projeto, que os AGV não iriam ter qualquer obstáculo. Os AGV demoram 10 segundos para cada carga, e descarga, de bagagens. Foi definido que não existirão erros nestas tarefas. O AGV escolhido para este projeto foi o *Under Ride*.

6.5 Tipos de Ligação

Existem dois tipos de ligações entre objetos, tipo A e tipo S. Nas ligações do tipo A há um *Input* e um *Output* enquanto que uma ligação tipo S é usada para fazer com que haja um ponto de referência abstrato entre dois objetos de forma a que estes comuniquem. Uma ligação do tipo S é bastante utilizada para a ligação do centro dos objetos. Assim, no *process flow* é possível utilizar o comando *.center.Object[1]*, por exemplo, nas atividades *Load* e *Unload* de objetos.

6.6 Atividades Mais Utilizadas

Nesta secção serão explicadas para que servem e como são programadas as atividades mais utilizadas ao longo do *Processflow*.

6.6.1 Assign Label

Uma atividade de *Assign Label* (ver Figura 6.6) cria ou altera o valor de uma *label* (variável). Ao utilizar esta atividade podem surgir duas situações: se for utilizada para uma *label* nova, esta é criada; se for utilizada e essa *label* já existir, apenas o seu valor irá ser alterado.

Neste projeto é utilizada para três situações: Associar *label* aos grupos de localização, incrementar a *label* e dar *reset* à *label* para que esta volte a ficar com valor 1.

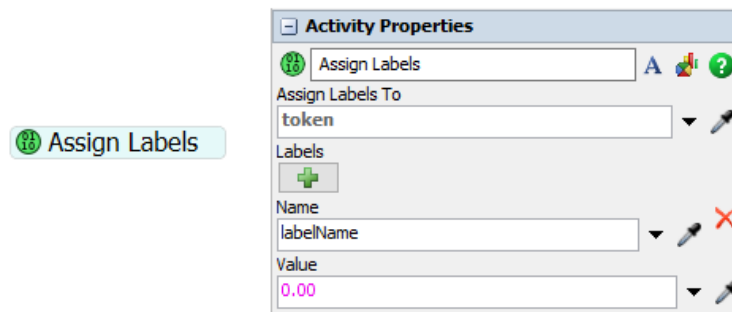


Figura 6.6: *Assign Label*, a) Atividade, b) Propriedades *default*

6.6.2 Wait For Event

Uma atividade de *Wait For Event* (ver Figura 6.7), tal como o seu nome indica, irá esperar que um determinado evento aconteça. No caso do *processflow* irá reter um *token* até que o evento aconteça.

Neste projeto foi utilizado para: bagagens darem entrada numa *queue* e para que a *queue* atinja um valor determinado.

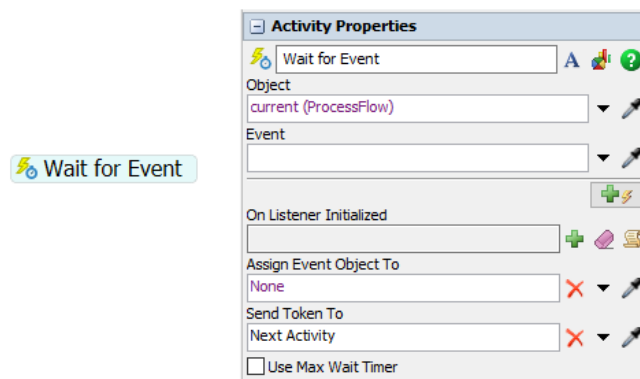


Figura 6.7: *Wait For Event* a) Atividade, b) Propriedades *default*

6.6.3 Decide

A atividade *decide* (ver Figura 6.8) enviará um *token* para uma de duas, ou mais, atividades possíveis com base nas condições que forem definidas. Por outras palavras, a atividade *decide* determina a próxima atividade para a qual um *token* deve ser enviado.

Neste projeto foi utilizado para: verificar se uma *queue* já atingiu um determinado valor, verificar se existem itens na *queue*, ou se está vazia, e verificar se há mais locais dentro de um grupo.

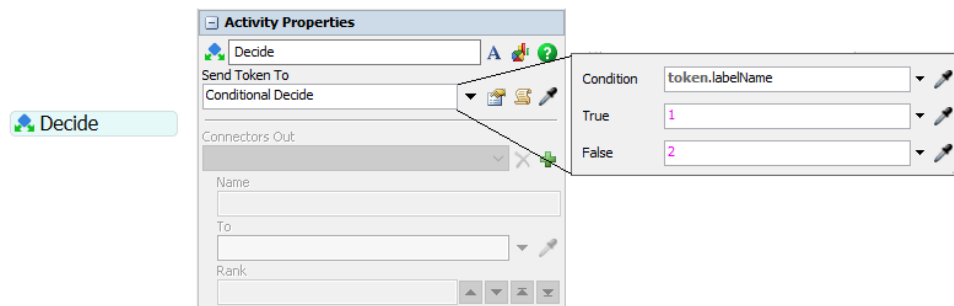


Figura 6.8: *Decide*, a) Atividade, b) Propriedades *default*

6.6.4 Pull/Push From/To List

As atividades *Pull From List* e *Push To List* (ver Figura 6.9) irão puxar, ou enviar, valores de, ou para, uma lista.

Neste projeto foram criadas duas listas globais, uma lista referente à partida de bagagens e uma lista referente à chegada de bagagens. Logo foram utilizadas

as duas atividades para as duas situações, chegada de bagagens e partidas das mesmas.

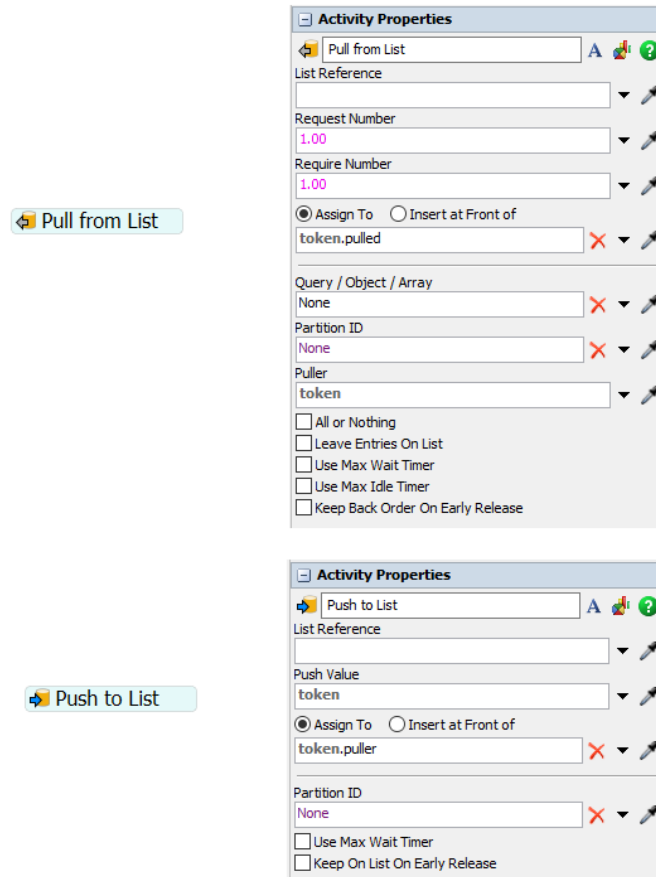


Figura 6.9: *Pull/Push From/To List*, a) Atividade, b) Propriedades *default*

Foi, ainda, dado *check* nas opções “*All or Nothing*” e “*Use Max Wait Timer*” ou na opção “*Use Max Idle Timer*”. A opção “*All or Nothing*” faz com que nenhum valor seja retirado da lista até que o *Require Number* seja atingido. Caso contrário (esta opção não esteja escolhida), os valores iriam ser puxados assim que estivessem disponíveis. A opção “*Use Max Wait Timer*” é o tempo máximo que esta atividade irá estar à espera dos itens, caso não aconteça irá seguir para outra atividade definida dentro desta opção. Enquanto que a opção “*Use Max Idle Timer*” é o tempo em que o objeto se encontra inativo, caso atinja este tempo irá seguir para a atividade definida.

6.6.5 Run Sub Flow

A atividade *Run Sub Flow* (ver Figura 6.10) é utilizada para gerar um *Sub Flow*. O *token* inserido nesta atividade será o *token* “pai” dos que serão gerados dentro do *sub flow*.

Neste projeto, os *Sub Flows*, foram utilizados para realizar as tarefas de *load* e *unload* de bagagens de forma a que o *token* esperasse por todas as bagagens para seguir para a próxima atividade.

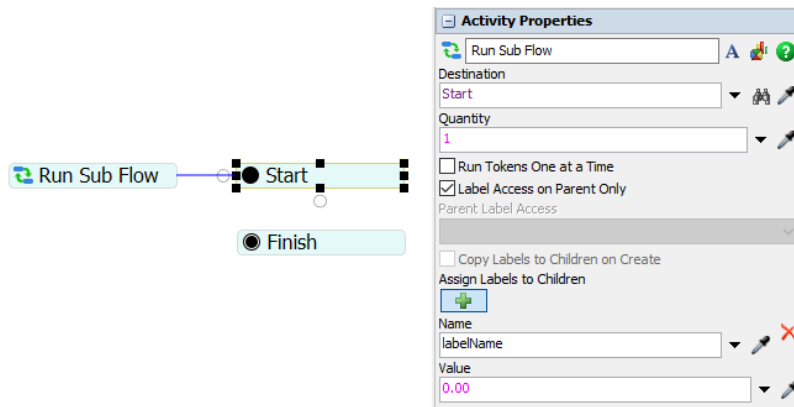


Figura 6.10: *Run Sub Flow*, a) Atividade, b) Propriedades *default*

Seguindo o exemplo da figura, o *token* filho terá o nome de *labelName*. O *token* pai não surge, porém, supondo que este se chama *mainLabel*, dentro do campo *Value* teria de se introduzir *token.mainLabel[creationRank]*.

Assim, o *token* pai (*mainLabel*) permanecerá nesta atividade até que todos os seus *tokens* filhos (*labelName*) tenham concluído o sub processo. Assim que os *tokens* filhos terminem, o *token* “pai” segue para a próxima atividade. Este tem de iniciar, sempre, por uma atividade de “*Start*” e terminar com uma atividade de “*Finish*”.

6.6.6 Sources

Uma *source* é utilizada para criar *flow items* (por exemplo, as bagagens). Cada *source* cria uma classe.

Neste projeto foram utilizados dois tipos de *source* (ver Figura 6.11): *Schedule Source* e *Event-Triggered Source*. Na *schedule source* é possível determinar em que momento irão surgir os objetos e quantidade dos mesmos. Enquanto que a “*event-triggered source*” apenas ativa quando um evento acontece e é possível associar um *token* aos itens desta *source*.

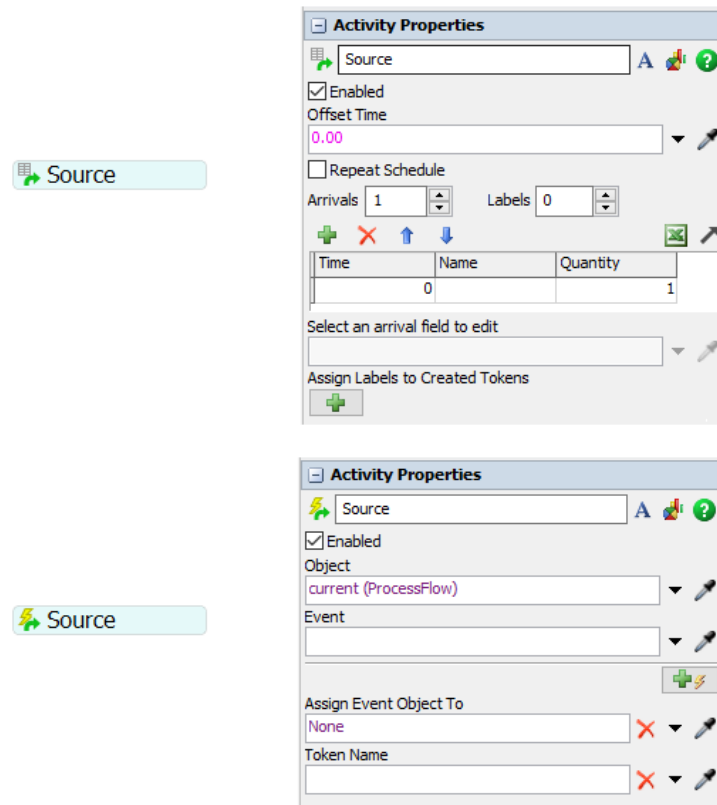


Figura 6.11: *Sources*, a) Atividades, b) Propriedades *default*

6.7 Classificação das Bagagens

Para as bagagens foram criados diferentes *tokens*: *Token.BagsIn* são referentes às bagagens de chegada, ou seja, as bagagens que chegam a este aeroporto. Enquanto que o *token.BagsOut* é referente às bagagens de partida, ou seja, as bagagens que partem deste aeroporto para outro.

Há que considerar outras referências para as bagagens prioritárias e as bagagens em transferência. O primeiro caso refere-se às bagagens de 1ª classe que serão transportadas no início do descarregamento da aeronave (ou avião). O segundo caso (não considerado neste estudo) refere-se às bagagens que são transferidas diretamente entre aeronaves. Em voos de escala, um avião chega e os PAX seguem de novo para outro avião, logo as bagagens seguem o mesmo percurso, saem de um avião que chega ao aeroporto para, logo de seguida, entrarem num avião que realizará uma partida.

6.8 Arquitetura do projeto

Antes de se entrar no desenvolvimento do *process flow* é apresentado um diagrama de blocos, onde é possível verificar o funcionamento do projeto (ver Figura 6.12). Neste é possível verificar o fluxo de bagagens, tanto das bagagens de chegada como das bagagens de partida.

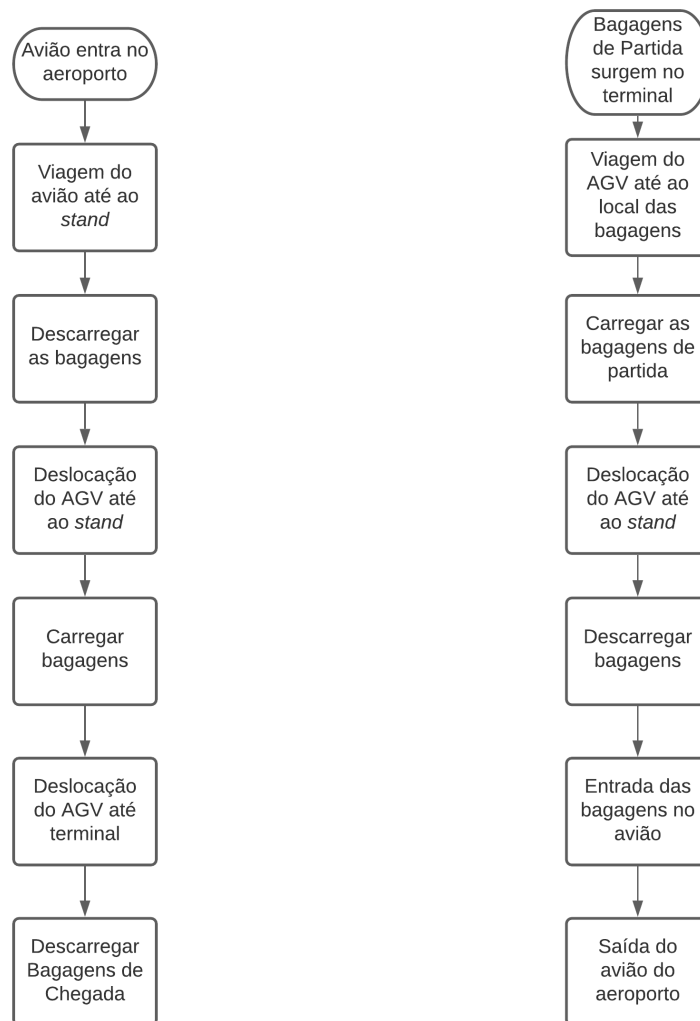


Figura 6.12: Diagrama de blocos, a) Bagagens de chegada, b) Bagagens de partida

Para um voo de chegada, o AGV apenas inicia o seu movimento, até ao local onde o avião se encontra, quando o avião já está parado. De seguida, ocorre a descarga das bagagens de chegada e, caso já haja bagagens no terminal, é feito o carregamento das bagagens de partida.

6.9 Process Flow

Para o funcionamento de todo o mecanismo deste projeto foi necessário criar um *process flow*, fluxograma específico do programa Flexsim. Este é dividido em três grandes *containers* - divisões para melhorar o aspecto e organização do *process flow* criado.

O primeiro *container* é destinado aos recursos do aeroporto, a criação de Bagagens (tanto as deste aeroporto como as de um aeroporto de fora) e o envio destas bagagens para listas globais do sistema. O segundo *container* é referente à implementação do Fluxo de Aviões e o terceiro *container* à implementação do Fluxo de AGV.

Para facilitar a compreensão do trabalho realizado no Flexsim, foram feitos os fluxogramas dos diferentes *containers*. Estes serão apresentados antes da introdução de cada um dos fluxos.

O fluxograma referente ao *container* das bagagens (ver Figura 6.13) é dividido em duas partes, visto que existem as bagagens de chegada e as bagagens de partida.

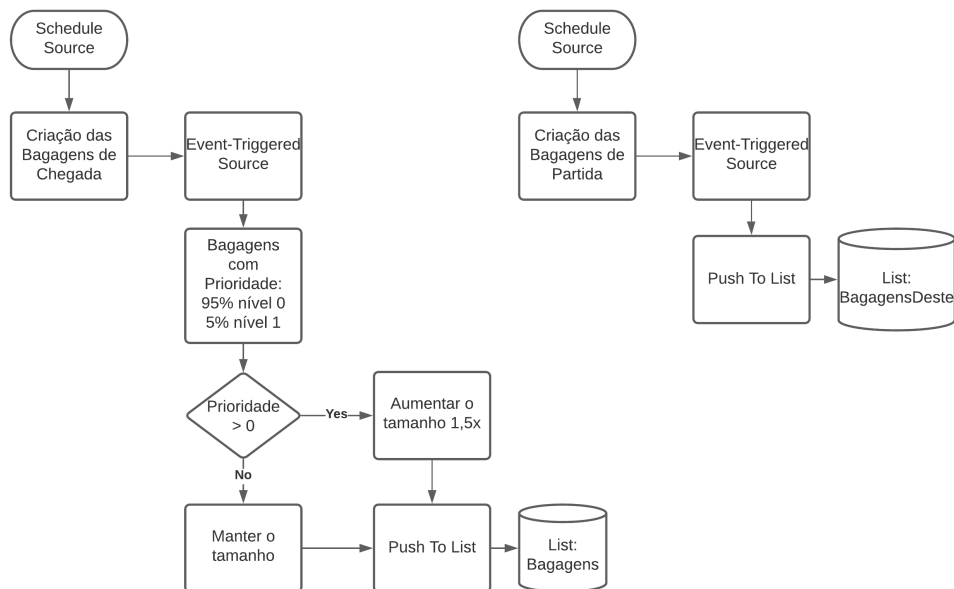


Figura 6.13: Fluxograma realizado para as bagagens, a) Bagagens de chegada, b) Bagagens de partida

6.9.1 Container das Bagagens

O fluxo de bagagens foi utilizado para sincronizar os restantes fluxos, o fluxo de Aviões e o fluxo de AGV. Tal como foi mencionado anteriormente, as bagagens foram criadas como um *“Flow Item Bin”*, logo foram criadas duas listas globais de forma a se poderem usar ao longo de todo o *processflow*. Uma para a saída das bagagens, *“PartidaDeBagagens”*, e a segunda para a chegada das mesmas, *“ChegadaDeBagagens”*. Apesar de terem tratamentos diferentes, ambas as listas têm as mesmas propriedades (ver Figura 6.14).

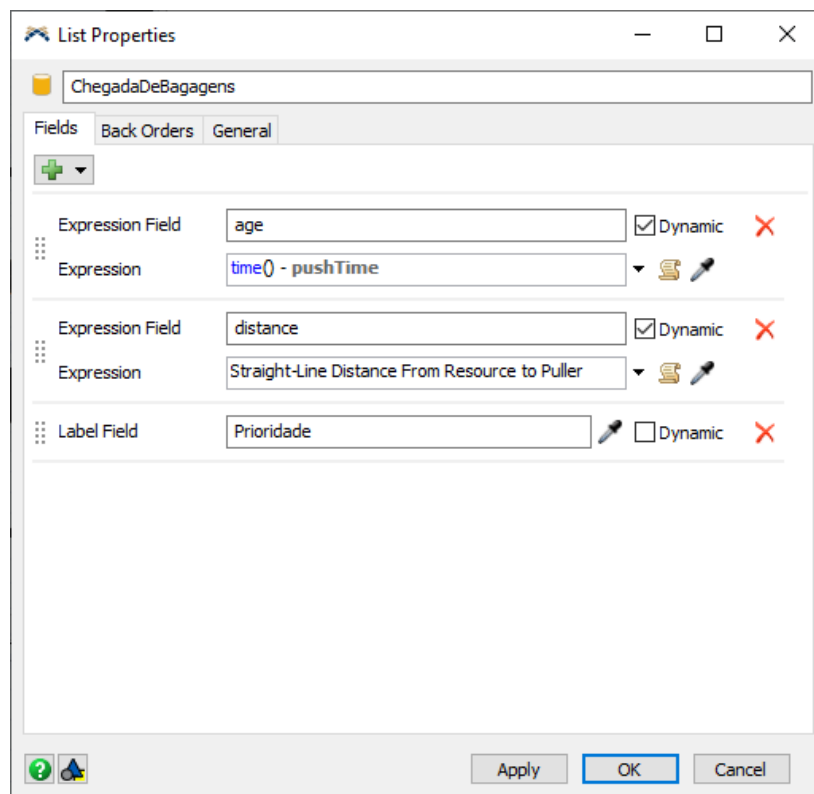


Figura 6.14: Propriedades das listas globais

As duas primeiras propriedades (*age* e *distance*) são automaticamente criadas pelo *software*, porém a terceira foi criada para tratar bagagens com prioridade.

