



AVALIAÇÃO DE POLÍTICAS DE SUPRIMENTO DE ÓLEO DIESEL A
UNIDADES MARÍTIMAS DE EXPLORAÇÃO E PRODUÇÃO DE PETRÓLEO

Diogo Lobato Diuana

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Produção, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Produção.

Orientadores: Virgílio José Martins Ferreira Filho
Edilson Fernandes de Arruda

Rio de Janeiro

Maio de 2017

AVALIAÇÃO DE POLÍTICAS DE SUPRIMENTO DE ÓLEO DIESEL A
UNIDADES MARÍTIMAS DE EXPLORAÇÃO E PRODUÇÃO DE PETRÓLEO

Diogo Lobato Diuana

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO
LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA
(COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE
DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE
EM CIÊNCIAS EM ENGENHARIA DE PRODUÇÃO.

Examinada por:

Prof. Virgílio José Martins Ferreira Filho, D.Sc.

Prof. Edilson Fernandes de Arruda, D.Sc.

Prof. Glaydston Mattos Ribeiro, D.Sc.

Prof. Jean David Job Emmanuel Marie Caprace, Ph.D.

RIO DE JANEIRO, RJ - BRASIL

MAIO DE 2017

Diuana, Diogo Lobato

Avaliação de políticas de suprimento de óleo diesel a unidades marítimas de exploração e produção de petróleo / Diogo Lobato Diuana. – Rio de Janeiro: UFRJ/COPPE, 2017.

XIV, 146 p.: il.; 29,7 cm.

Orientadores: Virgílio José Martins Ferreira Filho

Edilson Fernandes de Arruda

Dissertação (mestrado) – UFRJ/ COPPE/ Programa de Engenharia de Produção, 2017.

Referências Bibliográficas: p. 111-113.

1. Simulação a Eventos Discretos. 2. Logística. 3. Indústria de Petróleo e Gás. I. Ferreira Filho, Virgílio José Martins, *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Produção. III. Título.

À minha tia, Yara Jordão Diuana (*in memoriam*), pelo apoio e inspiração indispensáveis
para alcançar meus objetivos.

Agradecimentos

Aos meus pais, Celena e Elias, pelo apoio e incentivo que sempre me deram para realizar meus estudos acadêmicos.

Aos professores Virgílio Ferreira Filho e Edilson Fernandes de Arruda pela orientação dedicada ao desenvolvimento da minha dissertação.

Aos professores Glaydston Ribeiro e Jean David Caprace por aceitarem participar da banca examinadora.

Ao corpo docente do programa de pós graduação em engenharia de produção da COPPE/UFRJ por contribuírem para minha formação acadêmica e profissional.

Aos funcionários administrativos do programa de pós graduação em engenharia de produção da COPPE/UFRJ pelo zelo em atender e resolver com prontidão as questões burocráticas.

Ao laboratório LORDE e todos os seus colaboradores pelo ótimo ambiente e infraestrutura oferecidos.

Aos colegas Gustavo Bittencourt, Rennan Chagas, Sayuri Valente e Rafael Longhi que me receberam ao laboratório e me ajudaram esclarecendo dúvidas sobre os projetos em andamento.

Aos colegas Victor Anselmo, Pedro Chaves e Ricardo Leite pelo auxílio no fornecimento dos dados necessários para conduzir a dissertação.

Ao programa de bolsas ANP_PRH21/Petrobras pelo suporte financeiro.

A todos meus familiares e amigos que me apoiaram e compreenderam meus momentos de ausência.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

AVALIAÇÃO DE POLÍTICAS DE SUPRIMENTO DE ÓLEO DIESEL A
UNIDADES MARÍTIMAS DE EXPLORAÇÃO E PRODUÇÃO DE PETRÓLEO

Diogo Lobato Diuana

Maio/2017

Orientadores: Virgílio José Martins Ferreira Filho

Edilson Fernandes de Arruda

Programa: Engenharia de Produção

As operações logísticas da área de exploração e produção *offshore* da indústria de petróleo e gás respondem por uma alta parcela do custo operacional total. Visto isso, é essencial projetar um sistema logístico eficiente, tanto para garantir a redução dos custos como para evitar riscos de desabastecimento das unidades marítimas. Por sua vez, as unidades marítimas demandam diversos tipos de suprimento para realizarem suas atividades, entre eles o óleo diesel. Este trabalho tem como objetivo comparar duas políticas logísticas de suprimento de óleo diesel para unidades marítimas, tendo como base o cenário produtivo de uma companhia petrolífera atuante no Brasil. A partir deste cenário, será então apurada qual política apresenta melhor desempenho, assim como o dimensionamento ideal da frota para cada uma. Para realizar este estudo será utilizada a ferramenta de simulação a eventos discretos.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

EVALUATION OF DIESEL SUPPLY POLICIES TO PEROLEUM
EXPLORATION E PRODUCTION MARITIME UNITS

Diogo Lobato Diuana

Maio/2017

Advisors: Virgílio José Martins Ferreira Filho

Edilson Fernandes de Arruda

Department: Production Engineering

Logistics operations in offshore petroleum exploration and production industry account for a high share of total operating cost. Since, it is crucial to design an efficient logistics system, both to ensure cost reduction and to avoid risks of shortage in offshore units. This work aims to compare two diesel supply policies to offshore units, based on the production scenario of an oil company operating in Brazil. From this scenario, it will then be determined which policy presents the best performance, as well as the optimal fleet size for each one. Discrete event simulation will be used to conduce this study.

SUMÁRIO

1	<u>INTRODUÇÃO</u>	1
1.1	Contextualização	2
1.2	Motivação e Objetivos	4
1.3	Estrutura do Trabalho	5
2	<u>REVISÃO BIBLIOGRÁFICA</u>	6
2.1	Logística <i>Offshore</i>	6
2.2	Simulação	9
2.2.1	Simulação aplicada à logística <i>offshore</i>	14
2.3	Simulação com Otimização	17
3	<u>DEFINIÇÃO DO PROBLEMA</u>	21
3.1	Descrição do Sistema Logístico	22
3.2	Simplificações e Premissas	24
3.3	Dados de Entrada do Modelo	27
3.3.1	Dados referentes aos <i>clusters</i>	27
3.3.2	Distâncias e tempos	32
3.3.3	Dados referentes aos PSVs oleiros	34
3.3.4	Custos	35
3.4	Modelagem Conceitual	38
3.4.1	Política de entregas sob demanda	38
3.4.2	Política de entregas programadas	39
3.5	Regras de Alocação de PSVs a <i>Clusters</i>	41
3.5.1	Formulação de minimização do risco do consumo do estoque de segurança	42
3.5.2	Formulação de maximização da autonomia dos <i>clusters</i>	46
4	<u>IMPLEMENTAÇÃO COMPUTACIONAL</u>	48
4.1	<i>Software</i> Utilizado	48
4.2	Estrutura de Modelagem	49
4.2.1	Locais	49
4.2.2	Entidades	50
4.2.3	Recursos	51
4.2.4	Processos	51
4.2.5	Rede de caminhos	53
4.2.6	Elementos auxiliares	54
4.3	Representação dos Elementos do Modelo	59
4.4	Lógica de Programação	61
4.4.1	Política de entregas sob demanda	61
4.4.2	Política de entregas programadas	64
4.5	Comunicação entre Simulador e Otimizador	67
4.6	Resultados de Saída do Modelo	69
4.7	Verificação e Validação	70
4.7.1	Verificações da Simulação	70
4.7.2	Validação da Simulação	73

4.7.3	Verificações da Otimização	73
5	<u>EXPERIMENTAÇÕES E RESULTADOS</u>	79
5.1	Determinação do Tempo de Aquecimento	80
5.2	Determinação do Número de Replicações	82
5.3	Resultados de Interesse.....	85
5.3.1	Nível médio de estoque.....	86
5.3.2	Nível de serviço operacional médio.....	87
5.3.3	Nível de serviço de segurança médio.....	87
5.3.4	Taxa média de utilização da frota	87
5.3.5	Lead-time médio	88
5.3.6	Custo de afretamento total	88
5.3.7	Custo operacional total da frota	88
5.3.8	Custo de estoque total	89
5.3.9	Custo de falta total	89
5.4	Experimento I	90
5.4.1	Comparação dos resultados do Experimento I	90
5.5	Experimento II.....	95
5.5.1	Custo do seguro	95
5.5.2	Custo de exposição ao risco	96
5.5.3	Comparação dos resultados do Experimento II	97
5.6	Experimento III.....	101
5.6.1	Regra FIFO <i>versus</i> minimização do risco de consumo do estoque de segurança 102	
5.6.2	Regra FIFO <i>versus</i> maximização da autonomia dos <i>clusters</i>	104
5.6.3	Comparação dos resultados entre as duas formulações	106
6	<u>CONCLUSÃO</u>	109
6.1	Trabalhos futuros	110
7	<u>REFERÊNCIAS</u>	111
	<u>APÊNDICE A – CURVAS DE ADERÊNCIA CONSUMO</u>	114
	<u>APÊNDICE B – ESTATÍSTICAS DESCRITIVAS</u>	115
	<u>APÊNDICE C – CURVAS DE ADERÊNCIA ENTREGAS</u>	116
	<u>APÊNDICE D – BLOCOS DE PROCESSOS E DIRECIONAMENTO</u>	117
	<u>APÊNDICE E – CÓDIGOS DO SIMULADOR</u>	121
	<u>APÊNDICE F – CÓDIGO FONTE DA DLL E DAS FORMULAÇÕES MIP</u>	141

LISTA DE FIGURAS

Figura 1.1 – Tipos de operações.....	3
Figura 1.2 – Logística de suprimento de óleo diesel.....	4
Figura 2.1 – Metodologia de simulação. Fonte: Chwif e Medina (2013).....	12
Figura 3.1 – Sistema logístico simplificado.....	23
Figura 3.2 – Testes de aderência.....	29
Figura 3.3 – Ranking de ajustes de funções.....	30
Figura 3.4 – Aderência da função lognormal para o <i>cluster</i> Sonda1.....	30
Figura 3.5 – fluxograma política sob demanda.....	39
Figura 3.6 – fluxograma política de entregas programadas.....	41
Figura 3.7 – Níveis de estoque acima do estoque de segurança.....	44
Figura 3.8 – Estado de dois clusters no momento do pedido.....	44
Figura 3.9 – Níveis de estoque abaixo do estoque de segurança.....	45
Figura 3.10 – Estado de dois clusters no momento do pedido.....	46
Figura 4.1 – Interface gráfica do promodel.....	48
Figura 4.2 – Processos e blocos de direcionamento.....	52
Figura 4.3 – Rede de caminhos “Rede_Mar”.....	53
Figura 4.4 – Sub rotina de atualização da taxa de consumo.....	61
Figura 4.5 – sub rotina de geração de pedidos.....	62
Figura 4.6 – Macro processo da entidade ePSV_OD. Entregas sob demanda.....	64
Figura 4.7 – Processo de geração de atendimentos.....	65
Figura 4.8 – Macro processo da entidade ePSV_OD. Entregas programadas.....	66
Figura 4.9 – Fluxograma do recurso “rPSV”.....	67
Figura 4.10 – comunicação entre simulador e otimizador.....	68
Figura 5.1 – Variação do nível de estoque da UEP7.....	81
Figura 5.2 – Variação do nível de estoque e médias móveis da UEP7.....	81
Figura 5.3 – Nível médio de estoque UEP7.....	85
Figura 5.4 – indicadores de desempenho Experimento I.....	91
Figura 5.5 – Lead-time médio.....	92
Figura 5.6 – Custos Experimento I.....	93
Figura 5.7 – Comparação custo total obtido no Experimento I.....	94
Figura 5.8 – Custo do seguro.....	96

Figura 5.9 – Custo de exposição ao risco	97
Figura 5.10 – Indicadores de desempenho do Experimento II.....	98
Figura 5.11 – Custos no Experimento II.....	99
Figura 5.12 – Comparação custo total para o Experimento II.....	99
Figura 5.13 – comparação indicadores: Regra 1 do Experimento III.....	102
Figura 5.14 – Comparação dos custos: Regra 1 do Experimento III.....	103
Figura 5.15 – comparação custo total: Regra1 do Experimento III.....	103
Figura 5.16 – comparação indicadores: Regra 2 do Experimento III.....	104
Figura 5.17 – comparação custos: Regra 2 do Experimento III	105
Figura 5.18 – comparação custo total regra2 exp.III.....	105
Figura 5.19 – comparação indicadores: Regra 1 e Regra 2 do Experimento III	106
Figura 5.20 – comparação custos: Regra 1 e Regra 2 do Experimento III.....	107
Figura 5.21 – comparação custo total regra 1 e regra 2 exp. III.....	107