6.9.1.1 Propriedades das Schedule Source

Para as bagagens surgirem no instante (e local) pretendido foi necessário criá-las através de uma *schedule source*.

As bagagens de chegada (bagagens de outro aeroporto) são geradas na *queue BagagensDeOutroAero* e foi definido como *token.BagagensFora* - esta informação

encontra-se no retângulo laranja (ver Figura 6.15). Enquanto que a informação dentro do retângulo vermelho diz respeito ao instante de tempo e à quantidade de bagagens que irão surgir na respetiva *queue*. Foi, também, necessário introduzir um novo campo, o destino das bagagens - este será associado ao *token.DestinoFora* de forma a ser utilizado pelos aviões quando tiverem de decidir para que manga ir (retângulo verde).

Bagagens Aeroporto de Fora

Time	Name	Quantity	DestinoFora
50		78	1
600		76	2
1200		75	3
1800		68	4
2400		82	5
15000		71	6

Activity Properties

Source

Enabled

Offset Time: 0.00

Repeat Schedule:

Arrivals: 43 Labels: 1

Create Object

Object: Bagagens

Quantity: 1

Create In: Create At:

BagagensDeOutroAero

Assign To: Insert at Front of:

token.BagagensFora

Object Flow: None

Assign Labels to Created Objects

Name: DestinoFora

Value: token.DestinoFora

Sink

Deallocate Shared Assets:

Token Data to Preserve:

Trace Histories:

Figura 6.15: *Schedule source* para as bagagens de chegada, a) Atividades, b) Propriedades

No Anexo A é possível verificar toda a tabela inserida nas *schedule source*, tanto das bagagens de chegada como das bagagens de partida.

Para as bagagens surgirem no terminal do Aeroporto Francisco Sá Carneiro, foi feita uma sequência semelhante à anterior alterando apenas: os instantes de tempo, as quantidades, o local onde as bagagens são criadas, a respetiva *label* associada às mesmas e os destinos (ver Figura 6.16). Neste caso as bagagens irão surgir na *queue BagagensDePartida* e terão como *label*, *token.Bagagens*. Estes *tokens* irão apenas ser usados para as *event-triggered source*, excepto o *token.Destino* que apenas será utilizado pelos AGV (a criação do mesmo encontra-se dentro do retângulo verde).

Bagagens Deste Aeroporto

Time	Name	Quantity	Destino
17100		74	1
18900		76	2
20100		68	3
21600		83	4
22800		80	5
24200		71	6
25200		72	7

Activity Properties

Source

Enabled

Offset Time: 0.00

Repeat Schedule:

Arrivals: 43 Labels: 1

Create Object

Object: Bagagens

Quantity: 1

Create In Create At

BagagensDePartida

Assign To Insert at Front of

token.Bagagens

Object Flow: None

Assign Labels to Created Objects

Name: Destino

Value: token.Destino

Sink

Deallocate Shared Assets:

Token Data to Preserve

Trace Histories

Figura 6.16: *Schedule source* para as bagagens de partida, a) Atividades, b) Propriedades

6.9.1.2 Propriedades das Event-Triggered Source

Depois de terem sido criados os objetos através das *schedule source*, foi necessário utilizar uma *event-triggered source* para cada um dos grupos de bagagens (chegadas e partidas).

Começando pelas bagagens de partida foi utilizada uma *event-triggered source* seguido de um *Push To List* (ver Figura 6.17). Nesta figura é possível verificar que esta *source* apenas é ativada quando as bagagens dão entrada na *queue BagagensDePartida*. O *token* mencionado na subsecção anterior é utilizado no campo *Entering Item* (retângulo laranja), e no valor a ser puxado para a lista *BagagensDeste* (retângulo vermelho).

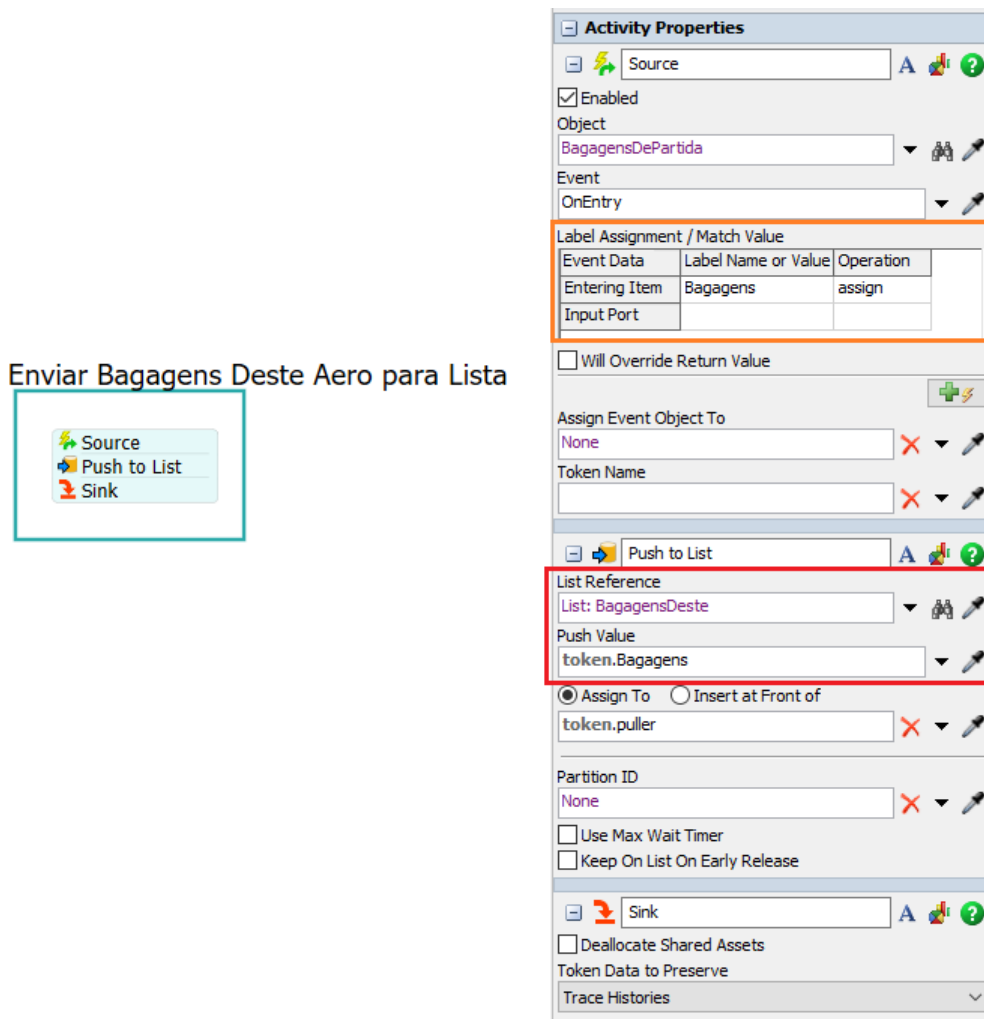


Figura 6.17: *Event-triggered source* para as bagagens de partida, a) Atividades, b) Propriedades

Para as bagagens que chegam ao aeroporto, foi necessário utilizar um método para se conseguir puxar, primeiramente, as bagagens prioritárias, ou seja, as bagagens de PAX de 1^a classe.

Foi utilizado, de forma semelhante, uma *event-triggered source* seguido de um *Push To List* (ver Figura 6.18) com as devidas modificações assinaladas nos retângulos. No retângulo laranja é possível verificar que o *token* é diferente, sendo neste caso o mencionado para as bagagens de chegada, *token.BagagensFora*. O mesmo acontece no campo para a lista (retângulo vermelho), onde é possível verificar que a lista para onde são enviadas as bagagens é diferente da anterior, assim como o valor a ser puxado.

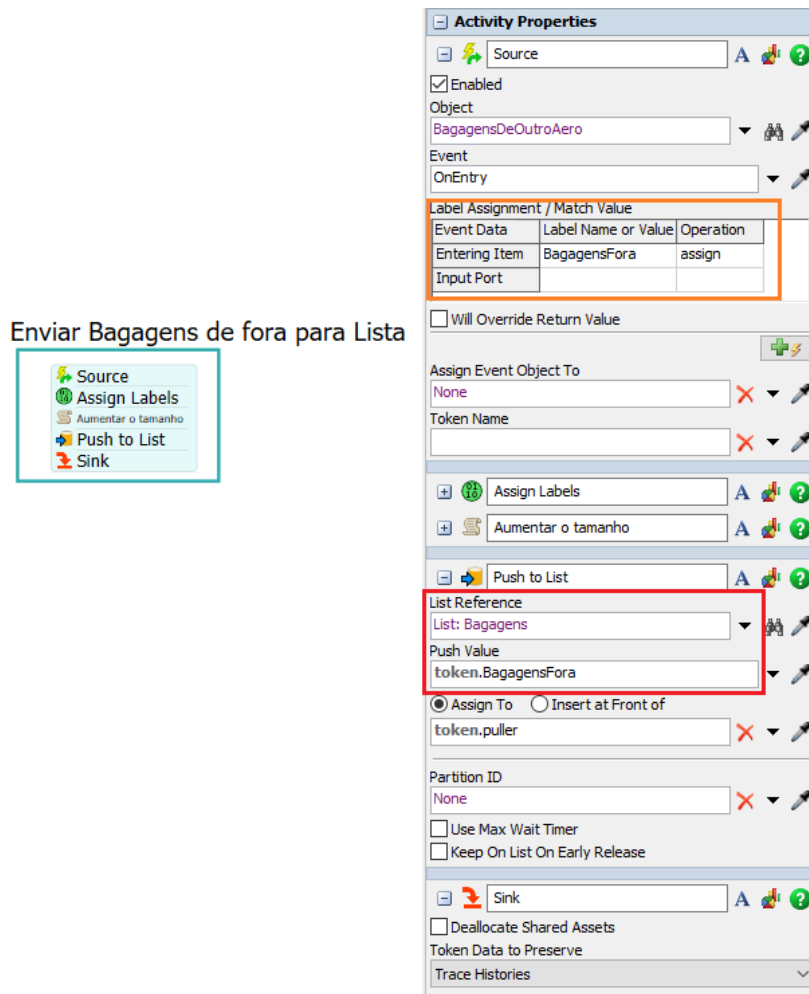


Figura 6.18: *Event-triggered source* para as bagagens de chegada, a) Atividades, b) Propriedades

Na Figura 6.18 é possível verificar um *Assign Labels* e uma atividade com código que tem o nome de “Aumentar o tamanho” (ver Figura 6.19). Ambas as atividades trabalham em conjunto para que as bagagens prioritárias se destaquem no meio das restantes. Na atividade de *Assign Labels* é criada uma nova *label* ao *token.BagagensFora*, ou seja, às bagagens de partida. Esta nova *label* tem percentagens específicas: 5% das bagagens será de PAX de 1^a classe, logo terá prioridade com nível 1, e as restantes 95% das bagagens terão prioridade com nível 0. No código, a parte que foi acrescentada, visto que o Flexsim automaticamente cria as quatro primeiras linhas, foi o comando *if* que verifica se as bagagens têm nível prioridade igual a 1. Caso tenham, o seu tamanho é aumentado em 1,5x.

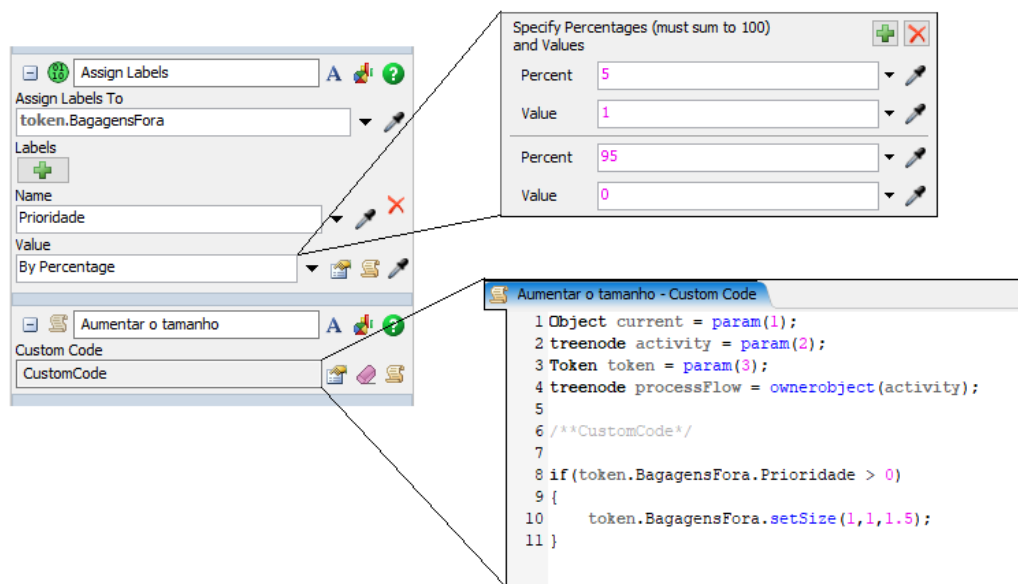


Figura 6.19: Propriedades do *Assign Labels* e Código

6.9.1.3 Recursos do Aeroporto

Por fim, surgem os recursos do aeroporto que dizem respeito às listas criadas para o tratamento das bagagens.

A diferença entre uma lista local e uma lista global é que a lista local apenas pode ser utilizada na mesma sequência de atividades enquanto que a lista global pode ser utilizada em qualquer parte da simulação.

Estas listas são importantíssimas para o funcionamento da simulação, pois são a ponte entre o fluxo de aviões e o fluxo de AGV. Para esta simulação foi necessária a utilização de duas listas (uma lista local e uma lista global) para cada conjunto de recursos (aviões e AGV – ver Figura 6.20). Os recursos do “aeroporto de fora” dizem respeito ao fluxo de aviões enquanto que os recursos “deste aeroporto” dizem respeito ao fluxo de AGV.

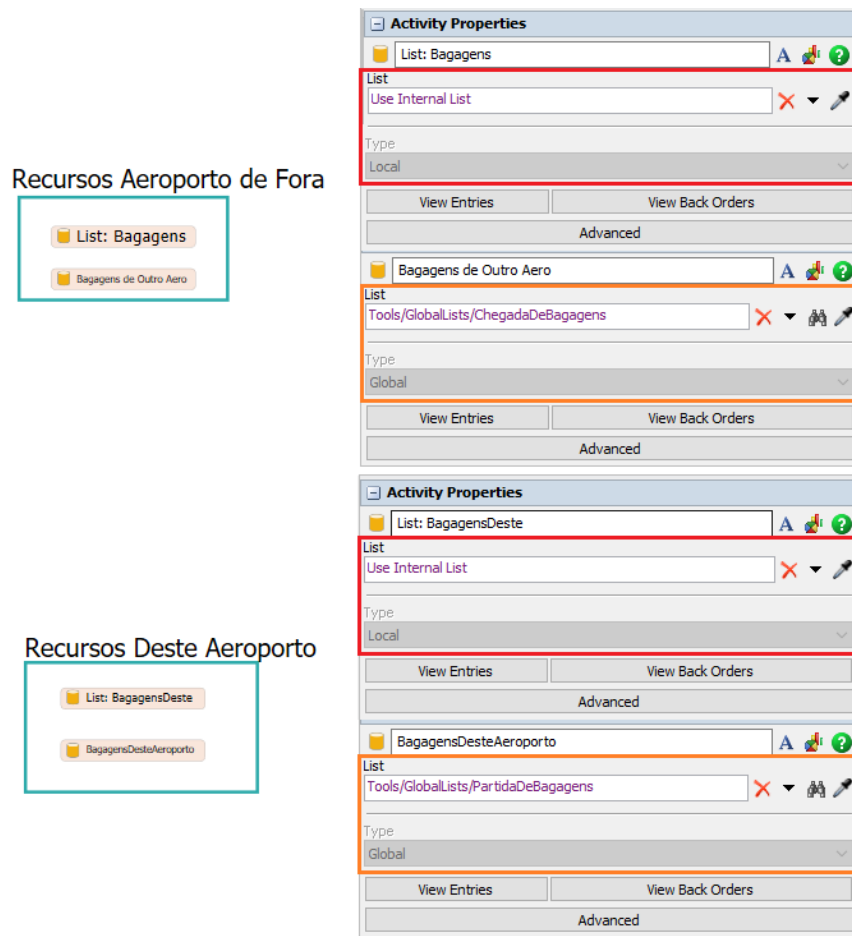


Figura 6.20: Recursos do aeroporto, a) Atividades, b) Propriedades

As duas listas locais (“List: Bagagens” e “List: BagagensDeste”), (retângulo vermelho), servem para puxar as bagagens criadas dentro das *event-triggered sources* e são enviadas para os fluxos de aviões e AGV, uma para cada fluxo. Enquanto que as listas globais (“Bagagens de Outro Aero” e “BagagensDesteAeroporto”), (retângulo laranja), são as que realizam a sincronização entre os fluxos. No caso da lista “BagagensDesteAeroporto”, as bagagens são puxadas dentro do *container* dos AGV e são enviadas para o *container* dos aviões, enquanto que na outra lista acontece exatamente o contrário, do *container* dos aviões para o *container* dos AGV.

Há, ainda, uma lista local (“Aviões no terminal”) criada exclusivamente para os aviões que se encontram parados no terminal à espera de bagagens (ver Figura 6.21). Através da propriedade *distance* (retângulo vermelho), é avaliada a distância de cada um ao *stand* que se encontra com as bagagens e, futuramente, o avião que se encontrar mais perto do local é o que irá realizar o voo de partida.

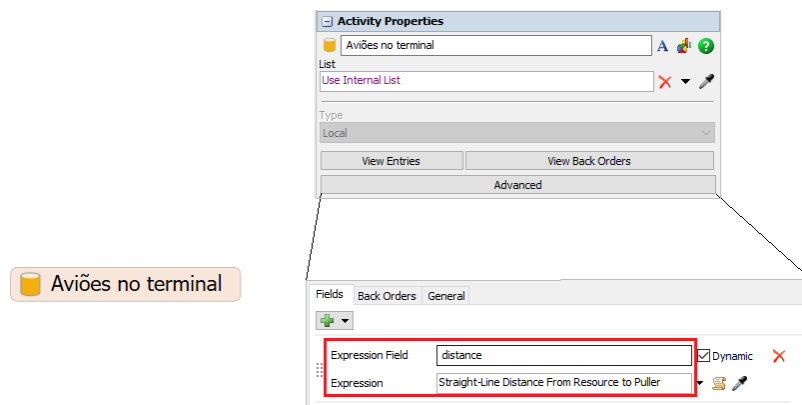


Figura 6.21: Lista dos aviões, a) Atividades, b) Propriedades

6.9.2 Fluxograma do Fluxo de Aviões

O fluxograma referente ao *container* do fluxo de aviões (ver Figura 6.22) é dividido em duas partes: lado esquerdo, o processo desde a chegada ao aeroporto até ao descarregamento de bagagens num dos *stands* e, do lado direito, o processo desde o carregamento das bagagens de partida até à saída do aeroporto.

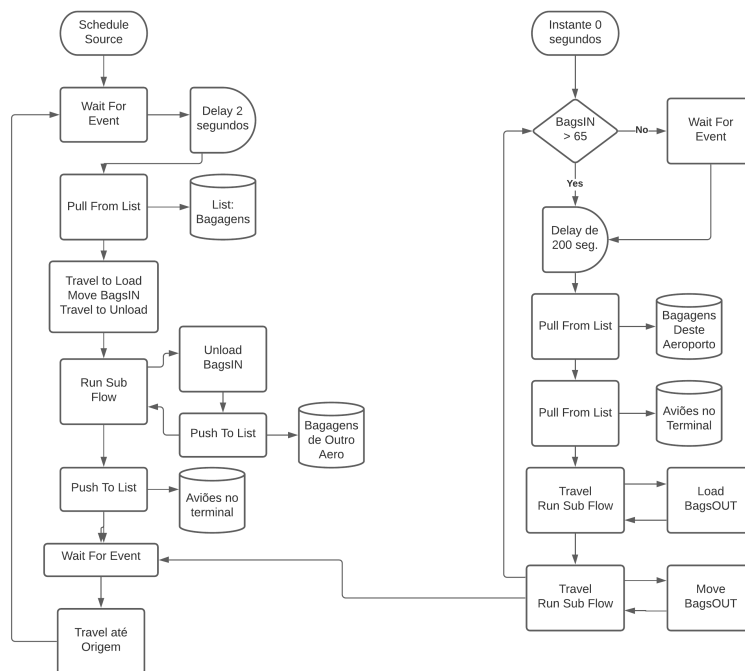


Figura 6.22: Fluxograma realizado para o Fluxo de Aviões

6.9.3 Container do Fluxo de Aviões

Para a implementação do circuito e, respectivo, fluxo dos aviões dentro do aeroporto foi, primeiramente, necessário utilizar uma *schedule source* com os objetos em formato de aviões (ver Figura 6.23). Nos retângulos vermelhos é possível verificar o comando utilizado para se identificar um objeto do modo de visualização 3D.

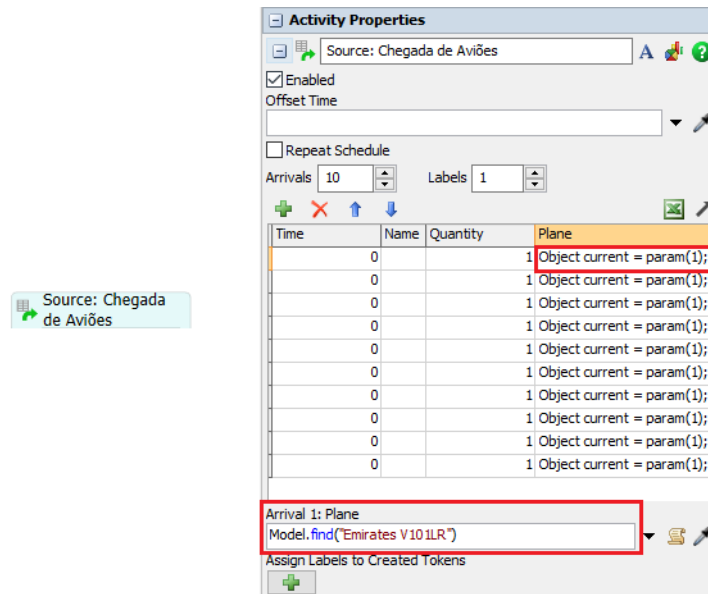


Figura 6.23: *Schedule Source*, a) Atividade, b) Propriedades

6.9.3.1 Pré Chegada ao Aeroporto

Antes do avião chegar ao aeroporto com as devidas bagagens é necessário, na simulação, esperar que as bagagens surjam na devida *queue* (“BagagensDeOutro-Aero”), logo foi utilizado um “Wait For Event” (ver Figura 6.24) que espera que haja entrada de itens dentro dessa *queue* (retângulo vermelho), antes do avião começar o seu percurso. É usado um *acquire* para que apenas um avião, de cada vez, realize o percurso até ao local de *load* das bagagens (retângulo laranja). Aliado a estas duas atividades surge um *delay* de apenas 2 segundos (retângulo verde), para que todas as bagagens estejam prontas a ser carregadas. Caso este não existisse e a passagem fosse instantânea, iria ocorrer um *bug* e o avião não recolhia todas as bagagens presentes na *queue*.

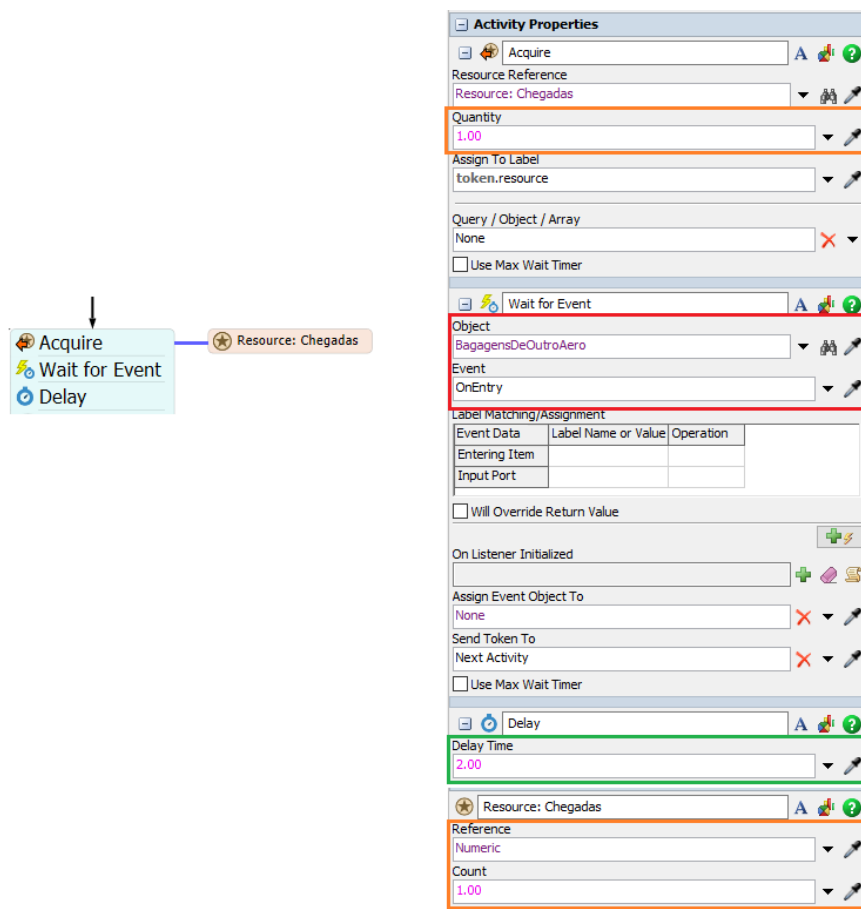


Figura 6.24: *Acquire*, *Wait For Event* e *Delay*, a) Atividades, b) Propriedades

6.9.3.2 Carregamento das Bagagens de Chegada

Após a ocorrência da chegada de bagagens à respectiva *queue*, é necessário mover um avião até ao local para realizar este processo de carregamento das bagagens (ver Figura 6.25).

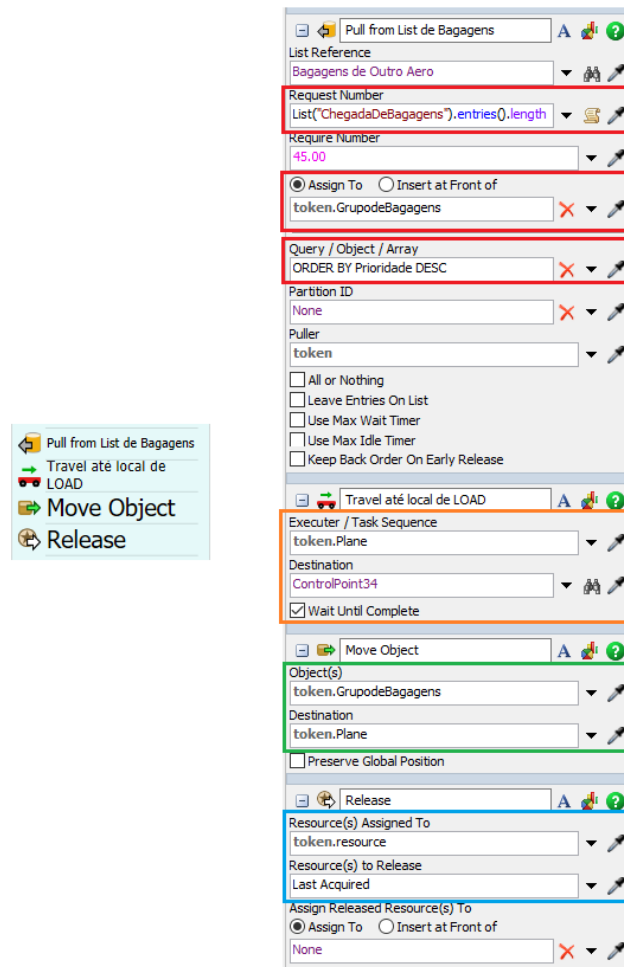


Figura 6.25: *Pull From List, Travel, Move Object e Release*, a) Atividades, b) Propriedades

Para tal, recorreu-se à lista local (definida para as bagagens de outro aeroporto) e puxou-se as bagagens enviadas através da *event-triggered source*. Esta atividade é realizada através de um “Pull from List”; foi preciso configurar de forma a que todas as bagagens que se encontrassem na lista fossem puxadas, através do comando “`List("ChegadaDeBagagens").entries().length`”. Ou seja: nome da lista, coluna pretendida e tamanho dessa coluna; e ordená-las de forma prioritária, pois na *queue* estão todas espalhadas. Para este conjunto de bagagens a ser transportado para o aeroporto criou-se uma nova *label*, o `token.GrupoDeBagagens` (estes dados encontram-se nos retângulos vermelhos).

De seguida, surge uma atividade de *travel*, nesta apenas tem de se indicar o local de destino pretendido (será o ponto de controlo ligado à *queue*) e o que se irá mover, `token.Plane` (retângulo laranja).

É dado o *check* em “*Wait Until Complete*” para que o *token.GrupoDeBagagens* não continue até o avião chegar ao local. Por fim, é dada a indicação para mover as bagagens (da *queue* para o avião) através de um *move* onde é necessário indicar o que mover (*token.GrupoDeBagagens*) e para que local será movido, *token.Plane* (retângulo verde).

Depois de se ter realizado o carregamento das bagagens, é necessário dar *release* ao recurso utilizado no *acquire* mencionado na sub subsecção anterior, de forma a que outro avião esteja preparado quando surgirem mais bagagens (retângulo azul).

6.9.3.3 Deslocação até ao Stand e Descarregamento

Após o carregamento das bagagens no avião, é preciso realizar um conjunto de atividades para que o avião se desloque até ao *stand* e realize o descarregamento das bagagens (ver Figura 6.26).

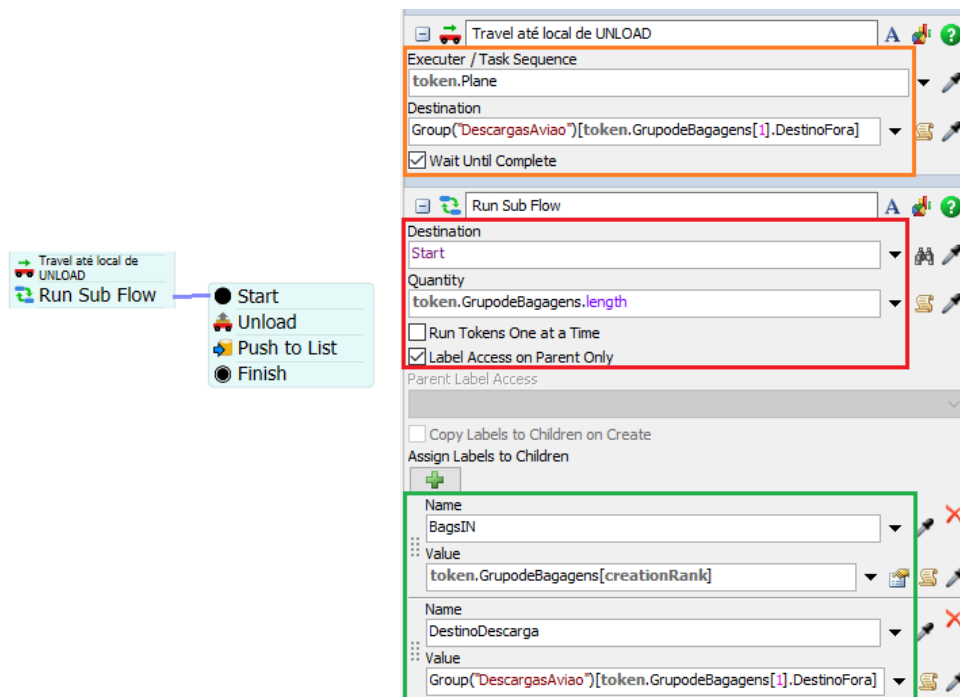


Figura 6.26: *Travel* e *Run Sub Flow*, a) Atividades, b) Propriedades

Para tal foi utilizado o comando *Travel*. Este necessita de dois dados: o que se irá mover e para onde se irá mover, ou seja, o *token.Plane* e o destino será “*Group(“DescargasAviao”)[token.GrupoDeBagagens[1].DestinoFora]*”. O *Group(“DescargasAviao”)* é onde estão indicadas todas as descargas possíveis que o avião pode obter (explicado na Tabela 6.1), o *token.GrupoDeBagagens[1]* é o

conjunto de bagagens de chegada, enquanto que o *.DestinoFora* está associado à última coluna incluída na *schedule source* das bagagens de chegada, onde se determinou para que *stand* seguiria cada avião.

De seguida surge um *Run Sub Flow*. Tal como indicado anteriormente, esta atividade é utilizada para que o *token* pai (*token.GrupoDeBagagens*) permaneça nesta atividade até que todos os seus *tokens* filhos (*token.BagsIN*) tenham concluído o *Sub Flow*, retângulo verde. O *token.BagsIN* é referente às bagagens de chegada enquanto que o *token.DestinoDescarga* é referente ao destino do avião.

Este *Run Sub Flow* irá realizar o descarregamento (*Unload*) das bagagens e o respetivo envio dessas bagagens para uma lista (ver Figura 6.27).

Na tarefa de *Unload* é necessário indicar: onde descarregar, o objeto e para onde descarregar. Logo, pode-se verificar que quem contém as bagagens é o *token.Plane* e o objeto é o *token.BagsIn*, *token* introduzido na configuração do *Run Sub Flow* na Figura 6.27. Por fim o local onde descarregar que é o *token.DestinoDescarga*, um dos *token* filhos criados (retângulo verde), o comando *.centerObjects[1]* diz respeito ao tipo de ligação que existe entre a *queue* que estará ligada ao ponto de controlo do *stand*.

Após o *Unload*, é necessário enviar estas bagagens para a lista global “List: Bagagens” de forma a se utilizar, futuramente, no fluxo de AGV. Para tal é necessário utilizar o *token.BagsIN* (referente às bagagens descarregadas) e indicá-lo no “*Push Value*” (retângulo laranja). É utilizado o “*Use Max Wait Timer*” com 0 segundos para que este envio seja instantâneo, caso uma (ou mais) não o seja é dado *release* e por isso é, também, utilizado o “*Keep On List On Early Release*”, este serve para que, mesmo que as bagagens tenham levado *release*, fiquem na lista até serem puxadas por um “*Pull From List*”.

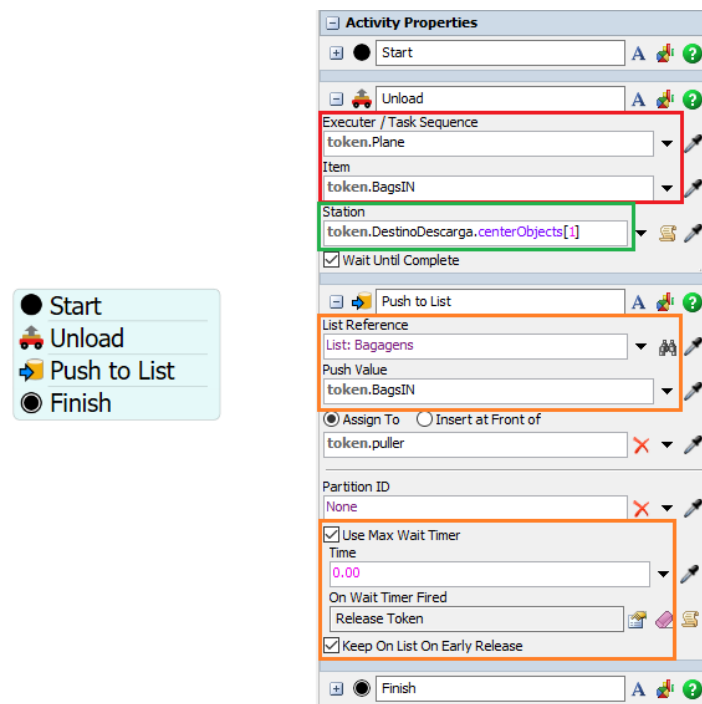


Figura 6.27: *Unload* e *Push To List*, a) Atividades, b) Propriedades

6.9.3.4 Aviões no Terminal

Tal como mencionado anteriormente, os aviões que se encontram no terminal são enviados para uma lista local (ver Figura 6.28). Após se realizar o *unload* das bagagens de chegada, os aviões são enviados para a lista “Aviões no terminal” (retângulo vermelho). Aliado a este envio existe um *Wait For Event*, que irá aguardar, até que o avião volte a receber bagagens para realizar o voo de partida (retângulo laranja). Quando o mesmo recebe as bagagens, irá realizar um *travel* até ao local de origem (onde começou) (retângulo verde).

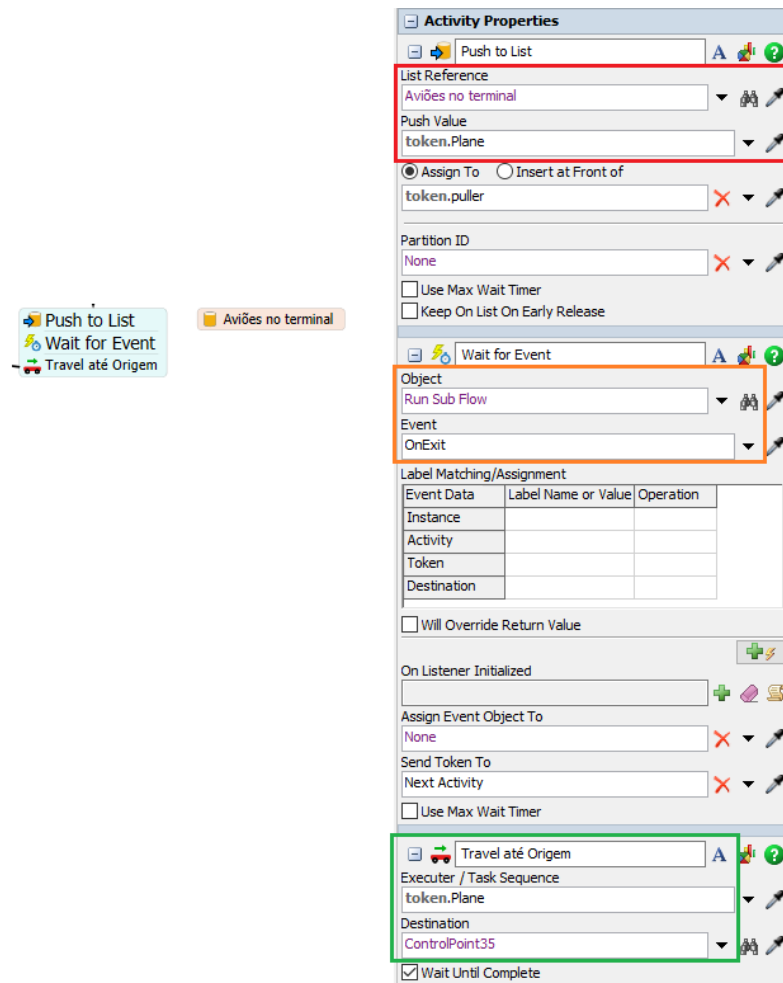


Figura 6.28: *Push To List*, *Wait For Event* e *Travel*, a) Atividades, b) Propriedades

6.9.3.5 Espera Pelas Bagagens de Partida

Depois dos aviões que estão no terminal estarem numa lista, é necessário esperar que existam bagagens de partida para seguir o voo (ver Figura 6.29). Sendo assim, é utilizado um *decide* para saber se já há bagagens em alguma *queue*. Este *decide* foi desenvolvido através de um código que inclui um ciclo *for*. Este irá passar por todas as *queues* inseridas no grupo “PartidasBags” e verificar se há mais de 65 bagagens em algumas das *queue* (retângulo vermelho). Caso tenha irá retornar o valor 1, caso não tenha irá retornar o valor 2. Estes valores são, respetivamente, a saída 1 (continuar para as restantes atividades) ou a saída 2 (entrar na atividade *Wait For Event*).

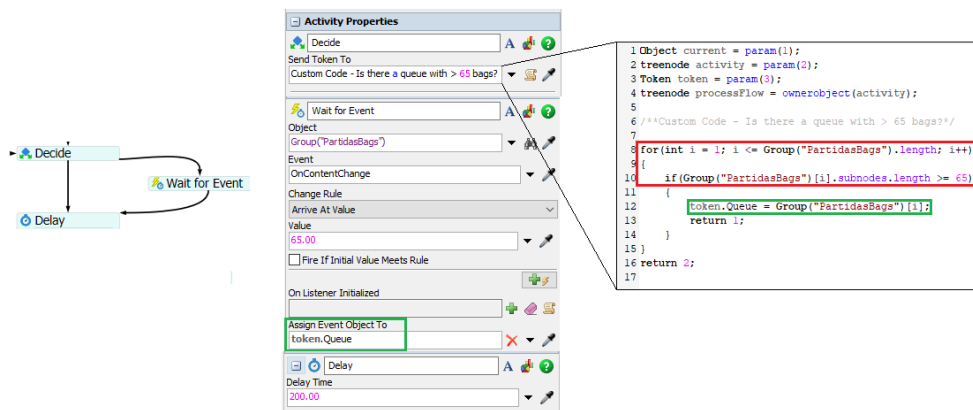


Figura 6.29: *Decide*, *Wait For Event* e *Delay*, a) Atividades, b) Propriedades

Caso entre no *Wait For Event*, irá esperar que haja 65 bagagens para continuar. Se alguma *queue* tiver mais de 65 bagagens, esta será associada ao *token.queue*, seja pela atividade *decide* ou pela atividade *wait for event* (retângulo verde).

Depois, é feito um *delay*, de 200 segundos. Como ambos os eventos (*decide* e *wait*) estão com condições para 65 bagagens, há a probabilidade de um dos AGV ainda estar a transportar o resto das bagagens para a *queue*. Visto que, tal como verificado no capítulo anterior há, em média, 79 bagagens por voo. Na simulação há voos em que, a quantidade de bagagens, chega às 84 unidades.

6.9.3.6 Aquisição das Bagagens Através de Listas

De seguida, surgem dois “*Pull From List*”, um correspondente aos aviões que estão no terminal e o outro referente às bagagens que se encontram na *queue* (ver Figura 6.30).