LISTA DE TABELAS

Tabela 2.1 – Conceitos subjacentes à simulação segundo Banks (1999).....	10
Tabela 3.1 – composição dos <i>clusters</i>	23
Tabela 3.2 – Informações dos <i>clusters</i>	28
Tabela 3.3 – Distribuição dos consumos diários	31
Tabela 3.4 – Distribuição dos volumes entregues	32
Tabela 3.5 – Matriz de distâncias em Km	33
Tabela 3.6 – Velocidades por trecho	34
Tabela 3.7 – Matriz de tempos de navegação em horas	34
Tabela 3.8 – Troca de tripulação e inoperância.....	35
Tabela 3.9 – Custos	36
Tabela 3.10 – Valores base para cálculo de custo de falta	36
Tabela 3.11 – Cronogramas de atendimento	40
Tabela 4.1 – Descrição dos locais	49
Tabela 4.2 – Descrição do locais “tanque”	50
Tabela 4.3 – Descrição das entidades	51
Tabela 4.4 – Atributos	54
Tabela 4.5 – Variáveis	55
Tabela 4.6 – Matrizes	55
Tabela 4.7 – Sub rotinas	58
Tabela 4.8 – Macros	59
Tabela 4.9 – Modelagem elementos físicos	60
Tabela 4.10 – Resultados do modelo.....	69
Tabela 4.11 – Verificação dados de consumo	71
Tabela 4.12 – Verificação dados de volume de pedido	72
Tabela 4.13 – Dados dos PSVs: verificação da formulação 1	74
Tabela 4.14 – Dados dos clusters: verificação da formulação 1	74
Tabela 4.15 – Tempos de navegação em horas: verificação da formulação 1	74
Tabela 4.16 – Alocações: verificação da formulação 1	75
Tabela 4.17 – Demanda não atendida: verificação da formulação 1	75
Tabela 4.18 – Dados dos PSVs: verificação formulação 2.....	76
Tabela 4.19 – Dados dos clusters: verificação formulação 2	76

Tabela 4.20 – Tempos de navegação:verificação da formulação 2.....	76
Tabela 4.21 – Autonomia e demanda não atendida: verificação formulação 2.....	77
Tabela 4.22 – Alocação: verificação da formulação 2	77
Tabela 4.23 – Demandas não atendidas: verificação da formulação 2.....	78
Tabela 5.1 – Experimentos	79
Tabela 5.2 – Coeficiente de variação para os dados de consumo.....	80
Tabela 5.3 – Nível médio de estoque UEP7	82
Tabela 5.4 – Resultados política sob demanda: Experimento I.....	94
Tabela 5.5 – Resultados política de entregas programadas: Experimento I.....	95
Tabela 5.6 – Resultados política entregas sob demanda: Experimento II.....	100
Tabela 5.7 – Resultados política entregas programadas: Experimento II	101
Tabela 31 – Processos e direcionamento de entidades	117

LISTA DE SIGLAS

E&P	Exploração e Produção
FIFO	<i>First In First Out</i>
ID	Número de identificação
MIP	Mixed Integer Programming
NT	Navio Tanque
OD	Óleo Diesel
PSV	<i>Platform Supply Vessel</i>
TQ	Tanque
UEP	Unidade Estacionária de Produção
UM	Unidade Marítima

1 Introdução

As atividades de exploração e produção (E&P) na indústria de petróleo e gás requerem uma infra estrutura logística eficiente para manterem suas operações sem paradas de produção. O sistema logístico da fase *upstream* da indústria se caracteriza pelos altos custos nas operações e o alto valor da produção. Portanto, ao mesmo tempo em que se deseja diminuir custos logísticos, as atividades não podem ser interrompidas devido a atrasos de entrega de cargas.

Problemas logísticos estão gradualmente se tornando mais complexos e um melhor entendimento do gerenciamento logístico como disciplina é a chave para lidar com os novos desafios (AAS, 2008). Um elemento central do gerenciamento logístico é o planejamento logístico. Segundo Aas (2008), o planejamento logístico mitiga efeitos indesejados como nível de serviço baixo ao cliente, desperdícios e a manutenção de estoque muito alto para execução das atividades logísticas. Planejamento logístico é a atividade que busca uma maneira inteligente de coordenar e orientar o uso dos recursos logísticos. Dois ingredientes principais para o planejamento logístico são a disponibilidade de informações logísticas e a capacidade de solução de problemas.

Por sua vez, as operações de E&P são apoiadas por um sistema logístico cujas principais atividades são o transporte de carga para as unidades marítimas assim como de carga de retorno, o transporte de pessoas e o armazenamento de materiais e equipamentos em terra e nas unidades marítimas (LEITE, 2012).

O presente estudo analisa o cenário logístico *offshore* de uma companhia petrolífera brasileira. A ênfase é dada à logística de suprimento de óleo diesel. Dessa forma, é realizado um estudo que utiliza a técnica de simulação a eventos discretos para experimentar duas políticas de suprimento e determinar qual a mais adequada para a companhia, através da análise dos resultados de indicadores de desempenho e dos custos logísticos. O estudo também aponta o dimensionamento ideal da frota de embarcações do tipo *Platform Supply Vessel* (PSV) destinadas ao transporte de óleo diesel, denominadas PSVs oleiros, para o cenário produtivo da empresa.

1.1 Contextualização

O cenário *offshore* de E&P da companhia em estudo é composto por unidades estacionárias de produção (UEPs), sondas de perfuração e unidades de manutenção e segurança, que funcionam como hotéis flutuantes em alto mar. A infra estrutura de apoio logístico conta com portos, aeroportos, armazéns em terra e centros de distribuição em mar, como os navios tanque. Por sua vez, as UMs demandam serviços de cargas e descargas de materiais e insumos necessários para a realização de suas atividades. Dessa forma, a companhia contrata diversos tipos de embarcações especializadas para realizar o transporte marítimo. Os produtos demandados pelas UMs variam de acordo com o tipo e as atividades destas. Além disso, dependendo do estágio do ciclo de vida do campo explorado, há padrões diferentes de consumos. Por exemplo, na fase de perfuração de poços, há grande demanda por produtos químicos, cimento, tubos e *risers*.

As cargas típicas podem ser classificadas como carga de convés, água, óleo diesel, fluidos e granéis (LEITE, 2012). Carga de convés são produtos transportados nos conveses dos PSVs que podem ou não estar unitizados em *containers*. Tais produtos podem ser subdivididos em carga geral, comida e água potável, tubos, *risers*, produtos químicos e lixo. O insumo água se refere à água industrial transportada em tanques dos PSVs. O óleo diesel também é transportado em tanques destinados para o produto. Fluidos compreendem salmoura e lama de perfuração e esses insumos também são transportados em tanques. Por fim, granéis compreendem cimento, baritina e bentonita, predominantemente, e também são transportados em tanques apropriados dos PSVs. Segundo Leite (2012), a composição da frota de PSVs da companhia é especializada por tipo de carga transportada. Dessa forma, os PSVs são classificados em PSVs oleeiros, PSVs fluideiros, PSVs graneleiros, PSVs cimenteiros e PSVs de carga geral e água, embora todos sejam capazes de transportar quaisquer tipos de carga. Esta é uma característica operacional peculiar à companhia estudada, visto que as operações logísticas realizadas por outras companhias geralmente utilizam-se de uma frota de PSVs multi-carga.

A logística de apoio *offshore* vem se tornando uma nova área de conhecimento no segmento de Petróleo e Gás (ILOS, 2013). O custo dessa operação é bastante elevado dado que a cadeia logística compreende a armazenagem e transporte das cargas, a

operação portuária e o transporte marítimo até as UMs, além das atividades como transporte de passageiros e serviços como movimentação de unidades marítimas.

Segundo Ilos (2013), a logística de apoio *offshore* pode ser dividida em três grandes tipos de operações: logística de cargas, logística de passageiros e serviços. A logística de cargas é responsável por movimentar todo tipo de carga necessária para a operação de perfuração e produção das UMs. A logística de passageiros trata da movimentação de pessoas entre o continente e as UMs. Essa operação é realizada por helicópteros. Por fim, os serviços compreendem atividades de apoio em alto mar como movimentação de unidades marítimas, manuseio de espias e combates a eventuais incêndios ou vazamentos, por exemplo. Os três tipos de operações são resumidos na Figura 1.1.

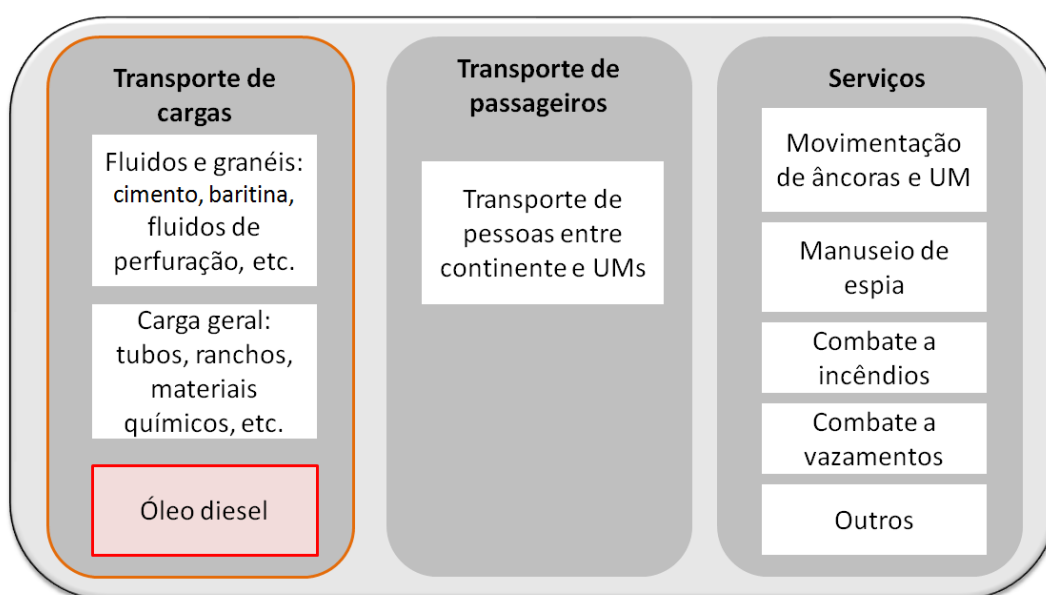


Figura 1.1 – Tipos de operações

Fonte: Adaptado da Web Page ILOS

Neste estudo, aborda-se a logística do transporte de cargas, em específico de óleo diesel, como destacado na Figura 1.1. A demanda por óleo diesel pode ser atendida por embarcações de apoio, que são abastecidas em um *hub* marítimo, isto é, um navio tanque. Em casos em que não há navios tanque em alto mar, os PSVs oleiros precisam ir ao terminal terrestre para abastecerem e atender às UMs.

Considerando um sistema onde há navios tanque, a logística de suprimento de óleo diesel pode ser resumida na Figura 1.2. Nesse sistema, o navio tanque abastece em algum terminal terrestre, navega para uma região próxima às UMs onde permanece abastecendo os PSVs oleiros que, por sua vez, atendem às UMs.