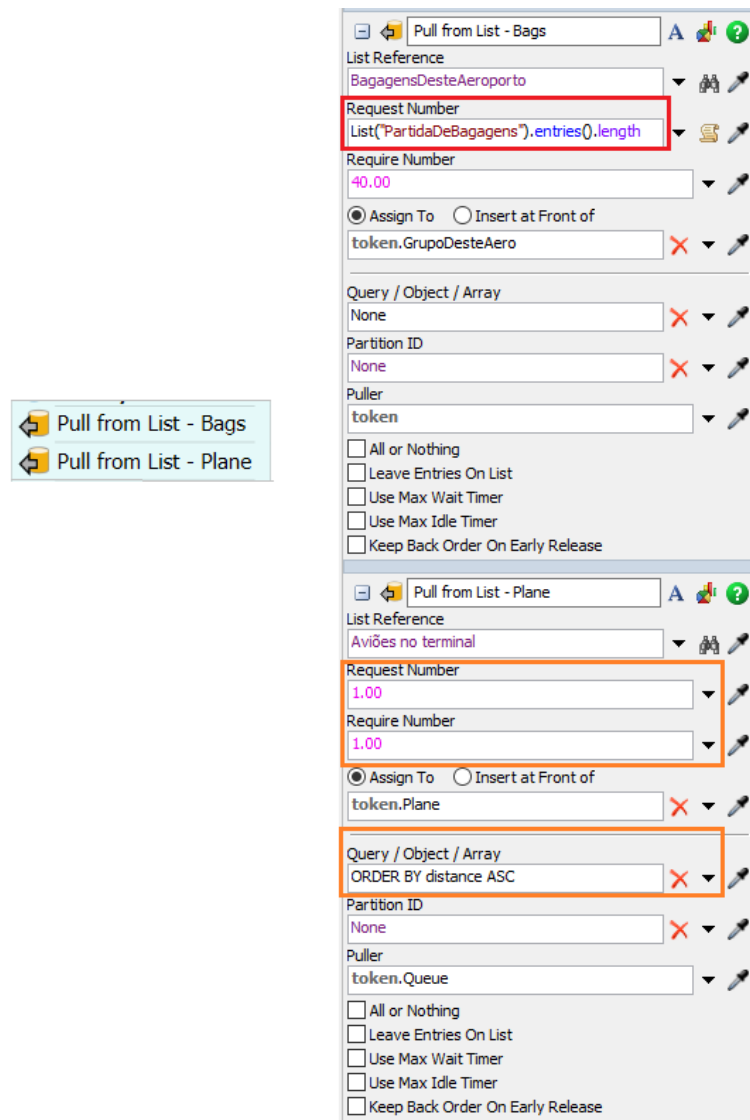


Figura 6.30: *Pull From Lists*, a) Atividades, b) Propriedades

O primeiro “*Pull From List*”, referente às bagagens, à semelhança do primeiro que surgiu, utiliza o comando `.entries().length` de forma a receber todos os itens que deram entrada naquele instante. Apesar do comando ser o mesmo, os itens estão na segunda lista global, `List("PartidaDeBagagens")` (retângulo vermelho). O conjunto destas bagagens fica associada ao `token.GrupoDesteAero`.

O segundo “*Pull From List*”, referente aos aviões no terminal, terá (tanto no *request number* como no *require number*) o valor 1, de forma a que apenas 1 avião da lista seja utilizado. Este avião será o que estiver mais próximo da *queue* com as bagagens visto que, como já se tem conhecimento de todas as posições dos

aviões que se encontram na lista, é apenas necessário verificar o que tem a menor distância à *queue* através do comando *ORDER BY distance ASC*, ou seja, para ordenar as distâncias por ordem ascendente (retângulo laranja).

6.9.3.7 Carregamento das Bagagens de Partida

De seguida, inicia-se o momento em que o avião começa a receber as bagagens. Para tal foram utilizados dois comandos: um *travel* e um *Run Sub Flow* para realizar o carregamento (ver Figura 6.31).

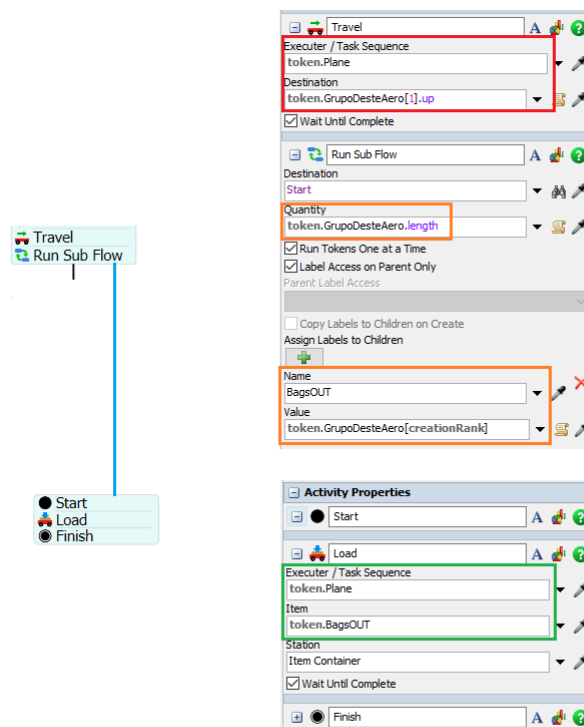


Figura 6.31: *Travel* e *Run Sub Flow*, a) Atividades, b) Propriedades

O comando *travel* surge com uma diferença em relação aos já visualizados. Este apresenta, no local onde se indicar o destino, o comando *.up* no fim do *token.GrupoDesteAero[1]* (retângulo vermelho). O comando *.up* retorna um apontador para o actual local das bagagens. Neste caso será uma das *queues*, entradas para o avião, que estará ligada a um ponto de controlo do percurso dos aviões. Assim, o avião desloca-se até ao ponto de controlo e quando as bagagens forem carregadas retorna uma referência de que o local já se encontra “limpo”, isto é, sem bagagens, e segue para a próxima atividade.

O *Run Sub Flow* terá como *token* pai o *token.GrupoDesteAero* e como *token* filho o *token.BagsOUT* (retângulo laranja). Dentro deste irá realizar-se o *load* das bagagens para o avião, para tal é necessário indicar o *token.Plane* como o objeto a mover e o *token.BagsOut* como itens a carregar (retângulo verde).

6.9.3.8 Saída do Aeroporto

Para realizar a simulação do voo de partida do avião é utilizado um *travel* até ao local da *sink* “SinkDosAviões” para que se realize a descarga das bagagens, dentro de um *Run Sub Flow* (ver Figura 6.32).

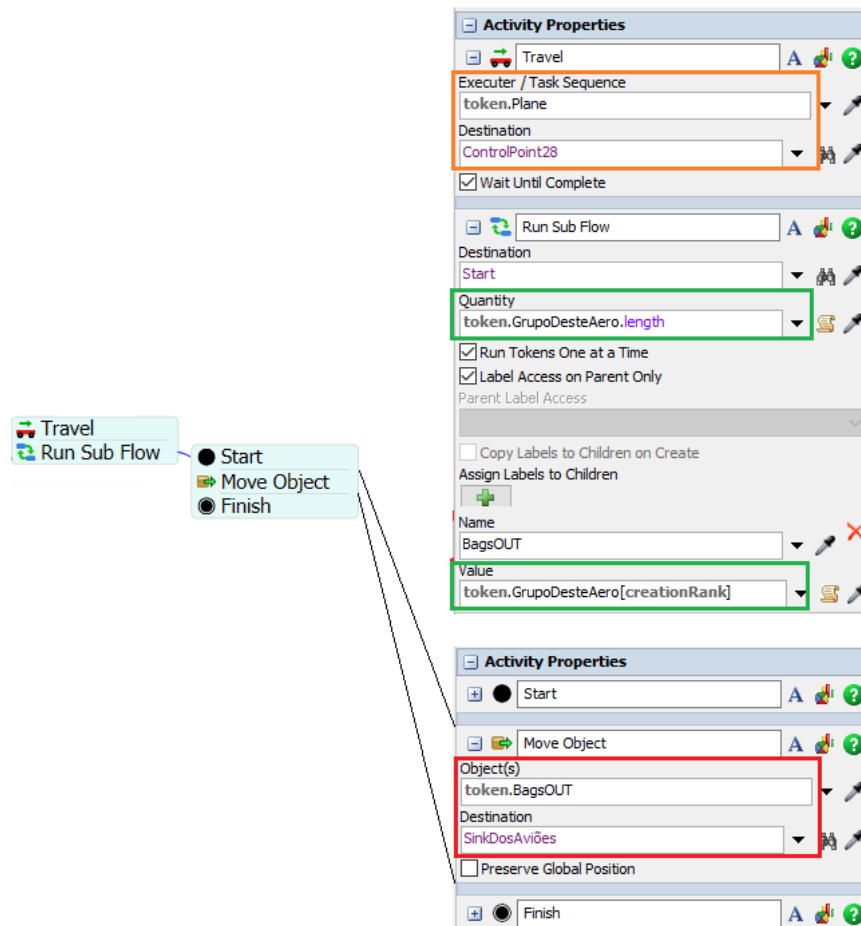


Figura 6.32: *Travel* e *Run Sub Flow*, a) Atividades, b) Propriedades

Dentro da atividade de *travel* é indicado para o avião o ponto de controlo que está ligado à *sink* (retângulo laranja).

Enquanto que no *Run Sub Flow* é possível verificar que é semelhante ao anterior, tem como *token* pai: *token.GrupoDesteAero* e como *token* filho: *to-*

ken.BagsOUT. Neste caso, como já não é relevante para este projeto, a saída do aeroporto é utilizado o comando *move* (retângulo vermelho), em vez de se utilizar o comando *unload*. A diferença é que com o comando *unload* é necessário esperar bagagem por bagagem, e com o comando *move* o descarregamento é instantâneo.

6.9.3.9 Aspeto Final do Container Referente ao Fluxo de Aviões

Depois de explicado, em detalhe, todas as atividades referentes à implementação do fluxo de aviões convém apresentar o aspeto do *container* (ver Figura 6.33) que engloba todas estas atividades.

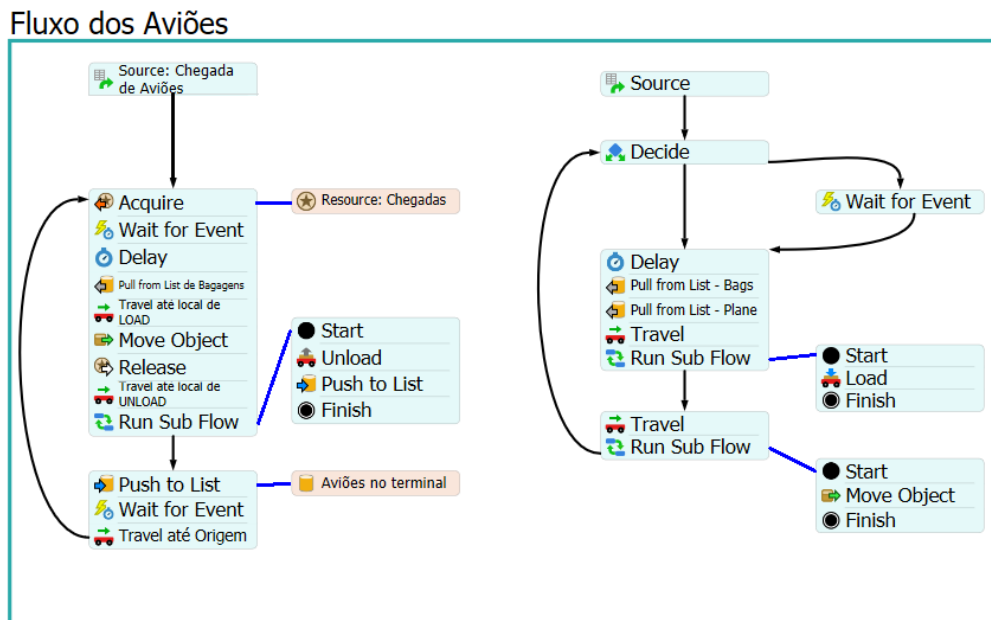


Figura 6.33: Aspeto final do *container* do fluxo de aviões

6.9.4 Fluxograma do Fluxo de AGV

O fluxograma referente ao *container* do fluxo de AGV (ver Figura 6.34) é dividido em duas partes: lado esquerdo, o processo desde a chegada ao aeroporto até ao descarregamento de bagagens num dos *stands* e, do lado direito, o processo desde o carregamento das bagagens de partida até à saída do aeroporto.

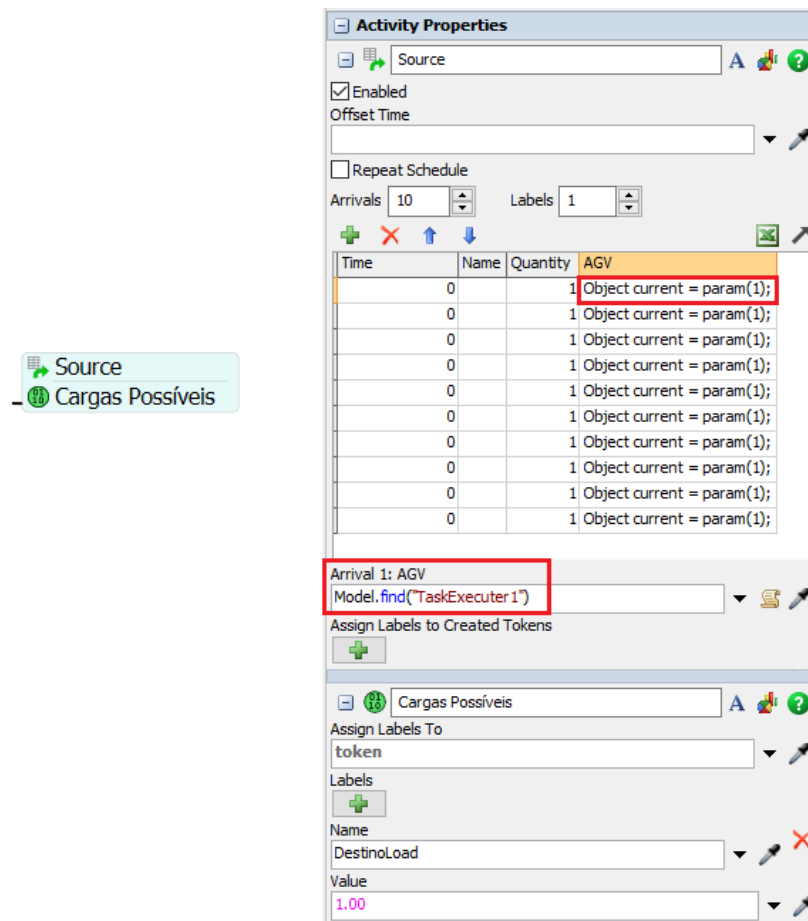


Figura 6.35: *Schedule Source e Assign Labels*, a) Atividades, b) Propriedades

6.9.5.1 Espera por Bagagens

Depois de os AGV estarem preparados, chega a parte onde se terá de esperar pelas bagagens (ver Figura 6.36).

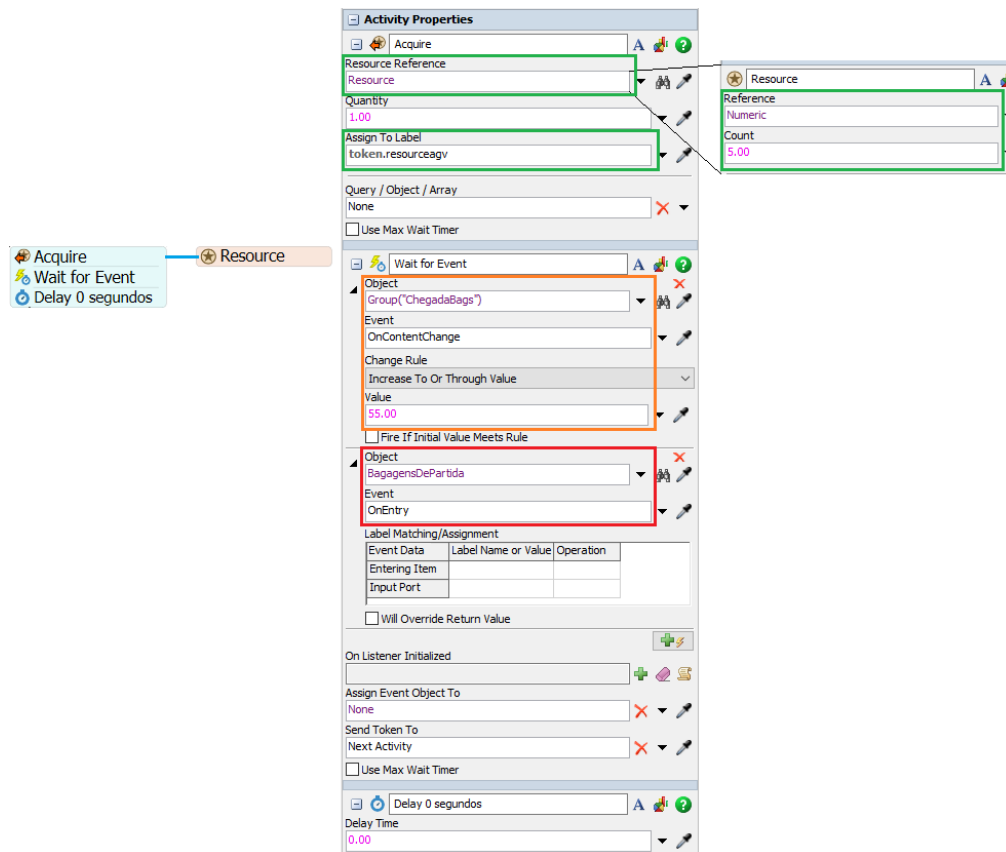


Figura 6.36: *Acquire*, *Wait For Event* e *Delay*, a) Atividades, b) Propriedades

Primeiramente, é utilizado um *acquire* para que apenas um determinado número de AGV (retângulo verde) façam a primeira chegada de bagagens, sejam estas bagagens de partida ou bagagens de chegada. Estes são associados ao *token.resourceagv*. No caso da figura, serão utilizados 5 AGV.

De seguida, surge um duplo “*Wait For Event*” onde se espera por um dos dois acontecimentos possíveis. Para tal é utilizado o Grupo “*ChegadaBags*”, referente às *queues* das bagagens de chegada. É verificado quando é que existem, pelo menos, 55 bagagens numa das *queue* (retângulo laranja). O segundo evento que estará à espera é caso surgem bagagens de partida no terminal do aeroporto, este evento irá ativar quando haja uma entrada de bagagens (retângulo vermelho).

De seguida, é utilizado um *delay* de 0 segundos para corrigir *bugs* de tempo. Sem esta atividade, no caso das bagagens de partida, o *software* passava imediatamente para o *acquire* sem continuar para as restantes atividades, porque naquele instante estariam 0 bagagens. Com este *delay*, que efetivamente não tem tempo, o problema ficou resolvido.

6.9.5.2 Onde Surgiram as Bagagens?

Depois de um dos eventos ter ocorrido, é necessário verificar qual deles foi e no caso de serem bagagens de chegada, verificar em que *queue* aconteceu (ver Figura 6.37).

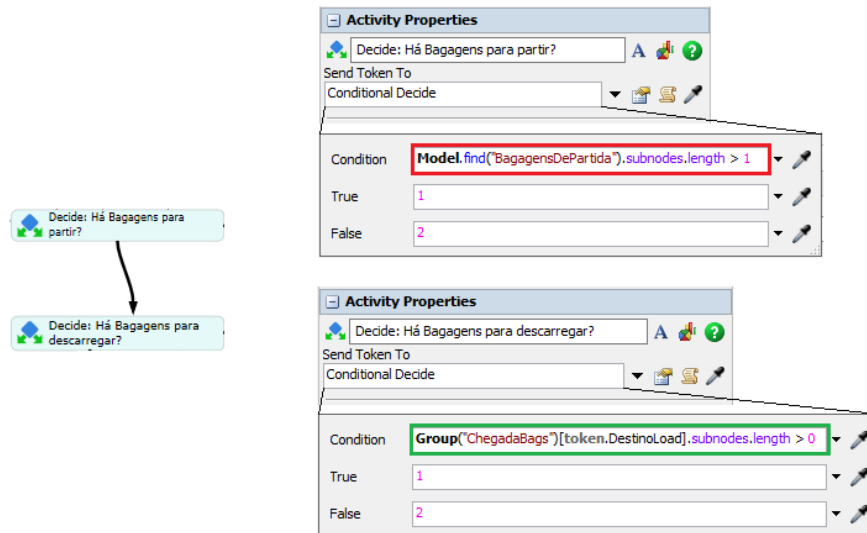


Figura 6.37: *Decide* de partida e *Decide* de chegada e, a) Atividades, b) Propriedades

Para tal são utilizados dois eventos *decide* em separado. O primeiro *decide* faz referência às bagagens de partida (retângulo vermelho). Existe a condição para verificar se na *queue* “BagagensDePartida” existe mais do que 1 bagagem e, caso exista, o AGV irá levar essa bagagem até a um *stand* definido. Caso não exista entra no segundo *decide* onde se verifica (na primeira vez) se no “Group(“ChegadaBags”)[*token.DestinoLoad*]” existe alguma bagagem. Na primeira vez que aqui entra será no primeiro elemento visto que o *token.DestinoLoad* foi iniciado com valor 1.

Caso a afirmação seja falsa, o segundo *decide* entra num *loop* (ver Figura 6.38). Primeiro o *token.DestinoLoad* é incrementado em 1 valor e de seguida corre o próximo elemento do Grupo “ChegadaBags” até não existirem mais elementos neste grupo, condição representada no retângulo azul. Caso não existam bagagens em nenhum deles, o *token.DestinoLoad* é repostado com valor 1 e segue para o *acquire* antes da atividade que inclui dois *Wait For Event*, através da ligação B representada.

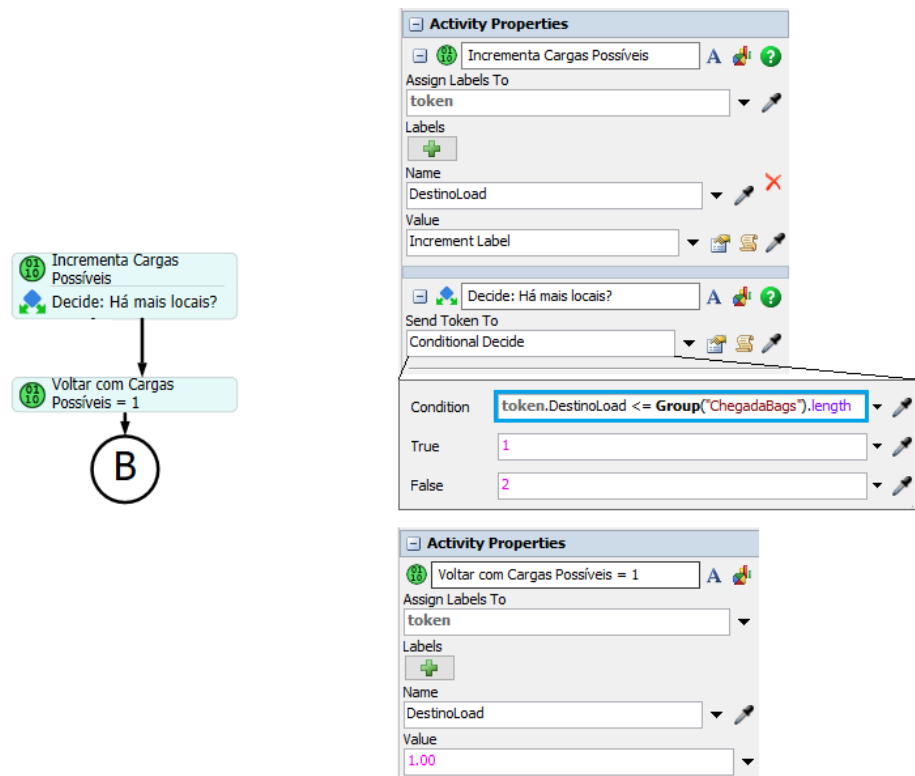


Figura 6.38: *Increment Label, Decide e Assign Label*, a) Atividades, b) Propriedades

6.9.6 Bagagens de Chegada

Após ocorrerem os dois *decides* e o respetivo ciclo para se saber em que *stand* se encontram as bagagens, foi necessário, para as bagagens de chegada, realizar um “*Pull From List*” e recorrer à lista global “*ChegadaDeBagagens*”, para realizar a viagem até ao *stand* (ver Figura 6.39).

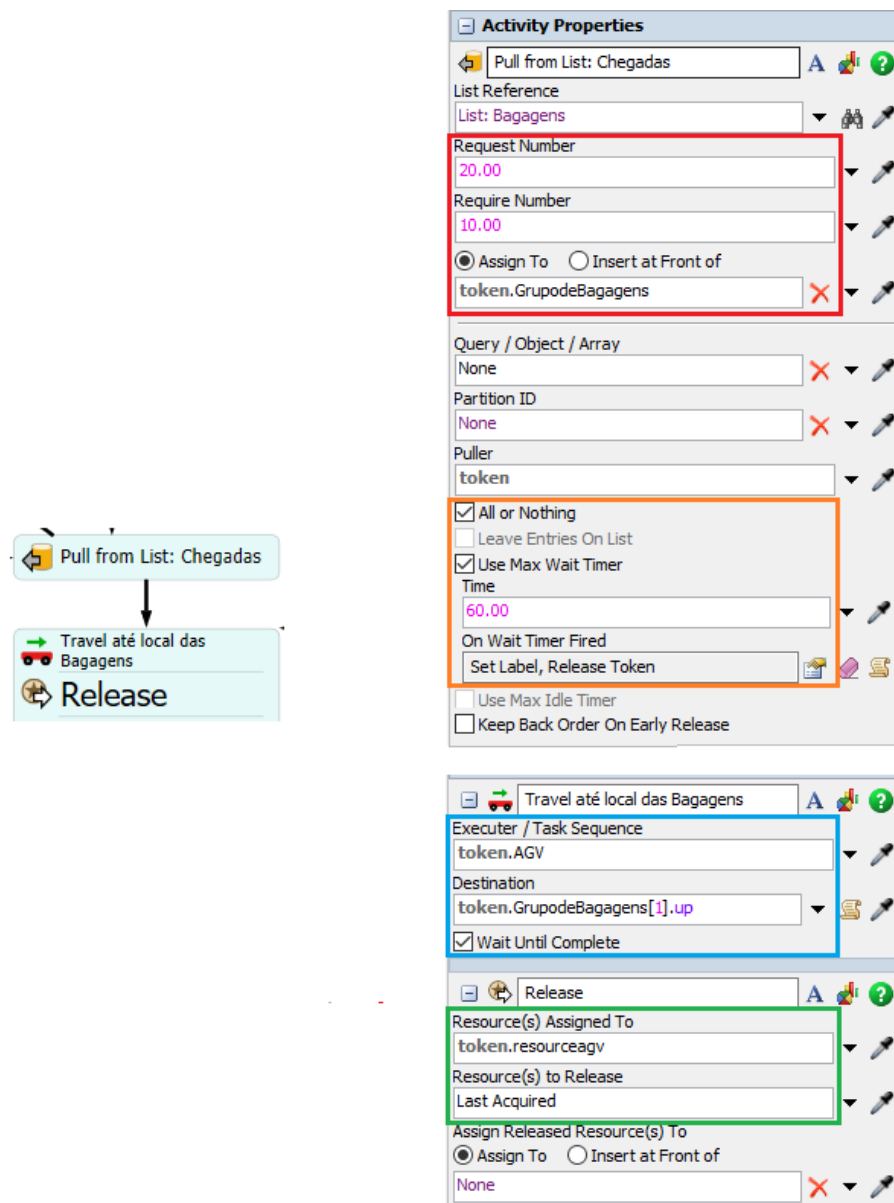


Figura 6.39: *Pull From List, Travel e Release*, a) Atividades, b) Propriedades

Como foi mencionado na subsecção dos AGV, estes apenas irão (no máximo) transportar 20 bagagens. No entanto, é necessário escolher um valor mínimo para os AGV realizarem o percurso. Foi definido um valor mínimo de 10 bagagens (retângulo vermelho). Ainda no “*Pull From List*” foram selecionadas as propriedades “*All or Nothing*” e “*Use Max Wait Timer*”. A opção “*All or Nothing*” faz com que nenhum valor seja retirado da lista até que o *Require Number* (10 bagagens) seja atingido. A opção “*Use Max Wait Timer*” serve para, caso o AGV entre

nesta atividade e não haja, pelo menos, 10 bagagens no *stand*, o AGV irá esperar o tempo imposto nesta opção, 60 segundos (retângulo laranja). A este conjunto de bagagens puxadas da lista foi dado o *token.GrupodeBagagens*.

Na atividade de *travel* é necessário, mais uma vez, indicar o destino e objeto que se irá mover. Para tal, foi necessário escolher o *token.AGV* e como destino foi utilizado, à semelhança do que aconteceu no percurso dos aviões, o comando *.up* no fim do *token.GrupodeBagagens[1]* (retângulo azul). No retângulo verde é possível verificar que após cada AGV chegar ao local do *stand* é dado o *release* do *token.releaseagv* para que o outro conjunto de AGV entre no “*Wait For Event*”.

6.9.6.1 Carregamento e Descarregamento das Bagagens de Chegada

Após a viagem do AGV até ao local do *stand*, é necessário proceder ao carregamento das bagagens para, de seguida, realizar o percurso para o terminal e descarregar as bagagens (ver Figura 6.40).

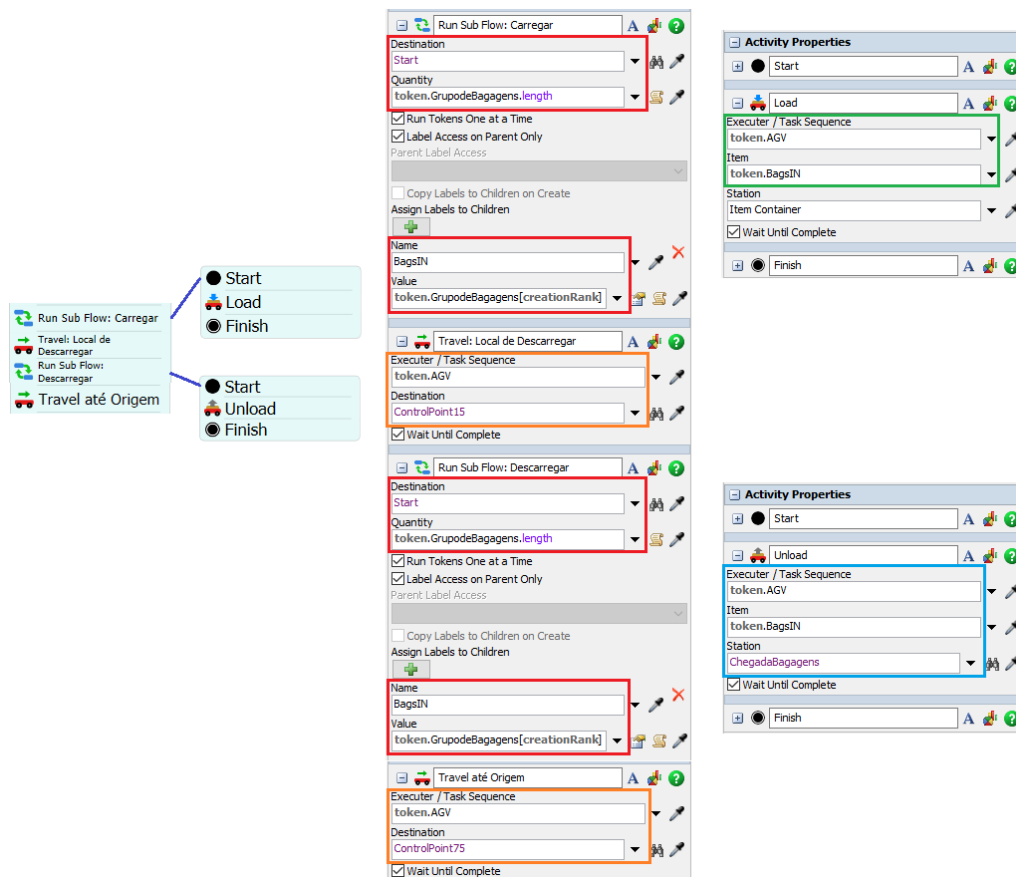


Figura 6.40: *Run Sub Flow*, *Travel*, *Run Sub Flow* e *Travel*, a) Atividades, b) Propriedades

Se forem avaliados, em simultâneo, os dois *run sub flows* (tanto o de carregamento como o de descarregamento), pode-se concluir que ambos são exatamente iguais. O que se irá alterar é o instante em que estes ocorrem e os seus respetivos *sub flows*. A primeira propriedade é o *token.GrupoDeBagagens.length*, para que todas as bagagens que estejam dentro do *token.GrupoDeBagagens* (conjunto das bagagens de chegada) sejam introduzidas dentro dos *sub flows*. Para ambos os *sub flows* é utilizado como *token* filho, o *token.BagsIN*. Esta informação pode ser encontrada dentro de ambos os retângulos vermelhos.

Nos retângulos laranjas, é possível verificar as duas atividades de *travel* existentes na figura. O primeiro *travel* diz respeito à viagem que o AGV percorre entre o *stand* onde se encontra até ao terminal, e o *ControlPoint15* diz respeito à *sink ChegadaBagagens*, mencionada na secção “Mapeamento 3D”. O segundo *travel* diz respeito à viagem entre a *sink* e o local de origem dos AGV, *ControlPoint75*, este encontra-se (no terminal) entre a *sink* e a *queue* referente às bagagens de partida.

Na atividade de *load* (retângulo verde), é possível verificar que foi utilizado o *token.AGV*, para o local onde as bagagens serão carregadas e o item a ser carregado é o *token* filho, *token.BagsIN*. A estação de onde as bagagens serão retiradas para carregar os AGV pode ser qualquer um dos *stands*, logo foi escolhida a opção “Item Container”. Basicamente, esta opção, puxa os itens da *queue* que estiver ligada ao *ControlPoint* do *stand* onde os AGV se encontram.

Na atividade de *unload* (retângulo azul), é possível verificar que, à semelhança da atividade de *load*, tanto o *token.AGV* como o *token.BagsIN* são novamente utilizados. Porém o que altera, em termos de propriedades, é a estação para a qual vai ocorrer esta atividade, *sink ChegadaBagagens*. As atividades de *load* e *unload* têm as mesmas propriedades: o objeto a realizar a atividade, o item a mover e de onde (ou para onde) realizar a mesma.

6.9.6.2 Há Mais Bagagens?

Visto que os AGV, através do *Pull From List* já referido, necessitam de 10 bagagens para realizar o seu percurso, é preciso confirmar se há mais bagagens naquele *stand*.

Para tal recorreu-se a dois *decides* (ver Figura 6.41): o primeiro servirá para criar um *loop* caso ainda haja mais de 10 bagagens após a primeira passagem dos AGV; o segundo servirá para verificar, caso haja menos que 10 bagagens, se ainda não está vazio.

Para estes é necessário utilizar o comando “*Group(“ChegadaBags”)[token. DestinoLoad].subnodes.length*”, onde: o grupo é o referente às *queues* onde podem estar as bagagens de chegada, o *token.DestinoLoad* é o elemento do grupo onde estas se encontram e o comando *.subnodes.length* devolve a quantidade atual

que existe naquela *queue*. Logo ao comparar com “<10” (no primeiro *decide*) e “>0” (no segundo *decide*) está-se a comparar a quantidade atual de bagagens.

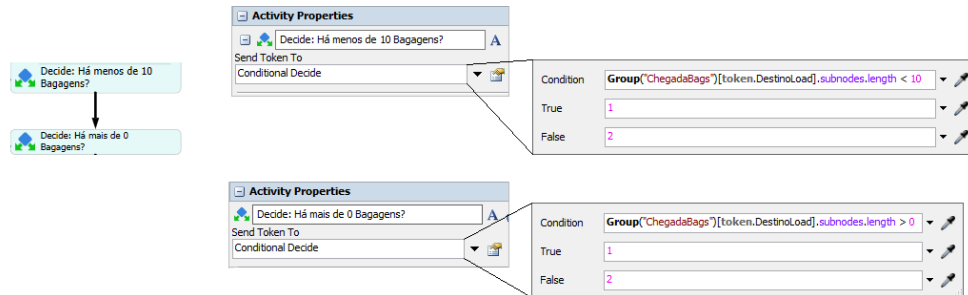


Figura 6.41: *Decides* bagagens de chegada, a) Atividades, b) Propriedades

Caso haja mais que 10 bagagens volta ao primeiro *Pull From List* para que os AGV voltem a realizar o processo de carregar e descarregar com um mínimo de 10 e um máximo de 20. Caso não haja, entra no segundo *decide* onde verifica se ainda existem bagagens; se não existirem retorna ao *decide* referido no início do Fluxo de AGV, o que verifica se existem bagagens na *queue* “BagagensDePartida”. Caso existam entre 0 a 10 bagagens entra no último *Pull From List* desta sequência realizada para as bagagens de chegada.

6.9.6.3 Último Pull From List

Neste último *Pull From List*, após se verificar que existem bagagens na *queue*, porém com um valor inferior a 10, é utilizado apenas 1 AGV para ir buscar o resto (ver Figura 6.42).

Para tal, foi necessário introduzir nos campos *Request Number* e *Require Number* os respetivos valores de 10 e 1 e, de seguida, associar na mesma ao *token.GrupoDeBagagens* (retângulo vermelho). De seguida, à semelhança do que aconteceu anteriormente, é utilizado um “*Max Idle Timer*” de forma a verificar se o AGV não se encontra inativo, neste caso, à 60 segundos.

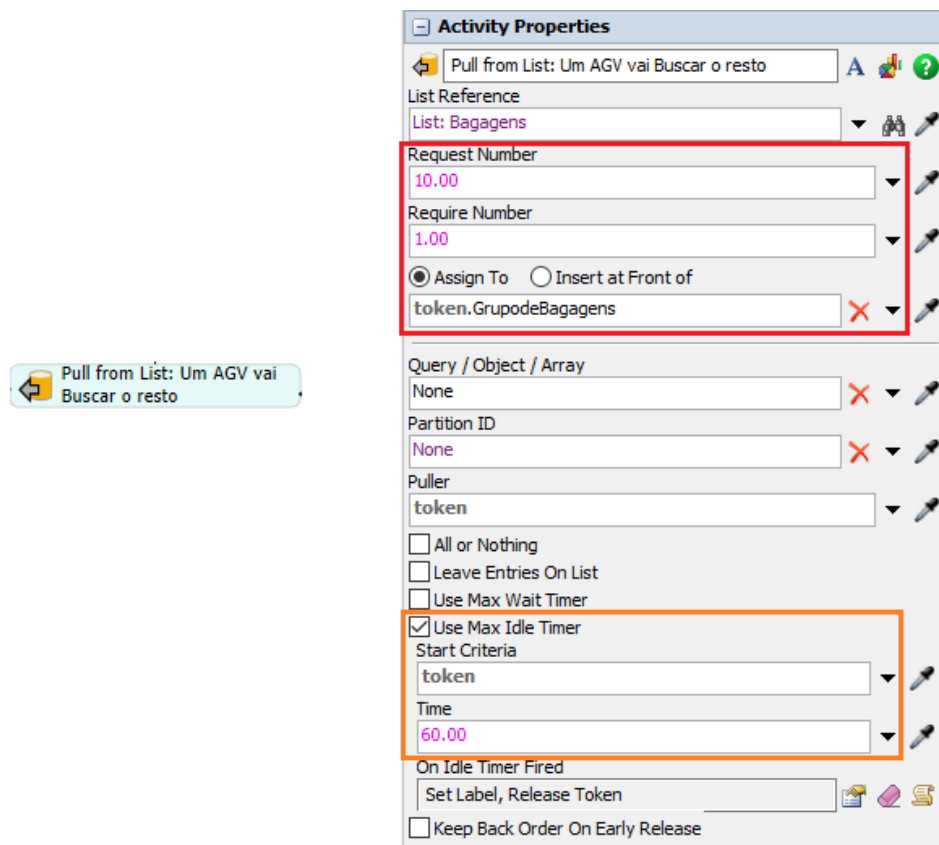


Figura 6.42: *Pull From List*, a) Atividade, b) Propriedades

Estas situações apenas acontecem caso um AGV esteja no percurso, entre o local de origem e o *stand* onde as bagagens se encontram, e o seguinte tenha chegado ao local de origem. O seguinte AGV irá avaliar se existem bagagens e, naquele momento, podem existir bagagens, porém o primeiro AGV já se encontra a caminho das mesmas e irá recolhê-las. Assim, o AGV iria ficar preso nesta atividade até que voltassem a aparecer bagagens naquela *queue*.

6.9.7 Bagagens de Partida

Após o primeiro *decide*, caso o AGV tenha verificado que existem bagagens no terminal para transportar, entra no primeiro “*Pull From List*” e é necessário recorrer à lista local criada pelo *event-triggered source* referente às bagagens de partida (ver Figura 6.43).

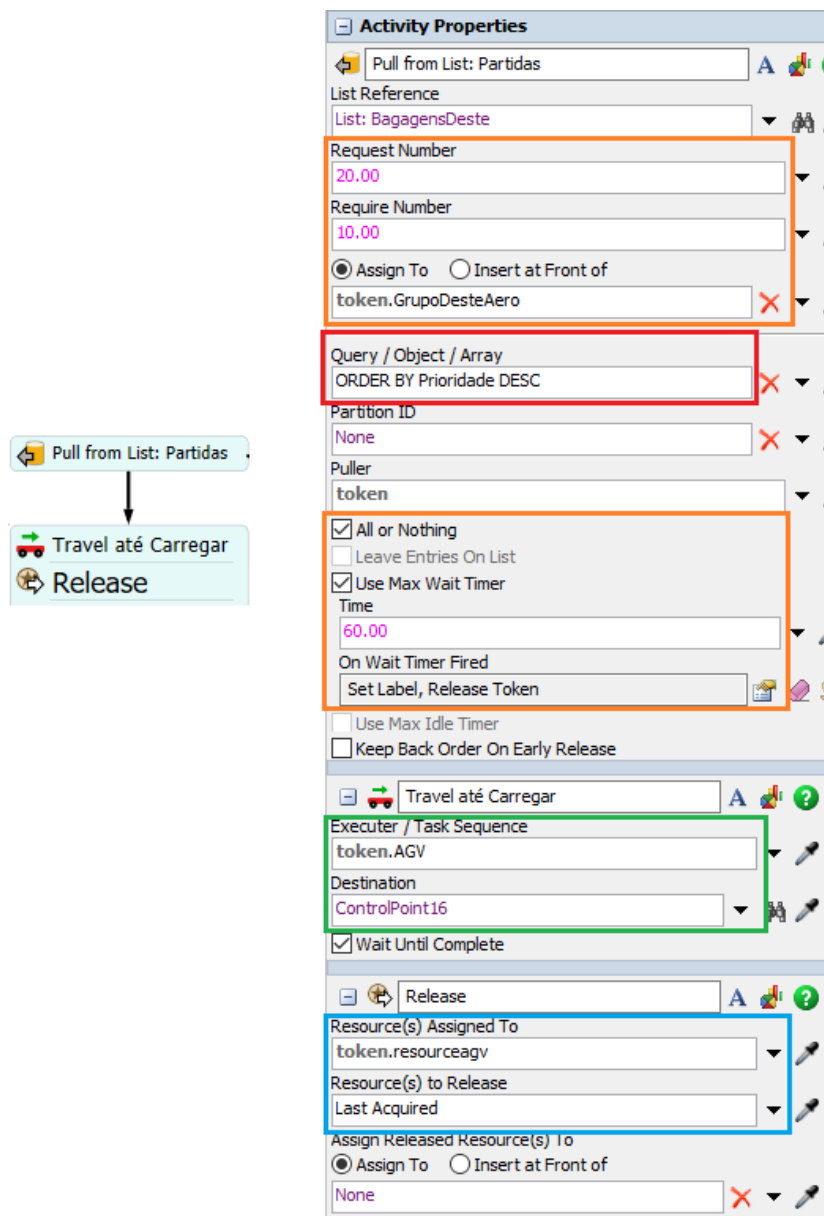


Figura 6.43: *Pull From List*, *Travel* e *Release*, a) Atividades, b) Propriedades

Primeiramente foi utilizado o “*Pull From List*”, onde as propriedades são semelhantes às encontradas no “*Pull From List*” das bagagens de chegada, à exceção de uma propriedade. Os AGV irão transportar, da mesma forma, um máximo de 20 bagagens e ter as propriedades “*All or Nothing*” e “*Max Wait Timer*” (retângulo laranja). Porém serão associadas ao *token.GrupoDesteAero*. Nesta atividade existe uma propriedade que não foi utilizada para as bagagens de chegada, que é a utilização da “*token.Prioridade*” onde as bagagens com prioridade são as

primeiras a serem transportadas para o terminal. A prioridade das bagagens é organizada de forma descendente (5% das bagagens são prioritárias).