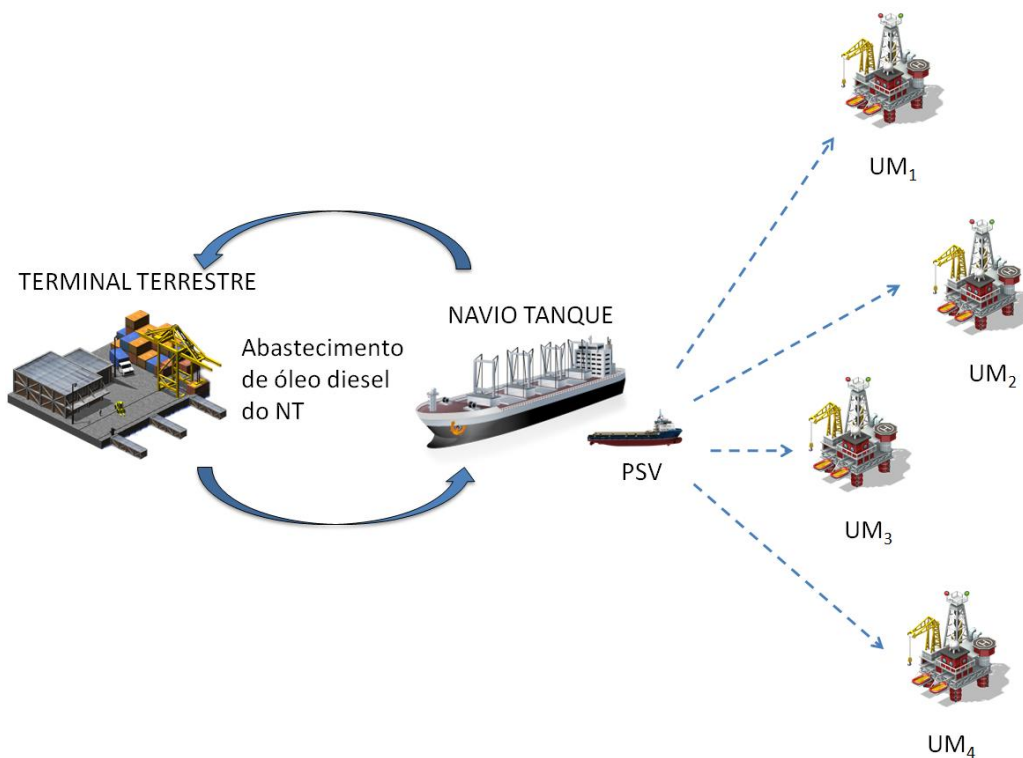


Figura 1.2 – Logística de suprimento de óleo diesel

1.2 Motivação e Objetivos

A importância do sistema logístico para operações de E&P *offshore* pode ser explicada devido ao alto custo de parada, seja das unidades de produção ou sondas de perfuração e também aos altos custos operacionais. Logo, é tão importante não deixar faltar suprimentos quanto reduzir os custos logísticos. Dessa forma, se faz importante um estudo de pesquisa operacional a fim de determinar o dimensionamento ideal da frota de navios supridores.

Um dos insumos mais importantes para a manutenção da produção das unidades marítimas é o óleo diesel. Este insumo é utilizado nas UMs para geração de energia elétrica, navegação e estabilização de posição no caso das sondas de perfuração, limpeza de dutos entre outras finalidades. Um eventual desabastecimento de óleo diesel implica na parada de produção de uma UEP ou na parada das operações de uma sonda de perfuração. O custo do afretamento diário de sondas de perfuração é muito elevado e, portanto, deseja-se evitar ao máximo atrasos nas operações devido à falta de óleo diesel. Além disso, o desabastecimento de uma UEP gera muito prejuízo quando se observa que o sistema de produção é contínuo e cada hora parada significa perdas de receita.

De acordo com Leite (2012), o sistema logístico de suprimento de óleo diesel da companhia em estudo é composto por dois terminais de abastecimento em terra onde os navios tanque reabastecem seus estoques e posteriormente navegam para uma região em alto mar próxima às unidades marítimas onde permanecem ancorados fornecendo óleo diesel para os PSVs oleiros. Por sua vez os PSVs oleiros abastecem as UMs da região.

Tendo em vista a importância do suprimento adequado de óleo diesel para as unidades marítimas, o objetivo do estudo é modelar e avaliar duas políticas logísticas de suprimento de óleo diesel, variando cenários de tamanho da frota usando a técnica de simulação a eventos discretos. Com isso, deseja-se determinar qual política é a mais adequada e qual o dimensionamento ideal da frota de PSVs oleiros, através da análise de indicadores de desempenho como nível de serviço, *lead-time*, taxa de ocupação da frota assim como os custos inerentes a cada um dos cenários avaliados para cada política.

1.3 Estrutura do Trabalho

O trabalho está estruturado em seis capítulos. O Capítulo 1 compreende a introdução ao tema, a motivação e os objetivos do trabalho. O Capítulo 2 apresenta a revisão bibliográfica relevante ao desenvolvimento do trabalho. O Capítulo 3 define o problema abordado e descreve as principais características do sistema logístico estudado. No Capítulo 4 é apresentada a implementação computacional do modelo de simulação desenvolvido. Em seguida, no Capítulo 5, é realizada a descrição técnica dos experimentos e seus resultados. Por fim, o Capítulo 6 discorre sobre as principais conclusões do trabalho e alguns apontamentos para trabalhos futuros.

2 Revisão Bibliográfica

A revisão bibliográfica do trabalho foi desenvolvida a partir de uma busca na literatura acadêmica sobre os temas de logística *offshore*, simulação, simulação aplicado na área de logística *offshore* e simulação com otimização. Esses são os temas abordados no trabalho, uma vez que o mesmo utiliza a técnica de simulação a eventos discretos para representar o sistema logístico *offshore* de um companhia petrolífera, combinando formulações matemáticas para otimizar o processo de alocação de embarcações supridoras aos *clusters* que é executado durante a simulação. Visto isso, também buscou-se na literatura o embasamento teórico sobre a combinação das técnicas de otimização e simulação.

2.1 Logística *Offshore*

As atividades de apoio responsáveis por transportar bens e pessoas entre a terra e as unidades marítimas de exploração e produção de petróleo são comumente denominadas por “logística *offshore*”. Outras expressões de uso comum na indústria de petróleo e gás são aquelas que classificam os setores de processamento desde a extração da *commodity* até a distribuição de seus produtos finais. Assim, o setor de exploração e produção é denominado “*upstream*”, o setor de transformação do óleo cru em subprodutos comercializáveis é denominado “*midstream*” e o setor de distribuição dos produtos finais é denominado “*downstream*”(FERREIRA FILHO, 2016). As atividades logísticas do setor “*upstream*” compreendem a movimentação de pessoal e suprimentos como equipamentos, combustível, água e comida entre os portos e as unidades marítimas e entre as unidades marítimas. Alguns estudos na literatura acadêmica abordam a questão de dimensionamento de frota e programação de rotas que utilizam técnicas de pesquisa operacional aplicadas na área. Outros destacam problemas de roteamento de veículos e programação de cronogramas das rotas. Ao decorrer do capítulo serão descritos os principais estudos e teorias que embasam esta dissertação.

De acordo com Kaiser (2010), o estudo logístico do setor *upstream* da indústria de óleo e gás tem se diversificado, porém ainda não foi bem desenvolvido e unificado teoricamente. O autor realizou um estudo no qual desenvolveu uma estrutura

metodológica para prever a intensidade das atividades de apoio logístico na região do Golfo do México com a finalidade de dimensionar a capacidade dos portos da região e a indústria de construção naval. Kaiser e Snyder (2010) também realizaram um estudo sobre o dimensionamento de frotas a partir da análise de dados de demanda por atividade realizada no setor *upstream* no Golfo do México, que compreendem a exploração, o desenvolvimento da produção, a produção e o abandono – ou desmobilização – que ocorre no final da vida útil de um campo de petróleo. Dessa forma, em cada estágio do ciclo de E&P existe uma demanda por algum tipo de embarcação e o tamanho da frota é dimensionado através da análise empírica dos dados relativos a esses estágios.

Grande parte dos estudos realizados acerca do tema possui origem nórdica, principalmente norueguesa. Tal fato é explicado pela exploração e produção de petróleo no Mar do Norte e no Mar de Barents, situados respectivamente na costa oeste e norte do país que se intensificou a partir da década de 1970. Halvorsen-Weare *et al.*(2012) fizeram um estudo no qual determinam a composição ótima da frota e a periodicidade de viagens para um conjunto de unidades marítimas de uma companhia petrolífera norueguesa. Neste estudo, os autores desenvolveram um método de planejamento logístico que utiliza uma formulação matemática de um problema clássico de pesquisa operacional – problema de roteamento de veículos periódico – no qual consideram as demandas semanais das unidades marítimas e as distâncias entre elas e o porto de onde partem os navios supridores. O problema visa minimizar os custos logísticos, de afretamento e navegação, e, ao mesmo tempo, manter um nível de serviço adequado. Para isso, utilizam o algoritmo de programação matemática para determinar o tamanho ideal da frota e os cronogramas semanais de atendimento dos navios da frota. Como resultado, o estudo conseguiu reduzir o tamanho da frota de 5 para 4 embarcações, mantendo um nível de serviço confiável às unidades marítimas. Fagerholt e Lindstad (2000) analisaram o modo operacional da logística *offshore* da mesma empresa norueguesa e propuseram 6 cenários alternativos de operação onde variam as janelas de tempo de operação nas unidades marítimas e as frequências de atendimento. A companhia estudada adotava uma política de janelas de tempo no qual não permitia atendimentos noturnos em algumas unidades marítimas. Os autores utilizaram a formulação do problema do caixeiro viajante para cada instância do problema e então determinaram os tipos mais adequados de embarcações, o tamanho ideal da frota, assim como o cronograma ótimo de viagens para as embarcações que atendem as unidades

marítimas em cada cenário avaliado. O objetivo do estudo também é minimizar o custo operacional total do sistema logístico. Halvorsen-Weare *et al.* (2013) desenvolveram um estudo sobre o transporte de gás natural liquefeito no qual utilizam técnicas de otimização considerando também parâmetros de incerteza como tempos de navegação e taxas de produção diária nas unidades marítimas para determinar o roteamento e programação dos navio transportadores de gás natural liquefeito entre as unidades marítimas e os portos de descarregamento do produto. AAS *et al.* (2007) abordam o problema de roteamento de embarcações a unidades marítimas considerando os fluxos de carga e descarga e o efeito da capacidade limitada de estoque também no cenário produtivo da costa norueguesa. Os autores usam a formulação de programação inteira mista para propor o roteamento ótimo para o cenário, levando em consideração parâmetros de demandas, distâncias, capacidades de armazenagem das unidades marítimas, janelas de tempo e custos.

Uglane e Friedberg (2012) propõem um modelo matemático para o problema de sequenciamento e programação de rotas para as operações de uma empresa petrolífera brasileira. Os autores também consideram no modelo o posicionamento dos navios tanque presentes na bacia.

Ribeiro e Iachan (2009) realizaram um trabalho de dimensionamento de frota de embarcações supridoras das plataformas de petróleo da Petrobras. Para conduzir o estudo, os autores modelaram as entregas das cargas por simulação e utilizaram um modelo de programação inteira para alocar rotas às embarcações. Os autores classificaram as demandas das unidades marítimas em três categorias de carga: carga de convés, carga de água e carga de diesel. Além disso, assumiram uma frota composta por embarcações capazes de transportar os três tipos de carga e embarcações especializadas para transporte de óleo diesel. O modelo de otimização foi formulado como um problema de roteamento de veículos com várias viagens. Como função objetivo os autores escolheram minimizar a distância percorrida assim como o número de navios supridores. Além da formulação do problema de otimização, os autores levaram em consideração as incertezas inerentes a alguns parâmetros utilizando simulação a eventos discretos. Os parâmetros de incerteza compreendem o tipo de carga solicitada pelas plataformas, a plataforma que requisitou o pedido, o momento em que a solicitação é gerada, o prazo máximo para atendimento da solicitação e a quantidade demandada da carga. Assim, utilizou-se uma heurística de roteamento e seleção de embarcações para

alocar as ordens e seus parâmetros gerados pelo simulador nas rotas. Por fim, o otimizador realiza as alocações ótimas das embarcações às rotas geradas na simulação.

2.2 Simulação

A simulação é uma importante ferramenta da pesquisa operacional que permite analisar sistemas estocásticos e tomar decisões estratégicas sobre aspectos quantitativos que se deseja dimensionar. Esta ferramenta é muito empregada em modelagem de sistemas que apresentam fatores de incerteza. Através da modelagem do sistema que se deseja estudar e a implementação computacional, é possível simular as operações durante um período de tempo e então verificar os resultados de interesse.

De acordo com Hillier e Lieberman (2008), a simulação é frequentemente utilizada para realizar análise de risco em processos financeiros, imitando e repetindo a evolução das transações envolvidas para gerar um perfil de possíveis resultados. Segundo os autores, a simulação também pode ser usada para analisar sistemas estocásticos que continuarão a operar indefinidamente. Hillier e Lieberman (2008) afirmam que há duas categorias principais de simulação: a simulação por eventos discretos e a simulação contínua. A simulação por eventos discretos é aquela em que as mudanças de estado do sistema ocorrem instantaneamente em pontos aleatórios no tempo como ocorrência de eventos discretos. A simulação contínua é aquela na qual as mudanças no estado do sistema ocorrem continuamente ao longo do tempo. As simulações contínuas exigem o emprego de equações diferenciais para descrever a taxa de mudança das variáveis de estado. Porém, muitas vezes é possível aproximar o comportamento de um sistema contínuo por um comportamento discreto e usar simulação a eventos discretos.