Na atividade de *Travel*, o AGV irá seguir até ao *ControlPoint16*, local da *queue* onde surgem as bagagens de partida (retângulo verde). De seguida, da mesma forma que aconteceu com as bagagens de chegada, caso os AGV entrem nesta sequência é necessário dar o *release* do *token.resourceagv* (retângulo azul), de forma a que outros AGV entrem na atividade “*Wait For Event*”.

6.9.7.1 Carregar Bagagens e Viagem até ao Stand

Para o carregamento das bagagens é necessário recorrer a um *Run Sub Flow* e dentro desse recorrer a uma atividade *load* (ver Figura 6.44).

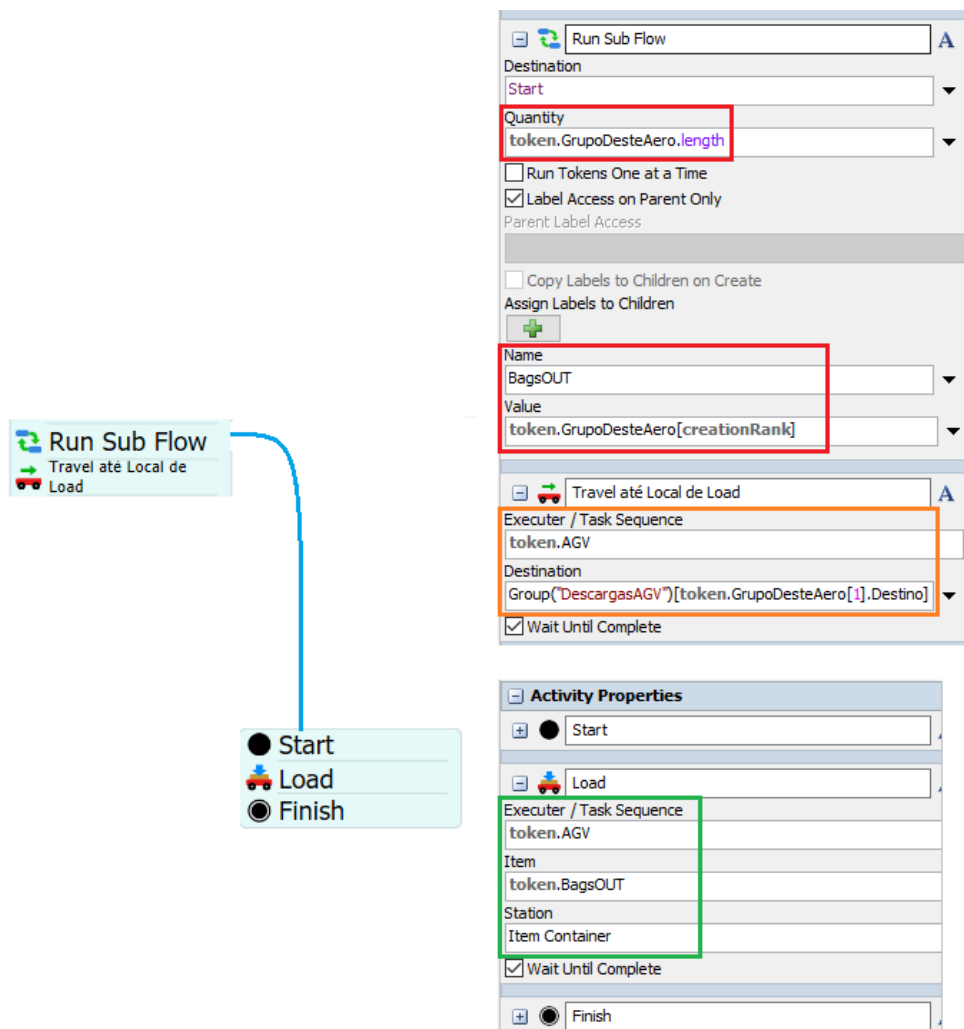


Figura 6.44: *Run Sub Flow* e *Travel* até stand, a) Atividades, b) Propriedades

Para o *Run Sub Flow* é necessário utilizar, como quantidade, o `token.GrupoDesteAero.length`, para que se utilize todas as bagagens que surgiram naquele instante na *queue* “BagagensDePartida”. Sendo o `token.GrupoDesteAero`, o *token* pai e o `token.BagsOut`, o *token* filho. Na atividade de *load* (retângulo verde), é indicado o AGV como local de destino das bagagens e o objeto a ser carregado é o `token.BagsOut` (*token* filho).

Na atividade de *travel* o destino tem um comando especial, `Group("DescargasAGV")[token.GrupoDesteAero[1].Destino]` (retângulo verde), onde é utilizado o grupo `DescargasAGV`, o *token* relativo ao conjunto de bagagens de partida e o `token.Destino` que é uma das colunas referentes à *schedule source* criada para as bagagens de partida.

6.9.7.2 Descarregar Bagagens e Viagem até ao Terminal

Para descarregar as bagagens foi utilizado outro *Run Sub Flow* e uma atividade de *Travel* onde os AGV retornam à posição de origem (ver Figura 6.45).

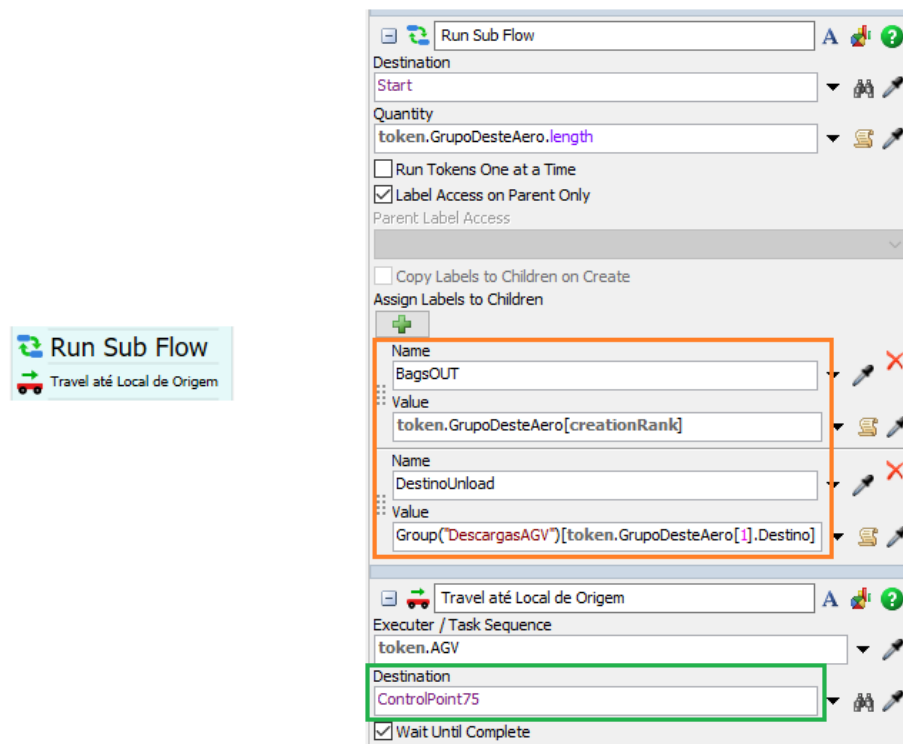


Figura 6.45: *Run Sub Flow* e *Travel* até terminal, a) Atividades, b) Propriedades

Para este *Run Sub Flow* foi necessário utilizar, no campo “Name”, dois *tokens* filhos (retângulo laranja): `BagsOut` referente às bagagens de partida; e `Desti-`

no *Unload* referente ao local onde os AGV irão realizar o *unload* das bagagens, que terá o valor da atividade *travel* referida na sub-subseção anterior.

A atividade *travel* faz com que o AGV se desloque até ao *ControlPoint75*, posição origem, posicionada entre a *sink* e a *queue* referente às bagagens de partida.

Em relação ao *sub flow* gerado (ver Figura 6.46), este tem a atividade *unload* e um *push to list*.

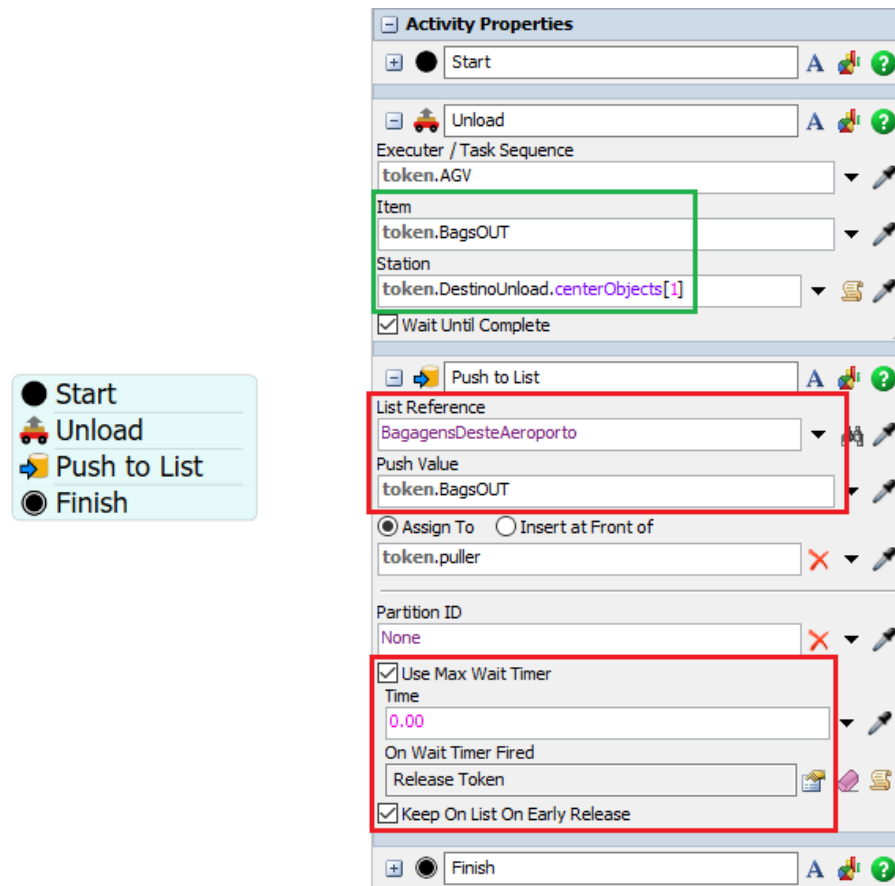


Figura 6.46: *Unload* e *Push To List*, a) Atividades, b) Propriedades

Na atividade de *unload* (retângulo verde), foi necessário utilizar os dois *tokens* filhos, *token.BagsOut* como item a descarregar e *token.DestinoUnload* como estação para onde descarregar. Com o auxílio de uma ligação do tipo S é possível utilizar o comando *.centerObjects[1]*, que irá enviar as bagagens para a *queue* que estiver ligada ao *ControlPoint* daquele *stand*.

Na atividade *Push To List* é necessário indicar o item a enviar para a lista global “PartidaDeBagagens”, *token.BagsOut* e, à semelhança do que aconteceu no

fluxo de aviões, são utilizadas as propriedades *Use Max Wait Timer* e *Keep On List On Early Release*. Como já foi referido anteriormente, estas complementam-se: o *Max Wait Timer* com 0 segundos faz com que seja um envio instantâneo (caso uma, ou mais, não o seja é dado *release* das mesmas) e por isso é utilizado o “*Keep On List On Early Release*” - esta segunda propriedade serve para que, mesmo que essas bagagens levem *release*, fiquem na lista até serem puxadas por um “*Pull From List*”, que neste caso é, o que se encontra no Fluxo de Aviões.

6.9.7.3 Há Mais Bagagens?

À semelhança das bagagens de chegada, recorreu-se à mesma ideia de ter um conjunto de dois *decides* (ver Figura 6.47), para verificar se existem mais de 10 bagagens e, se sim, voltar à sequência de carregamento com um máximo de 20 bagagens; caso não haja entra no segundo *decide*, onde verifica se a *queue* está vazia; caso esteja, retorna ao primeiro *decide* do fluxo de AGV; caso haja menos do que 10 bagagens na *queue* entra no último *Pull From List* desta sequência.

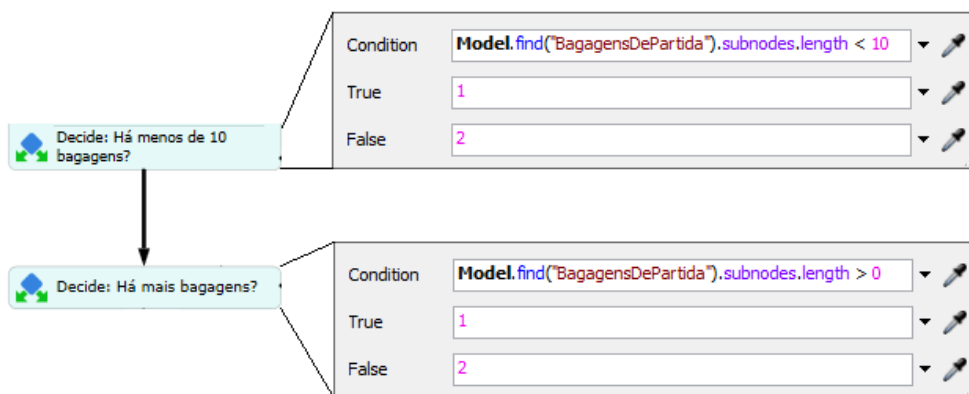


Figura 6.47: *Decides* bagagens de partida, a) Atividades, b) Propriedades

6.9.7.4 Último Pull From List

Neste último *Pull From List*, apenas um AGV irá levar as bagagens que restam ao *stand* onde o avião se encontra (ver Figura 6.48).

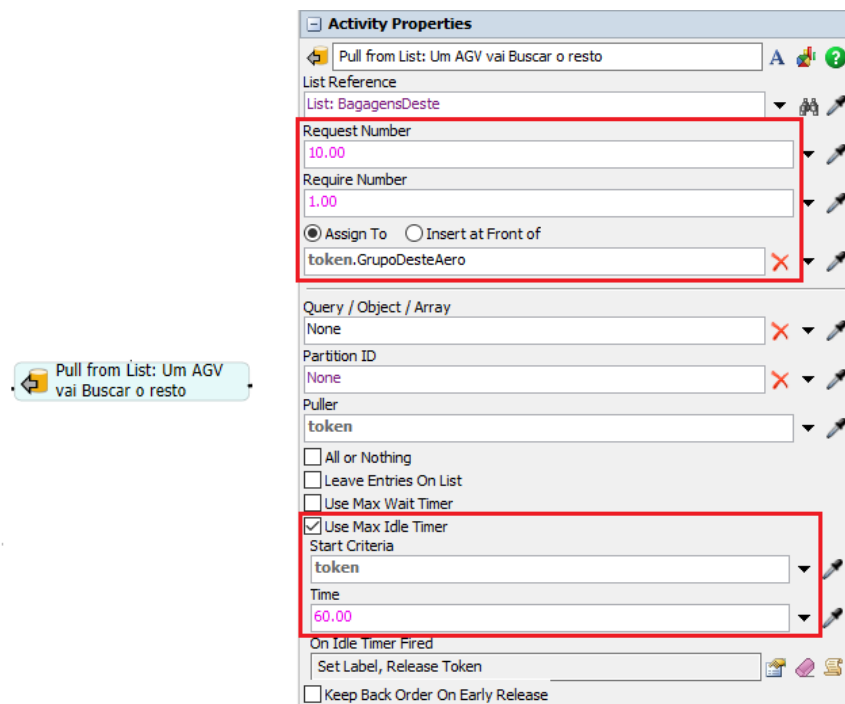
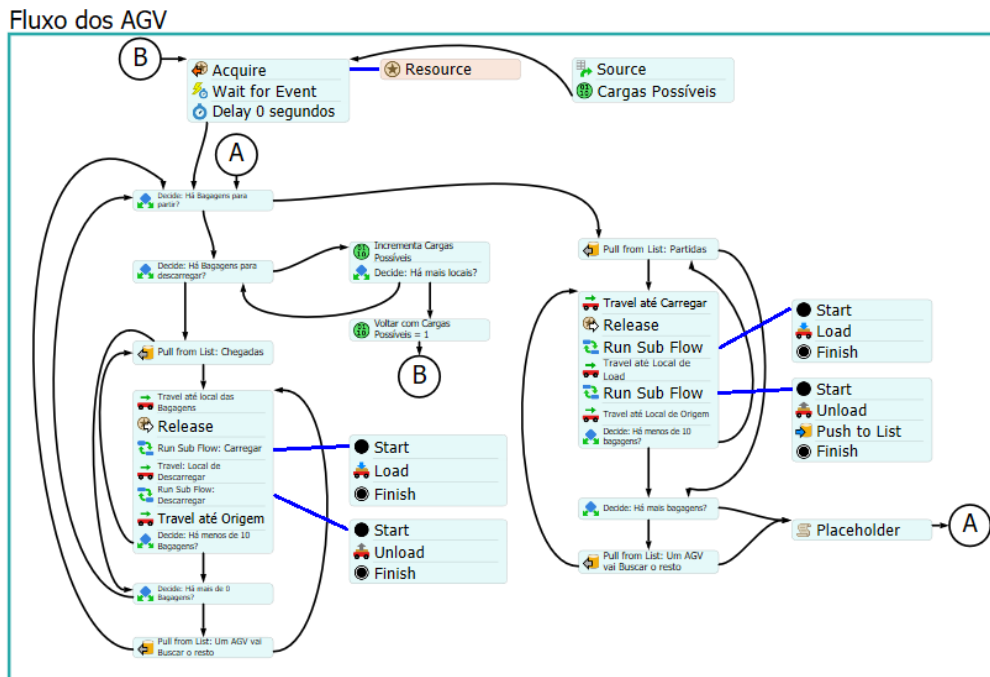


Figura 6.48: *Pull From List*, a) Atividade, b) Propriedades

Para tal, foi necessário introduzir um máximo de 10 bagagens e um mínimo de 1 bagagem para que o AGV realize a sequência; a este conjunto associa-se, de igual forma, o *token.GrupoDesteAero*. É, também, utilizada a propriedade “*Max Idle Timer*” com um tempo de inatividade máximo de 60 segundos.

6.9.7.5 Aspeto Final do Container Referente ao Fluxo de AGV

Depois de explicado, em detalhe, todas as atividades referentes à implementação do fluxo de AGV convém apresentar o aspeto do *container* (ver Figura 6.49) que engloba todas estas atividades.

Figura 6.49: Aspeto final do *container* do fluxo de AGV

6.9.8 Aspeto Final do Processflow

De forma a se ter uma ideia de tudo o que engloba o *processflow*, desde a criação de bagagens até ao fluxo de AGV e de aviões, é apresentado o aspeto final de todo o *processflow* (ver Figura 6.50). Assim, é possível verificar tudo o que foi feito para que a implementação deste projeto funcionasse, quando aliado com o desenho 3D criado e apresentado na secção “Mapeamento 3D”. As ligações representadas a azul não aparecem (a menos que se carregue nas diferentes listas), porém foram apresentadas para se perceber a que atividades estão associadas e se perceber que apenas as listas globais saltam entre fluxos enquanto que as listas locais se encontram dentro do mesmo fluxo.

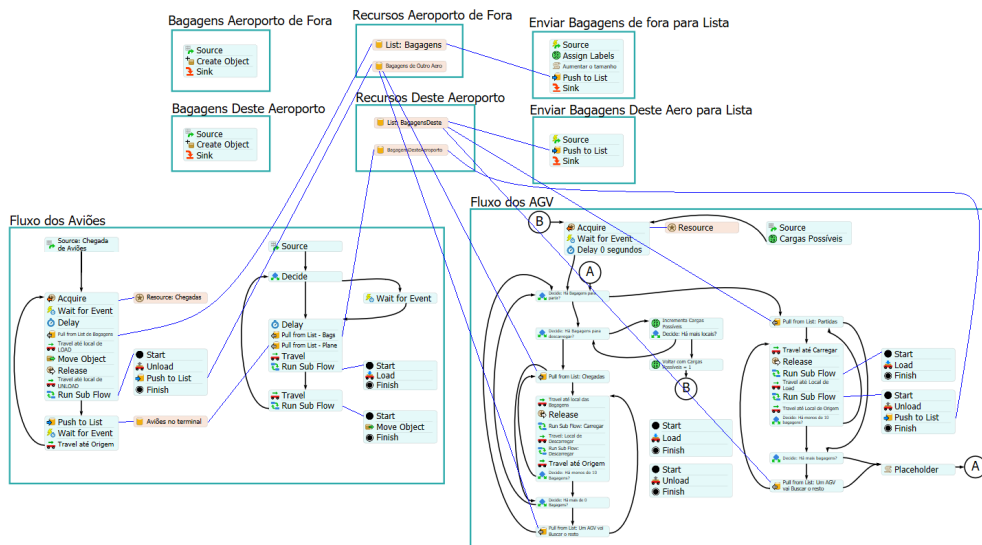


Figura 6.50: Aspeto final do *processflow*

6.10 Conclusão do Capítulo 6

Neste capítulo foi possível explicar e demonstrar todo o trabalho prático realizado neste projeto, o desenho do aeroporto em 3D, a criação de bagagens como um *flow item bin*, assim como a definição de prioridades nas mesmas, e a importação dos restantes objetos para tornar o projeto o mais realista possível.

Foi, também, explicada toda a implementação do *software* do projeto, ou seja, as implementações dos fluxos, tanto dos aviões, como dos AGV. A forma como foi implementada a ordem dos *stands*, tanto para os AGV, como para os aviões, e, por fim, um aspeto geral de todo o *processflow*.

Capítulo 7

Testes Realizados e Resultados Obtidos

Neste capítulo são apresentados todos os testes realizados assim como os resultados obtidos: os cenários (testes) englobam diferentes quantidades de AGV e o resultado é avaliado através do tempo de espera, este é o tempo que as bagagens de chegada demoram (a partir do momento em que saem do avião) a chegar ao terminal do aeroporto.

7.1 Valores das Schedule Sources

Após o projeto estar 100% funcional, ou seja, que os aviões tragam as bagagens e que este percurso esteja em comunicação com o percurso dos AGV através das bagagens, seguiu-se para o próximo passo, avaliar resultados. Para que este projeto tenha algum efeito em termos de resultados, é necessário avaliar o trabalho desenvolvido. Para tal, foi avaliado e analisado o tempo de espera pelas bagagens que chegam ao aeroporto, ou seja, o tempo que os passageiros irão esperar para que as suas bagagens cheguem ao terminal, em função dos AGV em funcionamento.

De forma a se ter resultados o mais próximo da realidade, foi utilizado um documento (ver Figura 7.1), facultado pelo Eng. André Siqueira, que permite verificar as horas de chegada e partida dos aviões durante 1 dia de trabalho. A figura representa o tempo que o avião está no aeroporto, desde que um avião chega a terra até ao momento em que parte. Por exemplo, o voo (TSC754 /TSC755 (YYZ-OPO-YYZ)), chega ao aeroporto às 8h da manhã e permanece no mesmo até ao fim do seu sombreado, ou seja, até às 9h45. Porém, como na realidade não chegam cinco aviões imediatamente ao mesmo tempo (lado esquerdo da figura), simulou-se a chegada dos mesmos a cada cinco minutos.

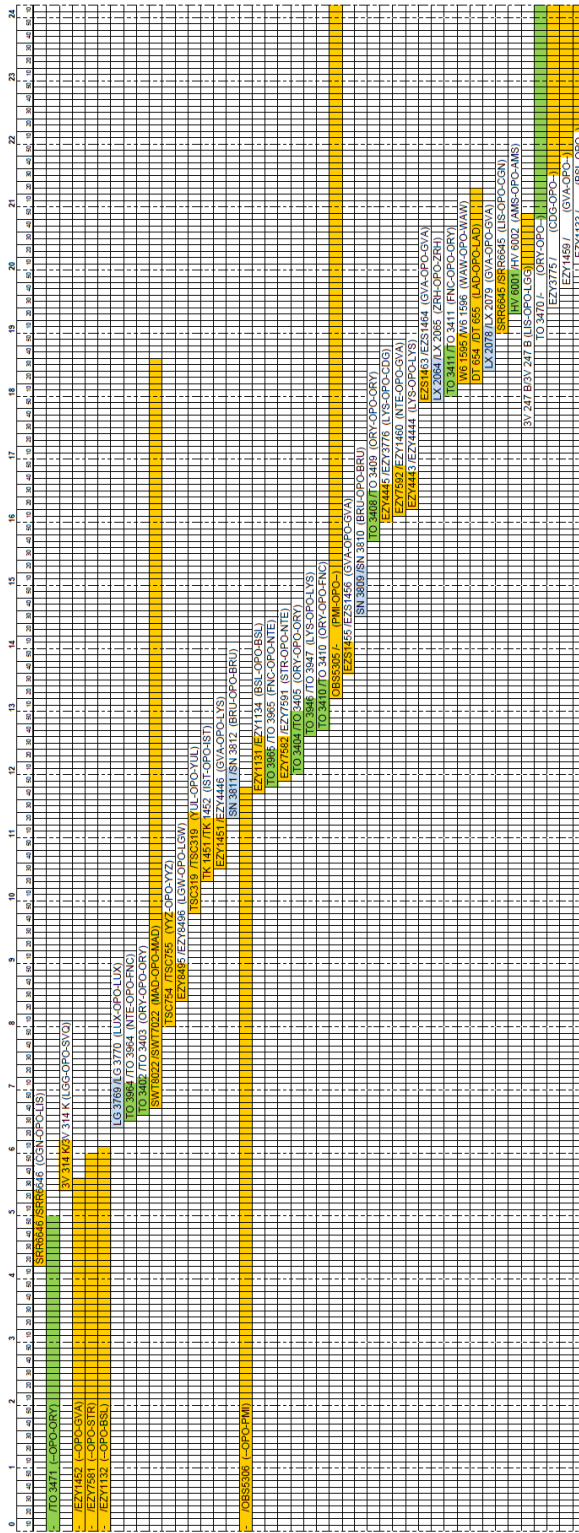


Figura 7.1: Planeamento dos aviões em terra

No Flexsim, isto traduz-se em números, números estes que serão o tempo, em segundos. Sendo as zero horas do dia (00h) o instante de tempo 0 segundos, os restantes irão ser contados a partir deste momento. A quantidade que surge em ambas as tabelas (ver Anexo A) é o número de bagagens em cada voo. A quantidade de bagagens é aleatória mas sempre com um número que faça sentido, de acordo com o analisado no Capítulo 5, onde se verificou a média de bagagens por voo.

Visto que o Flexsim começa sempre a simulação do projeto às 0h00, seguiu-se o planeamento a partir dessa hora até ao fim dos voos. Na totalidade foram 86 voos realizados neste dia (43 voos de chegada + 43 voos de partida). De forma a não ficar nenhum avião no aeroporto, no fim da simulação, também se simulou a partida dos últimos voos presentes no planeamento.

7.2 Testes com Vários Cenários

Nesta secção são demonstrados os diferentes testes com diferentes quantidades de AGV para o dia em que há o planeamento acima apresentado. Como apenas se teve acesso ao planeamento naquele dia e todos os dias há diferentes números de voos a entrar e sair do aeroporto, a escolha do melhor cenário, após as análises do testes, irá prender-se ao que tiver o melhor resultado face aos dados do planeamento verificado na figura 7.1.

Começaram-se os testes com 13 AGV em funcionamento e com todos estes a ir a todas as entradas, ou seja, não há nenhum AGV que seja fixo a determinado *stand*. Cada *stand* é um dos locais onde as bagagens possam estar. No total existem 9 *stands*. Os dados foram extraídos para uma folha de cálculo em Excel e, assim, é possível verificar os tempos (médio, mínimo e máximo) de espera em cada *stand* (saída).

7.2.1 Cenário A: 13 AGV

Realizou-se o teste com 13 AGV, em grupos de 6 (ver Tabela 7.1). Ou seja, os AGV são divididos em dois grupos de 6 AGV, sempre que chega (ao aeroporto ou ao terminal) um grupo de bagagens irão, somente, 6 AGV recolher essas bagagens. Caso surgem, quase no mesmo instante, outro grupo de bagagens, os segundo grupo de 6 AGV irá recolher esse montante. Se tivessem todos os AGV a recolher o primeiro montante de bagagens, só no fim do percurso é que iriam recolher o segundo montante. Como são 13 AGV e são 2 grupos de 6, irá existir sempre 1 AGV que vai “descansar” e, assim, pode carregar as baterias durante esse tempo. Para este teste, obtiveram-se os seguintes resultados.

Tabela 7.1: Resultados obtidos com 13 AGV

	13 AGV		
Saídas	Tempo Médio	Tempo Mínimo	Tempo Máximo
Saída 1	2364,73	907,25	4680,32
Saída 2	2077,25	1096,05	3293,49
Saída 3	2044,29	1128,16	3619,45
Saída 4	1982,08	1089,52	3748,39
Saída 5	1996,6	1071,11	2686,81
Saída 6	1648,69	980,09	3189,08
Saída 7	1566,04	766,98	2491,11
Saída 8	1743,42	929,47	2615,42
Saída 9	1973,71	865,64	2379,52
Média	1932,98	981,59	3189,29

Neste cenário, o tempo médio de espera foi de 1932,98 segundos, ou seja, 32 minutos e 13 segundos. O tempo mínimo de espera pelas bagagens prioritárias é de 766,98 segundos, ou seja, 12 minutos e 47 segundos. O tempo máximo de espera foi de 4680,32 segundos, ou seja, 78 minutos e 1 segundo.

De seguida, verificou-se o tempo mínimo de espera para bagagens não prioritárias (ver Figura 7.2), de forma a se comparar os dois tempos.

13 AGV	
Saída 1	<u>1904,25</u>
Saída 2	<u>1776,45</u>
Saída 3	<u>1705,96</u>
Saída 4	<u>1661,3</u>
Saída 5	<u>1771,51</u>
Saída 6	<u>1547,89</u>
Saída 7	<u>1504,79</u>
Saída 8	<u>1477,27</u>
Saída 9	<u>2125,5</u>
Média	<u>1719,44</u>

Figura 7.2: Tempos de espera com 13 AGV para bagagens não prioritárias

Assim, pode-se concluir que com 13 AGV, a média do tempo de espera mínimo, pelas bagagens prioritárias, foi de 981.59 segundos, ou seja, 16 minutos e 22 segundos. Enquanto que, a média do tempo de espera mínimo, pelas bagagens não prioritárias, foi de 1719.44 segundos, ou seja, 28 minutos e 39 segundos.

7.2.2 Cenário B: 14 AGV

De seguida, realizou-se o teste com 14 AGV, em grupos de 7 (ver Tabela 7.2). Neste caso, quando chega um grupo de bagagens, irmão 7 AGV recolher esse montante.

Tabela 7.2: Resultados obtidos com 14 AGV

	14 AGV		
Saídas	Tempo Médio	Tempo Mínimo	Tempo Máximo
Saída 1	2114,48	1097,25	4680,32
Saída 2	1804,47	898,97	2567,26
Saída 3	1889,45	1128,16	2739,8
Saída 4	1995,06	1089,52	3908,36
Saída 5	2004,46	1071,11	2802,13
Saída 6	1742,99	980,09	3596,5
Saída 7	1545,24	766,98	2319,4
Saída 8	1744,28	929,47	2513,4
Saída 9	1981,37	865,64	2379,52
Média	1869,09	980,8	3056,3

Neste cenário, o tempo médio de espera foi de 1869.09 segundos, ou seja, 31 minutos e 9 segundos. O tempo mínimo e máximo, de espera, são os mesmos do cenário anterior.

De seguida, verificou-se, a média do tempo mínimo de espera para bagagens não prioritárias (ver Figura 7.3), de forma a se comparar os dois tempos.

14 AGV	
Saída 1	<u>1685,05</u>
Saída 2	<u>1776,45</u>
Saída 3	<u>1705,96</u>
Saída 4	<u>1665,19</u>
Saída 5	<u>1638,91</u>
Saída 6	<u>1547,89</u>
Saída 7	<u>1504,79</u>
Saída 8	<u>1477,27</u>
Saída 9	<u>2154,01</u>
Média	<u>1683,95</u>

Figura 7.3: Tempos de espera com 14 AGV para bagagens não prioritárias

Assim, pode-se concluir que com 14 AGV, a média do tempo de espera mínimo, pelas bagagens prioritárias, foi de 980,8, ou seja, de 16 minutos e 21 segundos. Enquanto que a média do tempo de espera mínimo pelas bagagens não prioritárias foi de 1683.95 segundos, ou seja, 28 minutos e 4 segundos.

7.2.3 Cenário C: 15 AGV

Este teste foi realizado com 15 AGV (ver Tabela 7.3) e, à semelhança do Cenário A, um dos AGV irá descansar, visto que são dois grupos de 7 AGV.

Tabela 7.3: Resultados obtidos com 15 AGV

	15 AGV		
Saídas	Tempo Médio	Tempo Mínimo	Tempo Máximo
Saída 1	1936,85	1097,25	4319,43
Saída 2	1783,46	898,97	2367,98
Saída 3	1904,28	1128,16	2607,19
Saída 4	1922,29	1089,52	3643,65
Saída 5	1968,29	1071,11	2669,53
Saída 6	1617,73	980,09	2669,29
Saída 7	1539,92	766,98	2351,47
Saída 8	1782,36	929,47	2484,89
Saída 9	1973,71	865,64	2379,52
Média	1825,43	980,8	2832,55

Neste cenário, o tempo médio de espera foi de 1825.43 segundos, ou seja, 30 minutos e 25 segundos. O tempo mínimo de espera é o mesmo do cenário anterior. Enquanto que o tempo máximo de espera desceu para 4319,43 segundos, ou seja, 71 minutos e 59 segundos.

De seguida, verificou-se a média do tempo mínimo de espera para bagagens não prioritárias (ver Figura 7.4), de forma a se comparar os dois tempos.

15 AGV	
Saída 1	1685,05
Saída 2	1776,45
Saída 3	1705,96
Saída 4	1667,14
Saída 5	1638,91
Saída 6	1547,89
Saída 7	1504,79
Saída 8	1477,27
Saída 9	2125,5
Média	1681

Figura 7.4: Tempos de espera com 15 AGV para bagagens não prioritárias

Assim, pode-se concluir que com 15 AGV, a média do tempo de espera mínimo pelas bagagens prioritárias foi de 980,8, ou seja, de 16 minutos e 21 segundos. Enquanto que a média do tempo de espera mínimo pelas bagagens não prioritárias, foi de 1681 segundos, ou seja, 28 minutos e 1 segundos.

Os tempos de espera mínimos são exatamente iguais para as bagagens prioritárias, quando comparado com o cenário anterior, com 14 AGV.

7.2.4 Cenário D: 16 AGV

De seguida, foi realizado um teste com 16 AGV, em grupos de 8 (ver Tabela 7.4).

Tabela 7.4: Resultados obtidos com 16 AGV

16 AGV			
Saídas	Tempo Médio	Tempo Mínimo	Tempo Máximo
Saída 1	1890,14	1097,25	3480,25
Saída 2	1771,36	898,97	2325,78
Saída 3	1869,83	1128,16	2561,38
Saída 4	1855,89	1089,52	3510,55
Saída 5	1934,36	1071,11	2536,93
Saída 6	1603,43	980,09	2568,52
Saída 7	1534,62	766,98	2316,48
Saída 8	1801,88	929,47	2513,4
Saída 9	1981,37	865,64	2379,52
Média	1804,76	980,8	2688,09

Neste cenário, o tempo médio de espera foi de 1804,76 segundos, ou seja, 30 minutos e 5 segundos. O tempo mínimo de espera é o mesmo do cenário anterior.

Enquanto que o tempo máximo de espera desceu para 3510,55 segundos, ou seja, 58 minutos e 31 segundos.

De seguida, verificou-se a média do tempo mínimo de espera para bagagens não prioritárias (ver Figura 7.5), de forma a se comparar os dois tempos.

16 AGV	
Saída 1	<u>1685,05</u>
Saída 2	<u>1776,45</u>
Saída 3	<u>1705,96</u>
Saída 4	<u>1657,14</u>
Saída 5	<u>1638,91</u>
Saída 6	<u>1547,89</u>
Saída 7	<u>1504,79</u>
Saída 8	<u>1477,27</u>
Saída 9	<u>2154,01</u>
Média	<u>1683,05</u>

Figura 7.5: Tempos de espera com 16 AGV para bagagens não prioritárias

Assim, pode-se concluir que com 16 AGV, a média do tempo de espera mínimo, pelas bagagens prioritárias, foi de 980,8 segundos, ou seja, de 16 minutos e 21 segundos. Enquanto que a média do tempo de espera mínimo pelas bagagens não prioritárias, foi de 1683,05 segundos, ou seja, 28 minutos e 3 segundos.

7.2.5 Cenário E: 17 AGV

Por fim, foi realizado um último teste com 17 AGV (ver Tabela 7.5). À semelhança dos Cenários anteriores com número ímpar de AGV, este também terá um AGV que irá descansar.

Tabela 7.5: Resultados obtidos com 17 AGV

	17 AGV		
Saídas	Tempo Médio	Tempo Mínimo	Tempo Máximo
Saída 1	1824,89	1097,25	3248,97
Saída 2	1770,39	890,42	2410,97
Saída 3	1883,65	1128,16	2561,38
Saída 4	1814,74	1089,52	3034,19
Saída 5	1957,16	1071,11	2685,27
Saída 6	1531,62	980,09	2600,33
Saída 7	1529,2	766,98	2271,74
Saída 8	1775,31	929,47	2626,57
Saída 9	2011,77	865,64	2379,52
Média	1788,75	979,85	2646,55

Neste cenário, o tempo médio de espera foi de 1788,75 segundos, ou seja, 29 minutos e 49 segundos. O tempo mínimo de espera é o mesmo dos cenários anteriores. Enquanto que o tempo máximo de espera desceu para 3248,97 segundos, ou seja, 54 minutos e 9 segundos.

De seguida, verificou-se a média do tempo mínimo de espera para bagagens não prioritárias (ver Figura 7.6), de forma a se comparar os dois tempos.

17 AGV	
Saída 1	<u>1685,05</u>
Saída 2	<u>1776,45</u>
Saída 3	<u>1705,96</u>
Saída 4	<u>1657,14</u>
Saída 5	<u>1638,91</u>
Saída 6	<u>1547,89</u>
Saída 7	<u>1504,79</u>
Saída 8	<u>1477,27</u>
Saída 9	<u>2267,18</u>
Média	<u>1695,63</u>

Figura 7.6: Tempos de espera com 17 AGV para bagagens não prioritárias

Assim, pode-se concluir que com 17 AGV, a média do tempo de espera mínimo, pelas bagagens prioritárias, foi de 979.85, ou seja, de 16 minutos e 20 segundos. Enquanto que a média do tempo de espera mínimo pelas bagagens não prioritárias, foi de 1695.63 segundos, ou seja, 28 minutos e 16 segundos.

Ainda se realizou o teste com 18 AGV porém os tempos começaram a subir, pois estariam demasiados AGV para este dia e iriam “atrapalhar-se” entre si.

7.3 Tabela Comparativa

Para se definir qual seria o melhor cenário para o dia em que foi feito este planejamento, foi realizada uma tabela comparativa (ver Tabela 7.6) onde é possível verificar os diferentes tempos máximos, mínimos e os tempos médios de espera, desde os testes com 13 AGV até ao teste realizado com 17 AGV. Nesta tabela surgem duas siglas: BP para Bagagens Prioritárias e BNP para Bagagens Não Prioritárias. Os tempos mínimos são sempre os mesmos devido ao facto de: para as bagagens prioritárias, como são as primeiras a serem recolhidas, o AGV que irá carregar e descarregar as mesmas, demora sempre o mesmo tempo; enquanto que o tempo de espera mínimo para as bagagens não prioritárias é sempre o mesmo devido ao facto de, no último voo de chegada, os AGV estarem livres e, assim, as bagagens são, imediatamente, descarregadas.

Tabela 7.6: Tabela comparativa com os valores de tempo de espera

Tempo	13 AGV	14 AGV	15 AGV	16 AGV	17 AGV
Máximo	4680,32	4680,32	4319,43	3510,55	3248,97
Mínimo BP	766,98	766,98	766,98	766,98	766,98
Mínimo BNP	1477,27	1477,27	1477,27	1477,27	1477,27
Médio	1932,98	1869,09	1825,43	1804,76	1788,75

Depois de se verificarem estes resultados, pode-se descartar os cenários com 13 e 14 AGV, visto que ambos apresentam um tempo máximo de espera mais alto do que os restantes.

De seguida, ao serem comparados os 3 restantes, se o cenário com 15 AGV for separado dos restantes, verificou-se que a média dos tempos mínimos de espera por bagagens não prioritárias é menor no Cenário C do que nos Cenários D e E, apesar dos valores mínimos serem os mesmos. Porém, o tempo de espera máximo é muito mais alto.