Segundo Banks (1999), simulação é a imitação das operações de um sistema do mundo real ao longo do tempo. De acordo com o autor, a simulação envolve a geração de uma história artificial do sistema e a observação dessa história para extrair inferências em relação ao sistema real que é representado. Dessa forma, é possível analisar o comportamento de um sistema fazendo suposições do tipo “e se” sobre o mesmo, isto é, construindo cenários hipotéticos. Banks (1999) enumera uma lista de conceitos subjacentes à simulação. Essa lista foi resumida na Tabela 2.1.

Tabela 2.1 – Conceitos subjacentes à simulação segundo Banks (1999)

Conceito	Significado
Modelo	É a representação de um sistema real.
Evento	É uma ocorrência que muda o estado do sistema.
Variáveis de estado do sistema	É a coleção de todas as informações necessárias para definir o que está acontecendo dentro do sistema em um dado momento do tempo.
Entidades	Uma entidade representa um objeto que requer definição explícita. Pode ser dinâmica, se move através do sistema ou estática, se serve outras entidades.
Atributos	São variáveis locais que armazenam valores relacionados às entidades.
Recursos	São elementos que fornecem serviço de transporte das entidades.
Lista de processos	São usadas para representar filas. As listas de processos são executadas geralmente de acordo com a regra FIFO (<i>First-in-first-out</i>).
Atividades e esperas	Uma atividade é uma duração de tempo conhecida <i>a priori</i> . Essa duração pode ser constante ou uma variável aleatória. Uma espera é uma duração indefinida de tempo causada por uma combinação de condições do sistema.
Modelo de simulação a eventos discretos	É o modelo de simulação no qual as variáveis de estado mudam de valores em momentos discretos de tempo.

Carson II (2005) afirma que a simulação é uma poderosa ferramenta de análise para projetar novos sistemas, reformar sistemas vigentes e até mesmo propor mudanças em regras operacionais. O autor define os principais conceitos da simulação por eventos discretos. Além disso, o autor enumera as diretrizes necessárias para construção de um projeto de simulação. Essas diretrizes são apresentadas a seguir:

1º Iniciação do Projeto

Consiste na primeira fase do projeto no qual o problema é definido e os objetivos são expostos. Nessa fase devem ser definidas as medidas de desempenho que se deseja analisar, o escopo do problema e o nível de detalhamento. Além disso devem ser desenhado o modelo conceitual e definida a linguagem de programação que será usada.

2º Execução do Projeto

Consiste no desenvolvimento do modelo e na coleta de dados. Nesta etapa o cliente ou interessado no projeto deve fornecer os dados necessários para a condução do projeto. A partir deste ponto os dados devem ser analisados e tratados pelos especialistas. Então o modelo conceitual deve ser traduzido em linguagem computacional.

3º Verificação e Validação

A verificação do modelo consiste em analisar se os resultados estão coerentes com a programação. Esta etapa consiste em depurar o código implementado. Algumas

técnicas como variar parâmetros de entrada para valores extremos ou usar valores médios dos parâmetros, conferência de outras variáveis de saída do modelo, utilizar as ferramentas de animação gráfica e de *debug* podem ser usadas nessa fase.

A validação do modelo deve ser realizada junto ao cliente e consiste em analisar se o modelo está representando de forma fidedigna a realidade.

4º Experimentação

Após o modelo ter sido verificado e validado, está pronto para realizar experimentos. Para isso, deve-se definir quais parâmetros serão variados no modelo, qual período de tempo que se deseja simular, qual tempo necessário de aquecimento e quantas replicações são necessárias para garantir a confiabilidade dos resultados.

5º Análise

A análise é baseada nas medidas de desempenho definidas na fase inicial do projeto.

6º Relatório

Os relatórios dos resultados dos experimentos e as análises devem ser mostrados tanto em apresentação como em documentos escritos.

Chwif e Medina (2013) resumem as principais etapas para construção de um modelo de simulação. Segundo os autores, essas etapas compreendem a metodologia seguida para se desenvolver tal modelo e estão representadas na Figura 2.1. Os autores dividem o desenvolvimento de um modelo de simulação em três etapas principais: a concepção do problema no qual o analista deve entender o sistema e traçar os objetivos que se deseja alcançar, desenhando um modelo conceitual; a implementação do modelo, etapa em que se traduz o modelo conceitual em modelo computacional (nessa fase cabe a verificação e a validação do modelo); e a etapa de análise, que consiste em experimentar o modelo e avaliar os resultados.

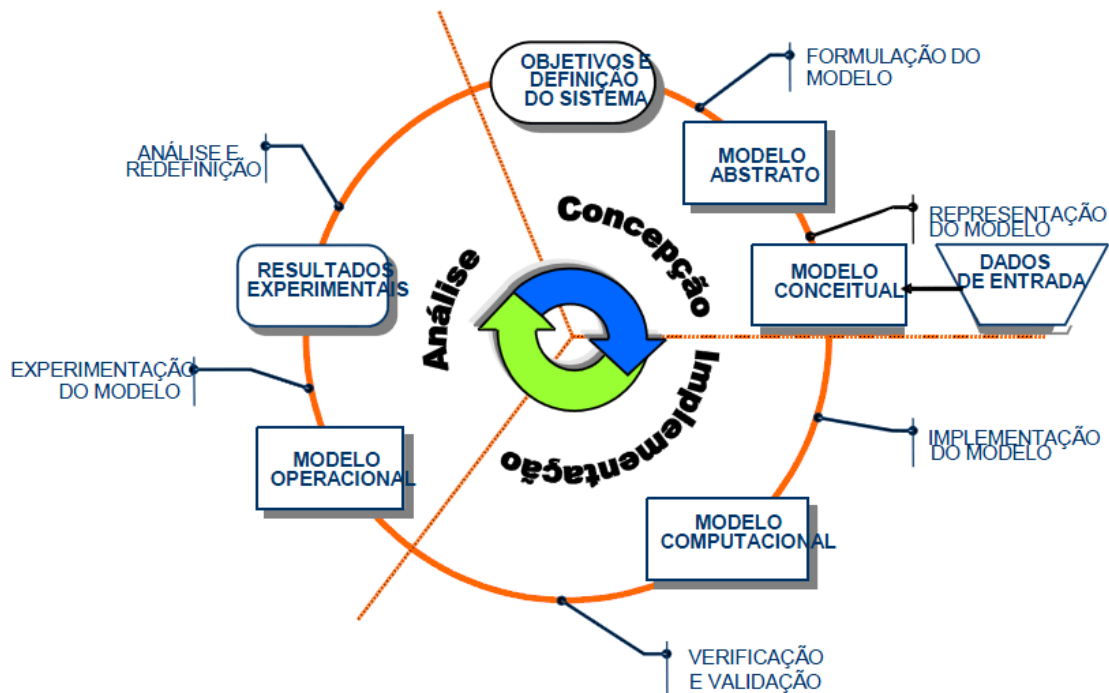


Figura 2.1 – Metodologia de simulação. Fonte: Chwif e Medina (2013)

As fases de verificação e validação são melhores descritas por Sargent (2005). Segundo o autor, modelos de simulação são usados para vários fins como planejamento de sistemas e desenvolvimentos de políticas operacionais. Os usuários desses modelos baseiam-se nos resultados dos mesmos e preocupam-se se o modelo e seus resultados estão “corretos” para serem aplicados na prática. Essa preocupação é abordada durante a fase de verificação e validação. A verificação consiste em certificar-se que a programação do modelo computacional está correta. A validação é definida como a fundamentação de que o modelo desenvolvido dentro do domínio de sua aplicação possui um satisfatório nível de acurácia. Dessa forma, o autor destaca a importância de relacionar o modelo com o mundo real de forma a garantir a credibilidade dos resultados do modelo. O autor descreve as curvas de relação entre o custo, do ponto de vista do desenvolvedor do modelo, o valor, do ponto de vista do usuário do modelo, e a credibilidade do modelo. Em uma escala de confiança de 1 a 100%, a curva de custo aumenta de forma crescente à medida que se busca confiança de 100% do modelo. Já a curva de valor aumenta com comportamento decrescente à medida que se aproxima de 100% de confiança. Ainda de acordo com Sargent (2005), existem três abordagens possíveis para conduzir a validação do modelo. A primeira abordagem é a mais comum e nela o processo de validação é conduzido pela própria equipe desenvolvedora do

modelo. Na segunda abordagem, a validação é conduzida pela equipe de usuários do modelo. A terceira abordagem, também chamada de verificação e validação independente, usa uma terceira parte, independente da equipe de desenvolvedora e dos usuários para determinar se o modelo é válido. Além disso, o autor apresenta técnicas de validação que vão desde usar o recurso de animação até testes de degeneração, por exemplo.

Tendo em vista o embasamento teórico da técnica de simulação, vale citar alguns estudos acadêmicos desenvolvidos a partir desta ferramenta. Procurou-se na literatura trabalhos que utilizassem simulação para diversos fins, posteriormente foi realizado um filtro para aplicações de simulação na área logística.

Pinto (2007) desenvolve um modelo de simulação para analisar diferentes políticas de gestão hospitalar. No estudo, pode-se analisar como diferentes políticas de gestão de estoque de medicamentos e capacidades de mão-de-obra (médicos) influem no custo do atendimento no hospital estudado (hospital de clínicas da Unicamp). A autora analisa duas políticas de estoque (puxada e empurrada) e duas políticas de capacidade (puxada e empurrada). A política de estoque puxada se baseia na reação à demanda em tempo real e a política empurrada planeja o estoque com base na previsão da demanda. Da mesma forma a política de capacidade de mão-de-obra pode ser efetivada com base na demanda real (puxada), com auxílio de horas extra ou a partir da antecipação da demanda (empurrada), caracterizada por um quadro de mão-de-obra fixo. Foram simulados quatro cenários (combinação das duas políticas) com base nos parâmetros de custos e os dados de entrada foram a taxa de chegada de pacientes, a taxa de atendimento dos pacientes, a demanda média por medicamentos e parâmetros como custos de estoque, custo da mão-de-obra. Os resultados obtidos permitiram apontar qual política apresenta menores custos e maiores níveis de serviço.

Silva *et al.*(2011) abordam a aplicação da simulação a eventos discretos junto a um método de decisão multi critério para definir o dimensionamento de um sistema de transporte marítimo de circuito fechado de minério de ferro para siderúrgicas, sob o cenário de uma mineradora atuante nas regiões nordeste e sudeste do Brasil. Para isso os autores modelam o sistema marítimo e simulam várias configurações variando o tamanho da frota e o volume do estoque máximo no armazém único da mineradora de forma que os recursos navios e armazém devem ser capazes de suprir a demanda da siderúrgica sem admitir paradas de produção devido à falta de minério. Os parâmetros avaliados são o custo operacional da frota e do armazém e o custo de falta da

siderúrgica. Após simular os cenários, o modelo de decisão multicritério analisa qual melhor alternativa.

Silva (2012) aplica a técnica de simulação para estudar *trade-off* em um sistema de transporte aquaviário de ciclo fechado. Dessa forma, o autor propõe dimensionar a frota de embarcações e os estoques do sistema aquaviário de transporte de minério de ferro avaliando o *trade-off* entre capacidade transportada (tamanho da frota) e a capacidade de estocagem. O sistema de transporte consiste em dois portos, um localizado na região sudeste e outro na região nordeste do Brasil (locais onde o minério de ferro é produzido) e um porto próximo à usina siderúrgica, onde o minério de ferro é entregue. O estudo visa dimensionar o estoque de minério de ferro que abastece a usina e a frota para transportar o insumo. Assim, o estudo considera aleatoriedades nos consumo de minério de ferro pela usina e nos tempos de navegação, carregamento e descarregamento das embarcações. Através de variação nos cenários como tamanho e composição da frota e capacidade máxima do estoque da usina, o estudo permite avaliar a melhor operação logística de forma a minimizar custos e ao mesmo tempo manter a usina sem parada de produção.

Ribeiro (2003) apresenta um modelo de simulação para avaliar a capacidade do Aeroporto de Congonhas, localizado em São Paulo e apontar seus gargalos operacionais tendo em vista diversos níveis de demanda de vôos. O autor modela os diversos sub sistemas do pátio de aeronaves do aeroporto e utiliza o *software* Arena para sua implementação. Para isso, o autor cria informações de vôos simuladas a partir dos dados históricos de pousos e decolagens e demais parâmetros como tempo de permanência em solo, tempos de embarque e desembarque de bagagens, tempo de taxiamento entre e outros. Como resultados, o modelo permite dimensionar os recursos necessários para as atividades aeroportuárias, a influência de cada fator operacional no tempo de permanência em solo das aeronaves assim como permite extrair as estatísticas de interesse.

2.2.1 Simulação aplicada à logística *offshore*

As operações logísticas *offshore* contam com muitos fatores de incerteza como condições climáticas, demandas das unidades marítimas, falhas de embarcações e

tempos operacionais e de navegação. Visto isso, alguns estudos utilizam a técnica de simulação para resolver problemas da área que envolvem fenômenos estocásticos.

Shyshou, Gribkovskaia e Barceló (2010) realizaram um estudo de dimensionamento da frota de navio do tipo AHTS (*Anchor Handling Tug Supply Vessel*) que são embarcações especializadas em operações de manuseio de âncoras. Esses navios são utilizados para auxiliar na movimentação de unidades marítimas, comumente sondas de perfuração. O custo de afretamento desses navios está entre os mais caros, e o valor varia de acordo com o horizonte de contratação. Contratos no mercado à vista são mais caros que contratos de longo prazo. O objetivo do estudo foi minimizar os custos de contratação desses navios com base na previsão das atividades futuras da companhia estudada. Para realizar o estudo os autores utilizaram um modelo de simulação no qual consideram a estocasticidade das durações das operações de movimentação das sondas e a variação da taxa de afretamento de curto prazo no cenário futuro. A partir da programação anual de movimentação de sondas da companhia, foram realizadas simulações que refletem o efeito das condições climáticas nos tempos operacionais dos navios AHTS assim como o efeito da variação aleatória das taxas de afretamento desses navios no futuro. Em determinados momentos, o número de navios AHTS contratado no mercado de longo prazo não são suficientes para atender o planejamento de movimentação das sondas. Assim a simulação visa determinar qual o tamanho ideal da frota contratada de longo prazo, descrevendo os fenômenos estocásticos que influenciam os tempos operacionais. Dessa forma, foram realizados alguns cenários variando o número de embarcações AHTS de contratos de longo prazo e então foi verificado qual cenário apresentou o menor custo.

Maisiuk e Gribkovskaia (2014) abordaram o problema de dimensionamento da frota de navios supridores *offshore* considerando a estocasticidade dos tempos de navegação e operação nos atendimentos às unidades marítimas de uma companhia de petróleo norueguesa. No estudo, os autores empregaram a técnica de simulação a eventos discretos para experimentar algumas configurações de tamanho de frota, considerando incertezas de condições climáticas – que afetam diretamente nos tempos de navegação e operação nas unidades marítimas – assim como das taxas futuras de contrato à vista dos navios supridores. Os autores usaram o *software* Arena. O simulador reproduz os cronogramas de viagens para os navios supridores, analisando as saídas dos tempos de navegação e operação e então determina o tamanho ideal da frota de navios que operam por contratos de longo prazo.

Ulstein (2014) faz um estudo do impacto das condições climáticas do Mar de Barents nas operações logísticas *offshore* da região. O autor conduz um estudo de simulação a eventos discretos considerando o impacto das condições climáticas como altura das ondas, ventos e congelamento do mar nos tempos operacionais dos PSVs nas unidades marítimas. Dessa forma, os fenômenos climáticos são descritos através de distribuições de probabilidade e são *inputs* do simulador. Os *outputs* do simulador são os tempos operacionais dos PSVs e o consumo de combustível. Além disso, o autor implementa um modelo de otimização formulado pela metaheurística de algoritmo genético para determinar a composição ótima da frota, tendo como função objetivo a minimização dos custos operacionais. Para isso, o otimizador lê os resultados da simulação para então, em um processo iterativo, executar o algoritmo genético variando a frota de PSVs.

Leite (2012) realizou um estudo de simulação para transporte de carga de convés a unidades marítimas de uma companhia petrolífera brasileira. Nesse estudo são testados diversos cenários de atendimentos, nos quais se varia a frequência de atendimentos e as quantidades máximas de unidades por *cluster*. A simulação considera a estocasticidade da demanda das unidades marítimas, assim como das durações das viagens, considerando os tempos de navegação, operação no porto e nas unidades marítimas.

Batista (2005) realiza uma análise operacional de embarcações de apoio marítimo às plataformas da Bacia de Campos da empresa Petróleo Brasileiro S.A (PETROBRAS), através de um modelo de simulação. No estudo, o autor modela o processo de operações das embarcações no porto de Macaé, que atende à Bacia de Campos. Através dos dados de intervalos entre chegadas das embarcações ao porto, tempos de permanência no fundeio, tempo de permanência no píer e outros dados operacionais, fez-se a análise da taxa de ocupação do porto criando uma ferramenta de auxílio à decisão gerencial.

Silva *et al.* (2010) apresentam um estudo de caso sobre o Terminal Alfandegado de Imbetiba (em Macaé), principal terminal de apoio *offshore* da Petrobras. Utiliza-se a modelagem e simulação a eventos discretos para testar diversos cenários onde se varia o número máximo de descarregamento simultâneo de caminhões em cada berço, o número de empilhadeiras e a demanda por produtos. Como resultado, o modelo apresenta a taxa de utilização dos berços, o *lead-time*, que compreende o tempo entre a saída do produto do armazém e seu embarque no porto.

Fuller, De Arruda e Ferreira Filho (2013) realizam um estudo para dimensionar a frota de navios aliviadores para um conjunto de plataformas. Como método, utilizam-se duas fases. Na primeira, é feita a simulação da produção diária de cada plataforma para definir a distribuição do intervalo necessário entre alívios. Na segunda, busca-se determinar a frota ótima de navios aliviadores analisando a rentabilidade do sistema, através da modelagem de uma Cadeia de Markov. No sistema, são calculados a rentabilidade esperada na venda do óleo produzido, o custo de afretamento dos navios aliviadores, o custo de operação das plataformas e o custo financeiro pelo atraso na produção, em função do número de navios aliviadores.

Ainda no cenário *offshore*, porém com integração com o processo de refino do óleo cru em terra, Limoeiro *et al.*(2008) avaliam, com uso da simulação, investimentos em refinarias e suas cadeias de suprimento, composta pelo transporte do óleo cru em navios ou dutos provenientes das plataformas e suas armazenagens em tanques. Como método do trabalho, os autores implementam, através do *software* Promodel®, a modelagem do sistema de terminais e refinarias da Petrobras, situados na região sudeste do Brasil. Assim, coletou-se dados como vazões de carga e descarga de navios que chegam ao terminal terrestre, as vazões nos dutos na transferência intra-terminais, os tempos de repouso de petróleo em tanques, a taxa de falhas em dutos e o intervalo entre chegadas de navios e seus volumes transportados. A partir desses dados, simularam-se cenários variando número de tanques, vazões de transferência entre navios e o terminal terrestre e alterações na programação de chegadas dos navios. Através das variações nas capacidades dos recursos, foi possível identificar os gargalos do processo e indicar as melhores configurações de capacidade de tancagem, frota de navios e programação de suas chegadas.

2.3 Simulação com Otimização

A técnica da simulação muitas vezes é utilizada em conjunto com métodos exatos de solução para auxiliar tomadas de decisões pontuais durante o período de simulação. A combinação dessas técnicas visa abordar de forma mais realista a programação real de rotas e cronogramas levando em consideração alguns fatores de incerteza, que são tratados no simulador. Dessa forma, sempre que uma condição é verificada no simulador ou mesmo através de uma rotina temporal fixa, alguma rotina de otimização é

chamada para realizar uma programação, retornando os novos resultados para o simulador. Alguns estudos que utilizam técnicas conjuntas de simulação e otimização foram destacados.

Song *et al.*(2008) propõem um sistema que integra simulação com otimização para um problema de balanceamento de linha de montagem. No estudo os autores desenvolvem um sistema de apoio à decisão em tempo real que utiliza técnicas de otimização para balancear uma linha de montagem. Para isso um sistema industrial de produção automático é simulado com o uso do *software* Promodel®. Os modelos de otimização são desenvolvidos em Delphi. A simulação consiste em reproduzir o funcionamento de uma unidade industrial de montagem com 31 operadores, 5 operações e 35 postos de trabalhos. A alocação dos operadores aos postos de trabalho é realizada no momento inicial da simulação. À medida em que algum desbalanceamento ocorre na linha de montagem, as rotinas de otimização são chamadas para realocar os operadores de acordo com seus níveis de eficiência que variam durante a simulação. As sub rotinas de execução dos algoritmos de otimização são chamadas dentro do Promodel® através de um arquivo DLL (*dynamic-link library*) que contém as funções com os códigos desenvolvidos em Delphi. A solução dada pelo otimizador da realocação dos operadores aos postos de trabalho é então obtida pelo Promodel® e a simulação continua, repetindo o processo até o fim da produção.

Halvorsen-Weare *et al.* (2013) realizam um estudo que combina simulação com otimização para o problema de roteamento e programação de cronograma para o escoamento da produção de gás natural liquefeito do porto de produção considerado para os clientes. Nesse estudo, é desenvolvido um modelo de simulação que trata as incertezas das taxas diárias de produção do porto fornecedor e os tempos de navegação dos navios de escoamento da produção. Dessa forma, sempre que ocorrem atrasos acima do permitido, a rotina de otimização é chamada dentro do simulador para reprogramar o roteamento das embarcações e então a nova solução é usada para o resto da simulação. Essas rotinas em conjunto traduzem a situação real de planejamento e programação de rotas.

Melouk *et al.*(2013) utilizam uma abordagem de simulação e otimização para auxiliar na seleção de estratégias de mitigação do congestionamento de uma rede rodoviária para melhorar o tempo de viagem nas estradas dos EUA. Tal abordagem é útil em situações onde o objetivo é encontrar um conjunto entre muitos conjuntos de especificações do modelo. Dessa forma, o modelo de simulação descreve um cenário

estocástico complexo e o componente de otimização fornece conjuntos de soluções de avaliação para o modelo de simulação. O modelo de otimização consiste em uma metaheurística baseada em Busca Tabu que gera soluções para o modelo de simulação avaliar. Durante a fase de simulação avalia-se qual solução minimiza o total de horas que um veículo gasta na rede rodoviária.

Melouk *et al.* (2013) apresentam um estudo sobre ao processo de produção de aço no qual utilizam simulação combinado com otimização para minimizar os custos com estoque em processo. O modelo de simulação, construído com o *software* Arena, reproduz o processo produtivo de aço que é acionado a partir de cronogramas de produção. Esses cronogramas discriminam o tipo de produto e todos os processos requisitados pela ordem do cliente. Para isso existe um estoque pulmão intermediário de peças para a fabricação e o objetivo é minimizar os custos com esse estoque ao mesmo tempo em que previne contra variabilidades previstas nos tempos de produção e nas falhas aleatórias das máquinas. O problema de otimização visa determinar o nível ótimo de estoque das peças intermediárias no início de cada campanha de produção. Como resultado, o estudo aponta reduções de custo com estoque ao ajustar o nível de estoque ideal para o processo produtivo.

Lopes e Da Cunha (2012) desenvolvem um modelo de simulação/otimização para auxiliar a operação de jatos de propriedade compartilhada. O objetivo do estudo é determinar a frota ótima de aeronaves e a alocação destas aos vôos diários de forma a minimizar viagens vazias. Para isso a demanda diária dos vôos é gerada através de simulação de Monte Carlo e o modelo de otimização busca a solução ótima para o cenário gerado. Os autores usam a linguagem VBA e o pacote de otimização What's Best!®.

Ko, Ko e Kim (2006) propõem um método de otimização/simulação para projetar uma rede de distribuição para um provedor terceirizado de serviços logísticos. No estudo é avaliado o desempenho dos armazéns onde um modelo de otimização usa o Algoritmo Genético para determinar as redes dinâmicas de distribuição. O modelo de simulação então é aplicado para medir o tempo de serviço de cada rede sob as incertezas de demanda dos clientes, tempo de preparação do pedido e tempo de viagem. O tempo do pedido consiste em uma variável aleatória do tempo de processamento do pedido no armazém e no tempo de viagem entre o armazém e o cliente. Em um procedimento de retroalimentação, utilizam-se os resultados do simulador como dados de entrada do otimizador. O modelo de otimização é formulado como um modelo de localização

capacitado de dois elos, multi período e multi produto. A instância considerada no estudo é composto por 3 clientes, com total de 90 lojas, 3 produtos e 10 potenciais localizações de armazéns. Dessa forma, o modelo de otimização gera soluções ótimas de locação dos clientes aos armazéns (considerando parâmetros de custos), alimentando o simulador que gera os dados aleatórios de demandas e tempos e verifica se aquela solução atende às medidas de desempenho desejadas.