O cenário com 16 AGV, quando comparado ao cenário com 17 AGV, difere de 16 segundos no tempo médio, porém no tempo máximo, o cenário D (16 AGV) demora mais 262 segundos (4 minutos e 22 segundos) do que o cenário E (17 AGV). Enquanto que, se forem comparados a média dos tempos de espera mínimos por bagagens não prioritárias, o cenário com 17 AGV demora mais 13 segundos. Entre estes dois cenários, o cenário com 17 AGV, apesar do seu tempo mínimo por bagagens não prioritárias ser mais alto por poucos segundos, ganha

4 minutos no tempo máximo de espera, pelo que é melhor em relação ao cenário com 16 AGV.

Posto isto, o aeroporto, neste dia, iria ter de escolher o cenário consoante o que fosse mais prioritário. Com o cenário C (15 AGV), em relação ao cenário E (17 AGV), iria perder perto de 1100 segundos no tempo máximo de espera num dos voos e 37 segundos no tempo médio de espera ao longo do dia, porém, como verificado nas tabelas presentes nas subsecções dos dois cenários, ganhar 15 segundos no tempo de espera mínimo (nas bagagens não prioritárias). Se a prioridade for diminuir o tempo de espera máximo e o tempo médio ao longo dia, o cenário a escolher será o Cenário E, com 17 AGV. Porém, se a prioridade for diminuir o tempo mínimo de espera para as bagagens não prioritárias, o melhor cenário será o Cenário C, com 15 AGV.

7.4 Conclusão do Capítulo 7

Neste capítulo apresentaram-se os resultados dos testes com diferentes cenários - os cenários apresentados têm 13, 14, 15, 16 ou 17 AGV. Após se efectuarem os vários testes, verificou-se que os melhores cenários a usar neste dia, seriam: ou Cenário C (15 AGV) ou o Cenário E (17 AGV), consoante o objetivo do aeroporto.

O tempo médio de espera, entre os cenários com 15 e 17 AGV, não é muito relevante visto que, em média, seriam menos 37 segundos de espera por bagagens; porém, olhando ao tempo máximo de espera, em relação ao cenário com 17 AGV, os passageiros iriam esperar menos 1100 segundos do que o tempo que estariam à espera caso estivessem com 15 AGV em pista.

Capítulo 8

Conclusões

Neste capítulo serão apresentadas as principais conclusões, conhecimentos adquiridos e eventuais melhorias a serem feitas.

8.1 Conclusão do Trabalho

Com este projeto foi possível obter conhecimentos sobre várias áreas, mais especificamente na área de AGV e robôs móveis, em tarefas de logística. Foi também possível ter conhecimento de vários *softwares* que permitissem simular robôs móveis.

Assim obteve-se vários conhecimentos sobre como trabalhar no *software* Flexsim pois teve de se realizar o *design* 3D do aeroporto, a importação dos objetos, realizar os percursos para os AGV e para os aviões e, de seguida, realizar a programação para cada um dos percursos. No final, realizou-se a sincronização dos percursos através das bagagens.

Por fim, realizou-se o planeamento do dia através de um documento do Aeroporto Francisco Sá Carneiro e concluiu-se que o melhor cenário seria a utilização de 17 AGV.

8.2 Ideias Para Desenvolvimentos Futuros

Como ideias para desenvolvimento futuro existem duas principais pelo que foi possível verificar.

A primeira ideia seria chegar com os AGV a mais pontos do aeroporto, ou seja, em vez de chegar apenas até aos 9 *stands* que se encontram mais próximos ao terminal, chegar também aos restantes pontos onde os aviões possam parar

para carregar e descarregar bagagens/ passageiros. Com isso seria, também, necessário realizar simulações de prioridade entre AGV e aviões de forma a se evitarem colisões.

A segunda ideia seria conseguir implementar o uso das bagagens em transferência; tal como já foi mencionado, são bagagens que não chegam ao terminal do aeroporto, visto que seguem, de imediato, para outro avião. Estas situações acontecem em voos com escalas.

Há ainda um pormenor que melhoraria o aspecto do projeto, pois não se conseguiu realizar devido a essa opção no modelo importado para as bagagens não permitir: mudar a cor das bagagens prioritárias em vez de lhes aumentar o tamanho. Isto daria um melhor aspeto e seria mais fácil interpretar as bagagens que chegassem.

Referências Bibliográficas

- [1] Braverabbit, “Autonomous robots moving in automated warehouse.” <https://www.istockphoto.com/video/autonomous-robots-moving-shelves-in-automated-warehouse-seamless-looping-gm1205786360-347488410>, Jan 2020. [cited on p. xv, 7]
- [2] “Linde c-matic, automated guided cart (agc).” https://www.linde-mh.com/en/technical/News-Detail_53184.html, Jan 2019. [cited on p. xv, 6, 7]
- [3] “Translift 4.0 - agv rebocador.” <https://www.transliftbr40.com/copia-agv-plataforma>. [cited on p. xv, 7, 8]
- [4] “Lift deck.” <https://www.jbtc.com/en/emea/automated-systems/products-and-applications/products/unit-load-agvs/lift-deck>. [cited on p. xv, 8, 9]
- [5] L. D. Express. <https://cacm.acm.org/news/234461-fleet-of-robots-to-park-cars-at-gatwick-airport/fulltext>, Feb 2019. [cited on p. xv, 10]
- [6] RoboticAutomationPL, “Manual forklift and agv in one!” https://www.youtube.com/watch?v=_onQ56w7JT0, Aug 2011. [cited on p. xv, 11]
- [7] A. Sharma, “20,000 autonomous mobile robots shipped in 2018.” <https://www.interactanalysis.com/20000-autonomous-mobile-robots-shipped-in-2018/>, Oct 2020. [cited on p. xv, 11]
- [8] H. Alves, “Navegação e controlo de um veículo móvel.” <https://repositorio-aberto.up.pt/bitstream/10216/58480/1/000142643.pdf>, Mar 2010. [cited on p. xv, 13]
- [9] “Automation and innovation for an efficient production line.” <https://www.indevagroup.com/news/industrial-automation-automatic-guided-vehicles-production-line/>, Nov 2020. [cited on p. xv, 13]

- [10] A. Thomsen, “Monte um robô seguidor de linha com arduino motor shield.” <https://www.filipeflop.com/blog/projeto- robo-seguidor-de-linha-arduino/>, Oct 2020. [cited on p. xv, 15]
- [11] “Incremental rotary encoders for automated guided vehicle (agv) control.” <https://www.quantumdev.com/incremental-rotary-encoders-for-automated-guided-vehicle-agv-control/>, Jan 2020. [cited on p. xv, 15]
- [12] “Agv control system.” <https://www.ikvrobot.com/agv/agv-control-system.html>, Jan 2014. [cited on p. xv, 16]
- [13] “Humber river, north america’s first fully digital hospital!” <https://www.bouygues-construction.com/en/press/news/humber-river-north-americas-first-fully-digital-hospital>, Apr 2017. [cited on p. xv, 20]
- [14] <https://blooloop.com/showcase/agv-dark-rides/>, Jan 2019. [cited on p. xv, 21]
- [15] S. Song, D. Choi, J. Hur, M. Lee, Y. J. Park, and J. Kim, “Development and application of mobile robot system for marking process in lngc cargo tanks.” <https://ieeexplore.ieee.org/document/5335175>, Aug 2009. [cited on p. xv, 23]
- [16] J. Lau, “Research edition of fetch & freight.” <https://fetchrobotics.com/fetch-robotics-blog/research-edition-fetch-freight/>, Aug 2019. [cited on p. xv, 24]
- [17] R. Chakma, S. S. Mahtab, S. Akter Milu, I. S. Emon, S. Sunjida Ahmed, M. J. Alam, M. Rahman Morshed, and L. Xiangyang, “Navigation and tracking of agv in ware house via wireless sensor network.” <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9077099>, 2019. [cited on p. xv, 25]
- [18] M. Hamid, A. Adom, N. Rahim, and M. Rahiman, “Navigation of mobile robot using global positioning system (gps) and obstacle avoidance system with commanded loop daisy chaining application method.” <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5069211>, march 2009. [cited on p. xv, 25, 26]
- [19] D. Edwards, “invia robotics expands picking system capabilities.” <https://roboticsandautomationnews.com/2019/11/19/invia-robotics-expands-picking-system-capabilities/26779/>, Nov 2019. [cited on p. xv, 26]
- [20] M. Saitoh, Y. Takahashi, A. Sankaranarayanan, H. Ohmachi, and K. Marukawa, “A mobile robot tested with manipulator for security guard application.” <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=525637>, May 1995. [cited on p. xv, 27]

- [21] “Flexsim 3d simulation modeling software.” <https://www.flexsim.com/flexsim/>, Nov 2020. [cited on p. xv, 30]
- [22] I. Sheludko and V. Bechard, “Flexsim pricing, features, reviews & comparison of alternatives.” <https://www.getapp.com/it-management-software/a/flexsim/>, Jun 2019. [cited on p. xv, 31]
- [23] “Visual components essentials - our core 3d manufacturing simulation product.” <https://www.visualcomponents.com/products/visual-components/essentials/>. [cited on p. xv, 32]
- [24] T. TARKAN, “Restaurant simulation model and report.” <https://www.simulationhelpdesk.com/downloads/fast-food-restuarant-simulation-model/>, Jul 2020. [cited on p. xvi, 33]
- [25] S. Corporation, “Simul8 pricing, reviews, features - free demo.” <https://www.softwaresuggest.com/us/simul8>, 2019. [cited on p. xvi, 34]
- [26] “Simulation modeling technics.” <https://focus-grp.com/simulation-techniques/>, 2020. [cited on p. xvi, 35]
- [27] A. Siqueira, “Gestão aeroportuária e turismo,” Jan 2021. [cited on p. xvi, 43, 44]
- [28] “The history of agv development.” <http://www.agvblog.com/51.html>, Oct 2019. [cited on p. 5]
- [29] “Types of agv with examples.” <https://www.agvnetwork.com/types-of-automated-guided-vehicles>. [cited on p. 6]
- [30] DAIFUKU, “Automated guided vehicles & carts.” <https://www.daifuku.com/us/solution/vehiclecart/>. [cited on p. 6]
- [31] “Automated vehicle options.” <https://motioncontrolsrobotics.com/robotic-applications/automated-material-handling/automatic-guided-carts-agc/>, Oct 2020. [cited on p. 6]
- [32] “Agv network home.” <https://www.agvnetwork.com/types-of-automated-guided-vehicles#agc>. [cited on p. 6]
- [33] “Towing agv.” <https://www.jbtc.com/en/emea/automated-systems/products-and-applications/products/towing-agvs>, 2020. [cited on p. 7]
- [34] R. Wydra, “Diferentes tipos de veículos auto-guiados.” <https://www.sinova.com.br/2020/04/23/conheca-os-tipos-de-veiculos-autoguiados/>, Apr 2020. [cited on p. 7]

- [35] “Unit load deck agv - agv systems.” <https://www.dematic.com/sv-se/products/products-overview/agv-systems/unit-load-plattform/>. [cited on p. 8]
- [36] MHI, “What kind of automatic guided vehicle is the right machine for the job?.” <http://s354933259.onlinehome.us/mhi-blog/what-kind-of-automatic-guided-vehicle-is-the-right-machine-for-the-job/>, Oct 2018. [cited on p. 8]
- [37] “Automatic forklift robots for material handling.” <https://www.agvnetwork.com/automatic-forklift-agv>. [cited on p. 9]
- [38] “Forkover.” <https://www.jbtc.com/en/north-america/automated-systems/products-and-applications/products/forked-agvs/forkover>, 2020. [cited on p. 9]
- [39] “Automated pallet mover, discovering the self driving pallet jack.” <https://www.agvnetwork.com/automated-pallet-mover>. [cited on p. 9]
- [40] “Counterbalanced.” <https://www.agvnetwork.com/automatic-forklift-agv#counterbalance>. [cited on p. 10]
- [41] <https://www.jbtc.com/en/north-america/automated-systems/products-and-applications/products/forked-agvs/counterbalance>, 2020. [cited on p. 10]
- [42] “Reach.” <https://www.jbtc.com/en/north-america/automated-systems/products-and-applications/products/forked-agvs/reach>. [cited on p. 10]
- [43] “Automated forklift robot system.” <https://www.agvnetwork.com/automatic-forklift-agv#reach-truck>. [cited on p. 11]
- [44] E. Romaine and B. Gionet, “Autonomous mobile robot (amrs) types and uses.” <https://www.conveyco.com/types-and-applications-of-amrs/>, Aug 2020. [cited on p. 12]
- [45] T. Stobierski, “Autonomous mobile robots.” <https://www.conveyco.com/technology/autonomous-mobile-robots-amrs/>, Jan 2021. [cited on p. 12]
- [46] “What is lidar.” <https://oceanservice.noaa.gov/facts/lidar.html>, Oct 2012. [cited on p. 12]
- [47] “Technology in focus: Bathymetric lidar.” <https://www.gim-international.com/content/article/technology-in-focus-bathymetric-lidar-2>, Nov 2016. [cited on p. 12]

- [48] J. Oliveira, “Desenvolvimento de um protótipo de veículo auto guiado para célula produtiva.” <https://lactec.org.br/dissertacoes/69-desenvolvimento-de-um-prototipo-de-veiculo-auto-guiado-para-celula-produtiva/>, 2011. [cited on p. 13]
- [49] R. Catarrinho and F. Simões, “Contribuição para o fabrico de um veículo guiado automaticamente.” <https://comum.rcaap.pt/bitstream/10400.26/25378/1/Rui-Francisco-Subtil-Matos-Catarrinho.pdf>, 2018. [cited on p. 13, 15, 16, 17]
- [50] J. Martinez, “Agvs: navigation systems.” <https://medium.com/@jlmartinez.es/agvs-navigation-systems-354b3469e6c3>, Apr 2020. [cited on p. 14]
- [51] H. Dhouibi, “Que agv escolher?.” <http://guide.directindustry.com/pt/que-agv-escolher/#2>, Sep 2019. [cited on p. 14]
- [52] C. J. Snyman, “Development of a navigation system for an autonomous guided vehicle using android technology.” <https://core.ac.uk/download/pdf/145049745.pdf>, Dec 2012. [cited on p. 14, 15]
- [53] N. Taglic, “What are automated guided vehicles?.” <https://6river.com/what-are-automated-guided-vehicles/>, Aug 2020. [cited on p. 14, 15, 17]
- [54] D. Thompson. https://www.me.ua.edu/me364/PDF/Steering_Ackerman.pdf, Nov 2009. [cited on p. 15]
- [55] “Control system.” <https://www.aimagv.com/control-system.html>, 2015. [cited on p. 16]
- [56] G. Corrêa and E. Colombini, “Navegação de robôs móveis em tarefas logísticas.” <https://www.prp.unicamp.br/inscricao-congresso/resumos/2019P15581A33657O5246.pdf>, Oct 2019. [cited on p. 19]
- [57] “Common agv applications.” <http://www.axter-agv.com/automated-guided-vehicles-agv/common-agv-applications/>, 2020. [cited on p. 20]
- [58] “Agv solutions.” <http://www.agvs.com.br/aplicacoes.html>, 2013. [cited on p. 21]
- [59] IKVROBOT, “These are the common applications of agv intelligent handling robot.” <https://www.ikvrobot.com/news/industry-trends/667.html>, Oct 2019. [cited on p. 21]
- [60] “The industry that makes supply chains work.” <https://www.mhi.org/casestudies?q&page1&fqindustry-group:Mobile+Automation+Group>. [cited on p. 21]

- [61] “Automated guided vehicle.” https://en.wikipedia.org/wiki/Automated_guided_vehicle, Dec 2020. [cited on p. 21]
- [62] J. Jacinto, “Controlling an agv-based amusement ride.” <https://www.totallyintegratedautomation.com/2015/12/controlling-an-agv-based-amusement-ride/>, Feb 2016. [cited on p. 21]
- [63] C. Benevides, D. Lodwig, E. Romaine, and B. Gionet, “Advantages and disadvantages of automated guided vehicles (agvs).” <https://www.conveyco.com/advantages-disadvantages-automated-guided-vehicles-agvs/>, Oct 2020. [cited on p. 22]
- [64] N. Doshi and R. Moriyama, “Application of simulation models in airport facility design,” in *Proceedings of the Winter Simulation Conference*, vol. 2, pp. 1725–1730 vol.2, 2002. [cited on p. 22]
- [65] A. Weber, “A new generation of agvs are appealing to small- and mid-size manufacturers.” <https://www.assemblymag.com/articles/92791-a-new-generation-of-agvs-are-appealing-to-small-and-midsize-manufacturers>, Apr 2015. [cited on p. 22]
- [66] “Advantages and disadvantages of automated guided vehicle.” <https://securityproductssite.wordpress.com/2018/01/10/advantages-and-disadvantages-of-automated-guided-vehicle/>, Jan 2018. [cited on p. 22]
- [67] J. Bahru, “Advantages and disadvantages of autonomous mobile robot (amr).” <https://www.dfautomation.com/advantages-and-disadvantages-of-automated-guide-vehicle-agv/>, Nov 2020. [cited on p. 22]
- [68] A. Williams, “How autonomous mobile robots are changing the logistics landscape.” <https://www.roboticsbusinessreview.com/supply-chain/autonomous-mobile-robots-changing-logistics-landscape/>, May 2018. [cited on p. 26]
- [69] C. L. Palma, “Flexsim software.” <http://www.leansimulation.org/flexsim-software>. [cited on p. 30]
- [70] “Airport simulation.” <https://www.flexsim.com/airport-simulation/>, Nov 2020. [cited on p. 30]
- [71] “Visual components 4.2 - the next generation of 3d manufacturing simulation.” <https://www.visualcomponents.com/products/visual-components/>. [cited on p. 31]
- [72] “Simio simulation software.” <https://www.simio.com/software/simulation-software.php>. [cited on p. 33]

- [73] “Simulation, production planning and scheduling software.” <https://www.simio.com/>, 2021. [cited on p. 33]
- [74] “Transportation and logistics simulation and scheduling software.” <https://www.simio.com/applications/transportation-logistics-simulation-and-scheduling-software/>, 2020. [cited on p. 34]
- [75] “Intuitive simulation software packed with powerful features.” <https://www.simul8.com/software/>, 2021. [cited on p. 34]
- [76] “Get robotic process automation right.” <https://www.simul8.com/applications/robotic-process-automation>, 2021. [cited on p. 34]
- [77] Anylogic, “Transportation simulation software.” <https://www.anylogic.com/transportation/>. [cited on p. 35]
- [78] “Warehouse simulation software.” <https://www.anylogic.com/warehouse-operations/>. [cited on p. 35]
- [79] Z. Ding, W. Gong, S. Li, and Z. Wu, “System dynamics versus agent-based modeling: A review of complexity simulation in construction waste management.” <https://www.mdpi.com/2071-1050/10/7/2484>, Jul 2018. [cited on p. 36]
- [80] A. Willmann, L. Katscher, T. Leiser, and C. Voelker, “A comparison of bottom-up and top-down modelling approaches in urban energy simulation,” in *A Comparison of Bottom-up and Top-down Modelling Approaches in Urban Energy Simulation*, 09 2019. [cited on p. 36]
- [81] S. Sumari, R. Ibrahim, N. H. Zakaria, and A. H. Ab Hamid, “Comparing three simulation model using taxonomy: System dynamic simulation, discrete event simulation and agent based simulation.” <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1033.1681>, Aug 2013. [cited on p. 36]
- [82] “Porto airport - francisco sá carneiro airport.” <https://www.introducingporto.com/porto-airport>. [cited on p. 39]
- [83] “Aeroporto francisco sá carneiro.” <http://www.visitmaia.pt/?geoarticleid=39>. [cited on p. 39]
- [84] Pedro, “Plano de expansão.” <https://aeroporto.news/plano-de-expansao/>, Jan 2016. [cited on p. 39]
- [85] “3d model store.” <https://www.cgtrader.com/>. [cited on p. 47]
- [86] “3d warehouse.” <https://3dwarehouse.sketchup.com/?hl=en>. [cited on p. 47]

[87] “3d models for free.” <https://free3d.com/>. [cited on p. 47]

[88] “Turbosquid.” <https://www.turbosquid.com/>. [cited on p. 47]

Anexo A

Schedule Source

Time	Name	Quantity	DestinoFora
50		78	1
350		76	2
650		75	3
950		68	4
1250		82	5
15000		71	6
19500		68	7
22800		74	8
23400		73	9
24000		73	1
24600		80	2
28800		79	3
30000		77	4
33600		81	6
35100		73	7
37000		75	8
40500		82	9
42300		73	1
42900		66	5
43500		73	3
44100		74	4
45600		76	2
46200		76	5
47400		74	6
48600		73	7
51600		72	8
54600		71	9
56100		76	1
57600		80	3
58800		79	4
62400		78	5
63600		77	6
64800		76	7
65400		70	8
66000		83	9
67200		68	1
68400		72	2
69300		74	3
71400		76	4
74700		77	5
77700		78	6
78900		79	7
80700		76	8

Time	Name	Quantity	Destino
18000		74	1
18900		76	6
19800		68	2
21600		83	3
22500		80	4
23400		71	7
26400		72	8
27600		73	9
28800		75	1
32400		74	4
34800		77	3
39600		78	6
40500		79	7
41400		80	8
42300		80	5
43200		68	9
44100		69	1
45000		72	2
45900		78	5
46800		79	3
48600		80	4
49500		74	5
51000		75	6
52500		72	7
54600		71	8
57000		67	9
58500		75	1
60000		74	3
61200		66	4
64200		79	5
66000		78	6
67200		77	2
68400		78	7
69300		76	8
70200		78	1
71400		78	2
72600		68	3
74700		78	4
75600		67	9
82800		78	5
83700		72	6
84600		76	7
85500		77	8

Figura A.1: Valores *Schedule Source*, a) Bagagens de Chegada, b) Bagagens de Partida