3 Definição do Problema

O estudo consiste em modelar duas políticas logísticas de suprimento de óleo diesel para unidades marítimas (UMs) que compreendem unidades produtoras de petróleo (UEPs) e sondas de perfuração que compõem o cenário *offshore* da companhia petrolífera. As políticas avaliadas diferem entre si sob o aspecto do acionamento do fluxo logístico. Em uma delas, este acionamento é realizado pelo cliente, na outra, o fornecedor é responsável por planejar as entregas. A primeira política é então denominada de “política de entregas sob demanda” e a segunda é denominada “política de entregas programadas”. Dessa forma, na política de entregas sob demanda, o atendimento à demanda é realizado à medida que as unidades marítimas disparam pedidos de óleo diesel. Já na política de entregas programadas, as unidades marítimas são divididas em grupos, para os quais viagens regulares de embarcações carregadas de óleo diesel são programadas com base no consumo esperado das unidades.

Na literatura acadêmica é possível encontrar termos que definem os cenários logísticos avaliados no estudo. Wanke (2016) apresenta o conceito de posicionamento logístico. Segundo o autor, há cinco categorias de decisão que devem ser articuladas para traçar uma estratégia de posicionamento logístico. Entre elas está a coordenação do fluxo de produtos. Esta categoria, utilizando-se dos termos empregados pelo autor, decide se o fluxo de produtos deve ser “puxado” ou “empurrado”. O fluxo puxado é acionado pelo elo que está mais próximo ao consumidor final e o fluxo empurrado é coordenado pelo elo que está mais próximo ao fornecedor inicial. Fazendo uma analogia com o estudo, a política de entregas sob demanda pode ser interpretada como o fluxo logístico puxado e a política de entregas programadas pode ser interpretada como o fluxo logístico empurrado. Ainda segundo Wanke (2016), em geral, a decisão entre puxar ou empurrar depende da análise conjunta de dois fatores: visibilidade da demanda e tempos de ciclo de ressuprimento e distribuição. Enquanto que o fluxo puxado é direcionado pela demanda real do cliente, o fluxo empurrado é direcionado pela previsão da demanda do cliente. Já o tempo de ciclo e de ressuprimento referem-se ao tempo de entrega do produto para o cliente.

Pode-se, mais uma vez, realizar um paralelo entre os conceitos apresentados de fluxo logístico com o presente estudo, visto que a política de entregas programadas é direcionada pelo consumo esperado das UMs (previsão da demanda) e a política de

entregas sob demanda é direcionada pelos pedidos das UMs (demanda real). Dessa forma, o sistema logístico conta com uma frota de PSVs que transportam exclusivamente óleo diesel, denominados PSVs oleiros destinados a suprirem um conjunto de UMs. Atualmente, a política logística adotada pela companhia em estudo é a política de entregas sob demanda.

Para conduzir o estudo, foi implementado um modelo de simulação a eventos discretos que reproduz o sistema logístico da companhia descrito pelas regras operacionais das políticas que se deseja avaliar. O modelo simula 10 cenários variando o tamanho da frota de PSVs oleiros para cada política avaliada.

Então, para comparar a eficiência operacional entre as duas políticas, analisam-se os resultados dos experimentos dos 10 cenários simulados para cada uma delas observando-se os indicadores de desempenho e os custos. Busca-se responder qual o cenário ideal para cada política, isto é, qual cenário apresenta menor custo logístico total mantendo-se um nível de serviço desejável, ou seja, aquele que não admite ocorrências de desabastecimentos. O estudo também avalia duas regras de alocação de PSVs que são executadas em uma sub rotina externa ao simulador. Essas sub rotinas substituem a regra padrão FIFO (*first-in-first-out*) de alocação adotada pelo simulador por formulações matemáticas de programação inteira mista que buscam soluções ótimas de acordo com critérios estabelecidos.

3.1 Descrição do Sistema Logístico

O sistema logístico é formado por 56 UMs sendo elas 44 UEPs e 12 sondas de perfuração. Essas unidades são organizadas em 11 grupamentos denominados *clusters*. O sistema conta com embarcações do tipo PSV exclusivas para transporte de óleo diesel, denominados PSVs oleiros e um navio tanque que é responsável por fornecer óleo diesel para os PSVs oleiros. Além disso, existe um porto para o qual os PSVs se dirigem sempre que precisam realizar troca de tripulação. A Figura 3.1 apresenta o sistema logístico simplificado do estudo, no qual são destacados os locais de visitaçao do PSV oleiro ao decorrer da programação.

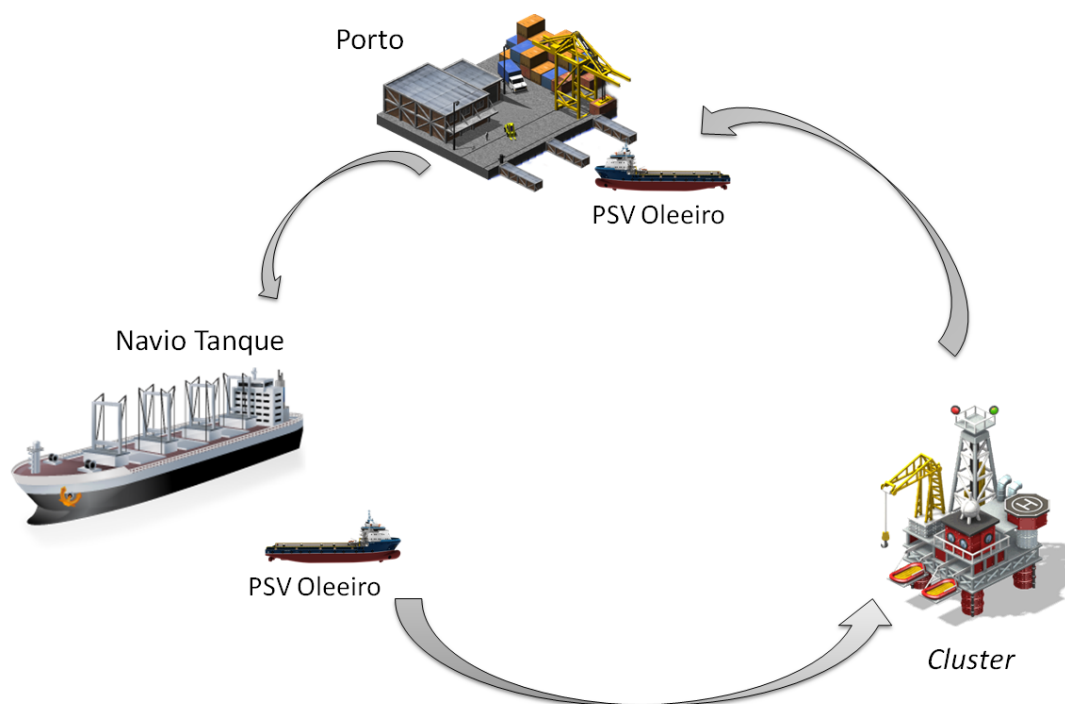


Figura 3.1 – Sistema logístico simplificado

Na Tabela 3.1 são apresentados os *clusters* e as UMs que os compõem, numeradas de 1 até 56. As UMs de numeração 1 até 44 são UEPs e as UMs de numeração 45 até 56 são sondas de perfuração. O número de UMs agrupadas por cluster varia. O *cluster* com menor número de UMs é o *cluster* Sonda3 e o *cluster* com maior número de UMs é o *cluster* UEP6. O critério de agrupamento adotado pela companhia leva em conta a proximidade geográfica entre as UMs e seus padrões de consumo. Além disso, a companhia distingue *clusters* compostos por UEPs e *clusters* compostos por sondas.

Tabela 3.1 – composição dos *clusters*

UEP1	UM1	UM2	UM3	UM4	UM5	-	-
UEP2	UM6	UM7	UM8	UM9	UM10	-	-
UEP3	UM11	UM12	UM13	UM14	UM15	UM16	-
UEP4	UM17	UM18	UM19	UM20	UM21	-	-
UEP5	UM22	UM23	UM24	UM25	UM26	UM27	-
UEP6	UM28	UM29	UM30	UM31	UM32	UM33	UM34
UEP7	UM35	UM36	UM37	UM38	-	-	-
UEP8	UM39	UM40	UM41	UM42	UM43	UM44	-
Sonda1	UM45	UM46	UM47	UM48	UM49	-	-
Sonda2	UM50	UM51	UM52	UM53	-	-	-
Sonda3	UM54	UM55	UM56	-	-	-	-

3.2 Simplificações e Premissas

Para desenvolver o modelo, algumas simplificações e premissas foram impostas, conforme seguem.

Simplificações:

1. Um *cluster* é representado fisicamente no modelo por um local único, mas descreve as características e o comportamento de um grupo de UMs.

Justificativa: o *software* Promodel® de modelagem e simulação a eventos discretos foi escolhido para conduzir o trabalho, pois é a ferramenta utilizada pela companhia estudada. Este *software* permite construir uma representação gráfica do sistema a ser modelado, porém sua estrutura de modelagem gráfica é bastante limitada em termos de escala, inviabilizando a construção de modelos com um número muito grande de locais. Por isso optou-se por utilizar um local único para representar um conjunto de UMs.

2. O modelo considera apenas um NT com capacidade de atender até no máximo três PSVs oleiros ao mesmo tempo. Esta simplificação pode ser interpretada como se houvesse sempre três NTs disponíveis na bacia e sempre com capacidade suficiente de suprimento.

Justificativa: o sistema logístico da companhia estudada conta com seis monobóias de atracação de NTs. Porém esses NTs não estão sempre disponíveis em alto mar, pois frequentemente precisam reabastecer em terminais terrestres. O número mais provável de NTs disponíveis em alto mar, de acordo com dados históricos da companhia é três. Para efeito de simplificação, observou-se qual a monobóia mais utilizada para atracação de um NT e considerou-se esta como local comum de três NTs. Dessa forma, o efeito de um único NT abastecer 3 PSVs simultaneamente representa a disponibilidade de três NTs num mesmo local em alto mar.

3. O estoque de óleo diesel do NT é repostado imediatamente após um PSV abastecer. Isso impõe uma capacidade infinita de suprimento para o NT.

Justificativa: a companhia opera com seis monobóias de NTs, mas o número mais provável de disponibilidade é de três NTs. Isso ocorre devido às frequentes visitas a terminais terrestres para reabastecimento. Para efeito de simplificação

- optou-se por não tratar as operações de abastecimento dos NTs em terra e, portanto, considera-se que sempre há três NTs disponíveis com estoque infinito.
4. A localização do porto é considerada distante 36 minutos do NT.
Justificativa: o tratamento das operações de troca de tripulação é considerada como uma indisponibilidade do elemento “recurso” no modelo de simulação. Nesse caso, a localização do porto não é real e sim apenas uma região física onde esses recursos permanecem por um tempo estabelecido por uma distribuição de probabilidade.
 5. A programação das indisponibilidades dos PSVs oleiros devido a troca de tripulação ou inoperância incide sobre o recurso “rPSV” e não sobre a entidade “ePSV_OD”. Esses elementos são descritos na Seção 4.2.
Justificativa: o elemento “recurso” no *software* utilizado pode ser programado para sofrer indisponibilidades em uma lógica própria destinada a esse procedimento. Já os elementos “entidade” necessitam de uma programação desenvolvida especificamente para eles, o que não é eficiente, do ponto de vista de processamento computacional.
 6. Os tempos de navegação são considerados constantes e calculados a partir das velocidades médias dos PSVs oleiros.
Justificativa: os dados disponibilizados pela companhia permite verificar os tempos de navegação dos PSVs entre as UMs assim como as distâncias percorridas. Dessa forma, calculou-se a velocidade média do PSV.
 7. Os fatores estocásticos que influem na velocidade de navegação e nos tempos de aguardo dos PSVs por condições propícias de operação e navegação, como alturas das ondas, vento e demais condições metereológicas não são considerados no modelo.
Justificativa: para efeito de simplificação foram calculados os tempos médios de navegação e operação e considera-se a estocasticidade das condições metereológicas refetida no cálculo dessas médias.
 8. O modelo simula apenas as operações em alto mar. A logística de abastecimento do NT nos terminais em terra não é abrangida no estudo.
Justificativa: para efeito de simplificação optou-se por não simular as operações de abastecimento dos NTs em terra.

9. As operações de transferência de óleo diesel ocorrem imediatamente após o PSV atracar na UM ou no NT. Os tempos de aguardo por condições meteorológicas favoráveis não são considerados no modelo.

Justificativa: os tempos operacionais foram fornecidos pela companhia através das taxas médias de transferência de óleo diesel. Essas taxas foram obtidas através da observação dos dados relativos ao tempo total de aguardo e operação e os volumes de óleo diesel transferidos. Logo a média já carrega as informações de aguardo por condições meteorológicas.

A simplificação de considerar os fatores estocásticos das condições meteorológicas refletidas nas médias dos tempos de navegação e tempos de operação, por um lado, afasta o modelo da realidade operacional, mas por outro, simplifica o processamento computacional, permitindo construir um primeiro objeto de análise das políticas avaliadas.

Premissas:

1. As vazões de transferência entre o PSV oleeiro e as unidades marítimas, representadas pelos *clusters* ou entre o PSV oleeiro e o NT são determinísticas. A taxa de transferência entre um PSV oleeiro e o NT é de 60 m³/hora e entre um PSV e um *cluster* é de 25 m³/hora.
2. A frota de PSVs oleiros é homogênea, isto é, todos os PSVs possuem a mesma capacidade de tanque de 2500m³.
3. Considera-se o custo de afretamento de um PSV oleeiro constante em relação ao tempo e igual para toda a frota.
4. As operações de abastecimento das UMs são realizadas dia e noite, isto é, 24 horas por dia.
5. Os PSVs oleiros possuem um tanque de óleo diesel para consumo próprio e o volume do tanque é suficiente para suprir mais que 28 dias de operação.
6. Os PSVs oleiros reabastecem seus tanques de óleo diesel para consumo próprio no NT.

Os dados de vazões médias de transferência de fluidos foram fornecidos pela companhia através da relação entre os dados históricos dos tempos operacionais de

carga/descarga dos PSV nas UMS e nos NTs e os volumes transferidos durante essas operações.

A partir das premissas 1 e 2, pode-se inferir que o tempo máximo gasto por um PSV oleeiro para realizar o abastecimento no NT é de 41,7 horas e o tempo máximo de operação em um *cluster* é de 100 horas.

A premissa 3 é considerada devido à política de contratação a longo prazo praticada pela companhia. Dessa forma o custo de afretamento de um PSV (em inglês *time charter*) é estabelecido no ato do contrato e não varia em menos de um ano. Além disso, ao considerar frota homogênea, pressupõe-se que os valores de contrato são os mesmos para todos os PSVs.

3.3 Dados de Entrada do Modelo

Os dados necessários para a construção do modelo são provenientes da base de dados da companhia estudada, base de dados públicos, assim como da literatura acadêmica.

Os dados de entrada são organizados em uma planilha eletrônica MS-Excel®. Esses dados são acessados pelo modelo durante a lógica de inicialização e importados para estruturas de dados internas como variáveis e matrizes e podem ser determinísticos ou estocásticos. Os dados determinísticos compreendem a capacidade máxima dos tanques dos *clusters*, os níveis de segurança adotados por cada *cluster* e seus pontos de pedido, a especificação da frota de PSVs oleiros, os tempos de navegação entre os locais do modelo, as taxas de transferência de fluidos entre outros dados necessários para construção do modelo. Já os dados estocásticos compreendem a distribuição da taxa de consumo dos *clusters*, a distribuição do volume de entrega ou volume do pedido, os tempos de permanência dos PSVs no porto para troca de tripulação, assim como a frequência e duração das inoperâncias dos PSVs (falhas aleatórias).

3.3.1 Dados referentes aos *clusters*

Os dados de entrada são referentes às especificações dos *clusters*, dos PSVs oleiros, assim como dados gerais, e dados de tempos de navegação. Na Tabela 3.2 podem-se verificar os dados de entrada referentes aos *clusters*.

Tabela 3.2 – Informações dos clusters

Local	Lat.	Long.	Capacidade máxima do tanque (m ³)	Estoque de segurança(m ³)	Lead-time médio (dias)	Consumo médio (m ³ /dia)	Lead-time*consumo	Ponto de Pedido(m ³)
UEP1	-22,71	-40,69	7014	1861	4	208,21	832,85	2693,85
UEP2	-22,60	-40,22	9169	3815	4	309,27	1237,09	5052,09
UEP3	-22,06	-39,84	14701	6223	4	259,55	1038,22	7261,22
UEP4	-22,37	-40,03	9739	4420	4	263,39	1053,57	5473,57
UEP5	-22,45	-40,08	9891	3664	4	145,41	581,62	4245,62
UEP6	-22,23	-40,26	3619	895	4	22,24	88,94	983,94
UEP7	-22,55	-40,10	7188	3355	4	43,63	174,52	3529,52
UEP8	-22,67	-40,62	4351	1025	4	60,33	241,31	1266,31
Sonda1	-22,43	-40,03	12980	3400	4	164,29	657,16	4057,16
Sonda2	-22,25	-40,44	6704	1510	4	81,51	326,03	1836,03
Sonda3	-22,66	-40,85	6575	1329	4	70,01	280,04	1609,04
NT	-22,30	-40,20	Infinito	-	-	-	-	-

3.3.1.1 Consumo

A modelagem dos dados de consumo foi obtida a partir da observação de um ano de dados de consumo das UMs consideradas no estudo. O período de observação compreende de 1º de janeiro de 2015 até 31 de dezembro de 2015. Os dados foram fornecidos em uma planilha MS-Excel® em que as linhas compreendem os 365 dias do ano e as colunas representam os volumes consumidos em cada data por cada UM. A planilha consta alguns dias sem registro de consumo, que então foram removidos da análise.

Após a coleta e primeira análise dos dados fornecidos, foram feitos alguns tratamentos estatísticos como remoção de *outliers* – dados muito destoantes do conjunto observado. Segundo Chwif e Medina (2013), existem algumas técnicas utilizadas para identificar *outliers* em uma amostra. O mais utilizado é a construção de intervalos de valores a partir dos quartis da amostra. Este método que foi utilizado para remover *outliers*.

Considerando que Q_1 e Q_3 são respectivamente os valores do primeiro e terceiro quartis, a amplitude inter-quartil, A é calculada pela diferença:

$$A = Q_3 - Q_1 \quad (3.1)$$

A partir da diferença definida na Equação (3.1), considera-se *outlier* qualquer valor que estiver abaixo de $Q_1 - 1.5A$ ou acima de $Q_3 + 1.5A$. O cálculo do corte de *outliers* é recomendação da companhia.

Após o tratamento dos dados, foram agrupados os dados relativos às UMs pertencentes a cada *cluster* e somados seus cosumos diários. Em seguida, foram realizados testes estatísticos para verificar se alguma função de distribuição de probabilidade se ajustava aos dados de consumo dos *clusters*. Para isso, foi utilizado o *software* StatFit®. Este software organiza os dados de entrada em histograma e então realiza diversos testes estatísticos para inferir qual melhor distribuição de probabilidade que representa o fenômeno estudado.

A partir dos dados de entrada, o StatFit® constrói o histograma da amostra e realiza o teste de aderência para as funções de distribuição de probabilidade que se deseja testar. Para isso, utilizam-se os testes de Kolmogorov-Smirnov (KS), Anderson-Darling (AD) ou o teste do qui-quadrado (X_2), de acordo com as características da amostra. Para estimar os parâmetros das funções, podem ser escolhidos o método da máxima verossimilhança ou o método dos momentos. Como exemplo, na Figura 3.2 pode-se verificar os resultados dos testes KS e AD para a distribuição Lognormal, ajustada para representar o consumo do *cluster* Sondal.

Lognormal		
minimum	=	148. [fixed]
mu	=	2.40221
sigma	=	0.968035
Kolmogorov-Smirnov		
data points		51
ks stat		0.114
alpha		5.e-002
ks stat[51,5.e-002]		0.187
p-value		0.483
result		DO NOT REJECT
Anderson-Darling		
data points		50
ad stat		0.63
alpha		5.e-002
ad stat[5.e-002]		2.49
p-value		0.62
result		DO NOT REJECT

Figura 3.2 – Testes de aderência

A Figura 3.3 mostra o ranking das distribuições ajustadas à amostra e seus parâmetros para o cluster Sonda1.

distribution	rank	acceptance
Lognormal(148, 2.4, 0.968)	71.9	do not reject
LogLogistic(148, 1.73, 11.3)	69.1	do not reject
Exponential(148, 16.3)	67.5	do not reject
Pearson 6(148, 28.2, 1.87, 4.07)	63.3	do not reject
Pareto(148, 9.91)	47.7	do not reject
Gamma(148, 1.43, 11.4)	43.6	do not reject
Pearson 5(148, 1.18, 8.08)	41.5	do not reject
Inverse Weibull(148, 1.03, 0.148)	40.	do not reject
Weibull(148, 1.2, 17.7)	38.	do not reject
Inverse Gaussian(148, 11.8, 16.3)	32.	do not reject
Beta(148, 198, 0.831, 1.55)	3.14	do not reject
Power Function(148, 198, 0.662)	0.197	reject
Triangular(147, 203, 150)	0.104	reject
Erlang(148, 1., 11.4)	1.33e-002	reject
Rayleigh(148, 15.3)	0.	reject
Uniform(148, 198)	0.	reject
Chi Squared(148, 12.)	0.	reject
Johnson SB(148, 43.2, 0.631, 0.688)	0.	reject

Figura 3.3 – Ranking de ajustes de funções

A partir do ranking das melhores curvas, escolhe-se a que mais convém para então ser usada como entrada do modelo de simulação. A primeira distribuição do ranking é aquela que tem o melhor ajuste para os dados analisados. Para o caso do exemplo, escolheu-se a primeira do ranking, isto é a distribuição lognormal (148; 2,4; 0,968), conforme mostra a Figura 3.4. Nesta distribuição o primeiro parâmetro é o valor mínimo, o segundo parâmetro é o μ e o terceiro parâmetro é o σ .

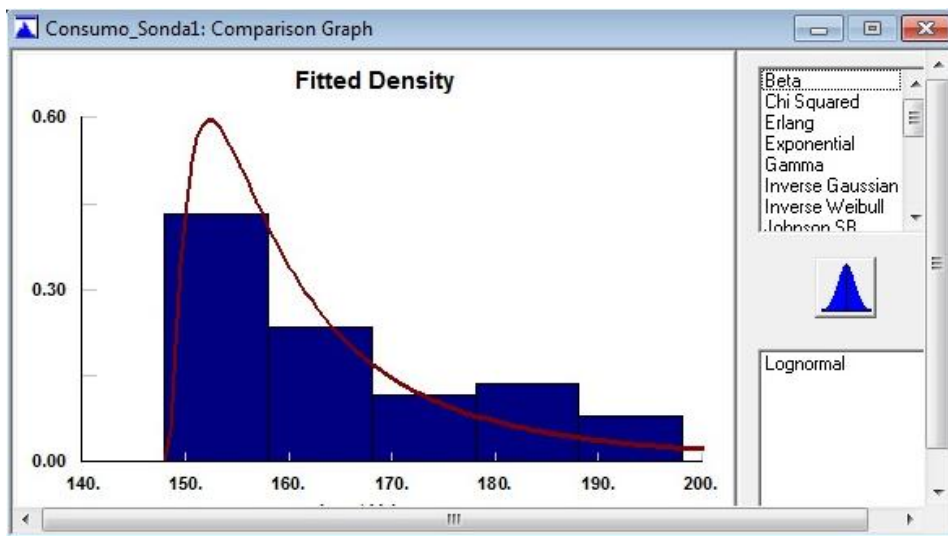


Figura 3.4 – Aderência da função lognormal para o cluster Sonda1

Todas as funções de distribuição de probabilidade ajustadas para os consumos encontram-se em Apêndice A – Curvas de Aderência Consumo.

A Tabela 3.3 apresenta as funções de distribuições ajustadas aos dados de consumo por dia, as médias dos dados coletados (após a retirada dos *outliers*) e os desvios-padrão.

Tabela 3.3 – Distribuição dos consumos diários

Cluster	Distribuição do consumo diário (m^3)	Média (m^3)	Desvio-padrão
UEP1	Triangular(172; 281; 172)	208,21	27,42
UEP2	Beta(241; 408; 0,828; 1,21)	309,27	51,05
UEP3	Pearson 6(210; 4,82e+004; 1,43; 1,23e+003)	259,55	37,95
UEP4	Weibull(220; 1,53; 49,5)	263,39	29,09
UEP5	Beta(123; 180; 0,8; 1,18)	145,41	18,60
UEP6	LogLogistic(19; 3,77; 3,07)	22,24	2,36
UEP7	Triangular(33; 64,5; 33)	43,63	8,67
UEP8	Triangular(54; 70,6; 55,2)	60,33	3,94
Sonda 1	Lognormal(148; 2,4; 0,968)	164,29	14,10
Sonda 2	Triangular(71; 101; 71,7)	81,52	7,73
Sonda 3	Uniform(56; 85)	70,01	9,15

As estatísticas descritivas dos dados de consumo após remover os *outliers* encontram-se no Apêndice B – Estatísticas Descritivas.

3.3.1.2 Entregas

A modelagem dos dados de entrega foi obtida a partir da observação de um ano de dados de entregas para aquelas UMs consideradas no estudo, relativo ao mesmo período de observação dos dados de consumo. Na planilha da companhia, há o histórico dos pedidos e entregas às UMs. Nela, constam a data de criação do pedido, a UM que realizou o pedido, a quantidade solicitada, a data de entrega e a quantidade entregue. Os dados de volumes entregues às UMs são relevantes somente para modelar a política de entregas sob demanda. Nessa política, os *clusters* solicitam uma quantidade definida de óleo diesel para o fornecedor. Para traduzir os pedidos individuais das UMs em um pedido único do *cluster*, foram agregados os pedidos de 4 dias observados. Dessa forma, criaram-se 90 grupos de 4 dias e somou-se a quantidade pedida por grupo por *cluster*. Foi escolhido realizar o agrupamento de 4 dias pois este é o *lead-time* médio observado das entregas: a partir da planilha de companhia, foi verificado que a média da diferença entre data de criação do pedido e data de entrega do pedido é de 4 dias. Dessa

forma, se para cada *cluster* houver pelo menos um pedido pertencente a cada um dos grupos, então no máximo haverá 90 observações por *cluster*.

Não houve remoção de *outliers* para as observações referentes às entregas. O procedimento para ajustar curvas de distribuição de probabilidades foi exatamente o mesmo realizado para os dados de consumo. A Tabela 3.4 mostra as funções de distribuição encontradas para os volumes entregues, utilizadas para modelar o comportamento do volume do pedido na política de entregas sob demanda, assim como as médias e os desvios-padrão para os dados analisados de cada *cluster*. As estatísticas descritivas dos dados de volume entregue encontram-se no Apêndice B – Estatísticas Descritivas.

Tabela 3.4 – Distribuição dos volumes entregues

Cluster	Distribuição do volume do pedido	Média (m^3)	Desvio-padrão
UEP1	Beta(67; 1,97e+003; 0,719; 1,4)	701,88	508,95
UEP2	Pearson 6(16; 2,44e+003; 2,87; 9,45)	823,69	515,61
UEP3	Inverse Weibull(3; 1,52; 1,92e-003)	930,80	746,11
UEP4	Beta(100; 3,14e+003; 0,998; 2,92)	854,92	587,48
UEP5	Gamma(20; 1,84; 353)	668,16	450,24
UEP6	Uniform(10; 115)	62,75	32,68
UEP7	Pearson 6(50; 5,49e+004; 2,; 194)	614,53	390,25
UEP8	Beta(42; 1,18e+003; 0,875; 2,49)	327,75	242,11
Sonda 1	Johnson SB(135; 1,15e+004; 3,83; 1,23)	756,64	443,21
Sonda 2	Weibull(7; 1,74; 565)	497,95	310,76
Sonda 3	Inverse Weibull(200; 1,6; 4,59e-003)	574,14	295,10

Todos os gráficos das funções de distribuição de probabilidade ajustadas para os volumes de pedido dos *clusters* encontram-se no Apêndice C – Curvas de Aderência Entregas.

3.3.2 Distâncias e tempos

Para modelar o sistema é necessário conhecer os tempos operacionais e os tempos de navegação dos PSVs. Os tempos de navegação foram calculados considerando a velocidade média dos PSVs de 15 Km/h. Já os tempos de operação nos *clusters* são traduzidos pela vazão de transferência de óleo diesel entre o PSV e os *clusters*. Para se ajustar a taxa de transferência, são realizadas correlações entre tempos operacionais nas UMs e volume transferido durante a operação. A partir da observação de um ano de

operações realizadas em UMs e nos navios tanque, pode-se ajustar uma vazão média de transferência como sendo a razão entre o volume transferido nas operações e os tempos gastos nas mesmas. Como os *clusters* são modelados como um conjunto de UMs, os tempos médio de navegação entre as UMs pertencentes a eles são incorporados ao tempo total de operação e assim o modelo emula os tempos operacionais e os tempos de navegação entre as UMs do mesmo *cluster* refletido pela vazão de transferência.

A localização dos *clusters* foi estabelecida através do cálculo do centro geométrico ou centróide do conjunto das UMs agrupadas no *cluster*. No sistema logístico real, o número de navios tanque disponíveis na bacia varia de 0 a 6. As bóias de atracação dos navios tanque possuem localização fixa, porém os navios que atracam nas bóias não são sempre os mesmos. Isso porque os navios tanque precisam retornar a algum porto para reabastecerem. Como simplificação, o modelo adota como premissa a existência de um único navio tanque que fica localizado na bacia. Essa localização foi definida a partir da análise da bóia de localização dos navios tanque que possui maior número de atracações.

A partir das coordenadas geográficas dos centros geométricos encontrados para localizar os *clusters*, foi calculada a matriz de distância entre os *clusters* e entre o NT e os *clusters*, como mostrada na Tabela 3.5.

Tabela 3.5 – Matriz de distâncias em Km

Cluster	NT	UEP1	UEP2	UEP3	UEP4	UEP5	UEP6	UEP7	UEP8	Sonda1	Sonda2	Sonda3
UEP 1	67,77	0,00	50,23	113,13	77,70	68,90	69,06	62,68	8,56	73,89	56,87	17,31
UEP2	33,41	50,23	0,00	71,30	32,14	21,79	41,83	12,65	42,00	26,08	45,54	65,67
UEP3	45,52	113,13	71,30	0,00	39,29	49,73	47,35	60,93	104,75	46,27	65,50	123,47
UEP4	19,04	77,70	32,14	39,29	0,00	10,44	28,80	21,82	69,14	7,67	44,59	90,51
UEP5	20,54	68,90	21,79	49,73	10,44	0,00	31,06	11,68	60,36	5,06	43,39	82,58
UEP6	10,53	69,06	41,83	47,35	28,80	31,06	0,00	39,69	61,23	33,29	18,61	77,23
UEP7	29,64	62,68	12,65	60,93	21,82	11,68	39,69	0,00	54,35	14,69	48,46	77,72
UEP8	59,49	8,56	42,00	104,75	69,14	60,36	61,23	54,35	0,00	65,37	50,12	24,08
Sonda1	22,85	73,89	26,08	46,27	7,67	5,06	33,29	14,69	65,37	0,00	47,04	87,64
Sonda2	25,74	56,87	45,54	65,50	44,59	43,39	18,61	48,46	50,12	47,04	0,00	62,01
Sonda3	78,05	17,31	65,67	123,47	90,51	82,58	77,23	77,72	24,08	87,64	62,01	0,00

Após calcular a matriz de distâncias, obteve-se a matriz de tempos de navegação entre os locais. Para isso, foram observados os dados históricos das velocidades de navegação dos PSVs oleiros oriundos da planilha de registros de operações de embarcações da companhia do período de um ano considerado na análise. A partir da

coleta dos dados, foi calculada a velocidade média. A Tabela 3.6 mostra as velocidades por trecho de interesse.

Tabela 3.6 – Velocidades por trecho

Trecho	Velocidade (km/hora)
NT-Cluster	15,18
Entre clusters	12,85
Cluster-NT	16,03

A velocidade difere por trecho analisado, pois o PSV desenvolve uma maior velocidade para navegar distâncias maiores. Além disso, o PSV parte do NT com estoque cheio e retorna com estoque vazio, desenvolvendo uma velocidade maior na volta. De acordo com as velocidades médias por trecho obteve-se a matriz de tempos, mostrada pela Tabela 3.7. Os tempos são resultados da divisão entre distância e as velocidades médias e são representados em horas. Além dos locais listados na matriz, o porto é considerado 36 minutos distante do navio tanque, e não se considera a necessidade de aguardo para atracação.

Tabela 3.7 – Matriz de tempos de navegação em horas

Locais	NT	UEP1	UEP2	UEP3	UEP4	UEP5	UEP6	UEP7	UEP8	Sonda1	Sonda2	Sonda3
NT	0,00	4,46	2,20	3,00	1,25	1,35	0,69	1,95	3,92	1,50	1,69	5,14
UEP1	4,23	0,00	3,91	8,80	6,05	5,36	5,37	4,88	0,67	5,75	4,42	1,35
UEP2	2,08	3,91	0,00	5,55	2,50	1,69	3,25	0,98	3,27	2,03	3,54	5,11
UEP3	2,84	8,80	5,55	0,00	3,06	3,87	3,68	4,74	8,15	3,60	5,10	9,61
UEP4	1,19	6,05	2,50	3,06	0,00	0,81	2,24	1,70	5,38	0,60	3,47	7,04
UEP5	1,28	5,36	1,69	3,87	0,81	0,00	2,42	0,91	4,70	0,39	3,38	6,42
UEP6	0,66	5,37	3,25	3,68	2,24	2,42	0,00	3,09	4,76	2,59	1,45	6,01
UEP7	1,85	4,88	0,98	4,74	1,70	0,91	3,09	0,00	4,23	1,14	3,77	6,05
UEP8	3,71	0,67	3,27	8,15	5,38	4,70	4,76	4,23	0,00	5,09	3,90	1,87
Sonda1	1,42	5,75	2,03	3,60	0,60	0,39	2,59	1,14	5,09	0,00	3,66	6,82
Sonda2	1,60	4,42	3,54	5,10	3,47	3,38	1,45	3,77	3,90	3,66	0,00	4,82
Sonda3	4,87	1,35	5,11	9,61	7,04	6,42	6,01	6,05	1,87	6,82	4,82	0,00

3.3.3 Dados referentes aos PSVs oleiros

Além dos dados de tempos de navegação, vale ressaltar que os tempos de operação dos PSVs oleiros no NT e nos clusters são determinísticos e variam conforme o volume transferido e das vazões de transferência, descritas na seção 3.2. A frota de

PSVs oleiros é homogênea e possui capacidade máxima de armazenamento de 2500 m³ de óleo diesel, como também citado na seção 3.2.

Os eventos de troca de tripulação e inoperância são descritos de acordo com a Tabela 3.8 e também foram obtidos a partir da observação dos registros operacionais das embarcações da companhia durante o período analisado.

Tabela 3.8 – Troca de tripulação e inoperância

Evento	Frequencia de ocorrência	Duração da ocorrência
Troca de tripulação	28 dias	Triangular (14; 16; 20) horas
Inoperância	Triangular (20; 30; 40) dias	Triangular (0,5; 1; 10) dias

3.3.4 Custos

Para analisar os resultados do custo total do sistema, são considerados os custos de afretamento de PSVs, os custos operacionais dos PSVs, os custos de estoque, isto é, capital imobilizado em estoque e os custos de falta, ou de parada.

A política contratual da empresa trabalha com contratos de longo prazo, isto é, os PSVs são alocados por um período superior a um ano. A partir da informação de especialistas logísticos e também segundo Uglane, Friedberg e Andersson (2012), o custo diário de afretamento de um PSV foi fixado em 30.000 USD. Já o custo operacional é considerado como 15% do custo de afretamento e representa os custos de consumo de combustível e desgaste/manutenção do equipamento. Este custo representa o custo do PSV em uso, isto é, operando em mar ou navegando. O custo operacional do PSV parado é igual a zero.

O custo de estoque compreende o custo do capital imobilizado em estoque. Este custo é calculado como o custo de oportunidade do capital imobilizado, considerando uma taxa de atratividade de 10% ao ano. Para tal cálculo, foi considerado que o preço do litro de óleo diesel é de 0,75 USD. Além disso, o custo de afretamento diário (em inglês, *time charter*) de uma sonda de perfuração é de 250.000 USD.

Os custos de afretamento dos PSVs, das sondas de perfuração, do barril de petróleo *Brent* e do litro de óleo diesel são necessários para calcular o custo total do sistema no fim da simulação. Os custos de afretamento dos PSVs e das sondas de perfuração foram definidos segundo informação de especialistas do mercado de logística de apoio *offshore* e também de acordo Uglane, Friedberg e Andersson (2012) que realizaram um estudo de caso de roteamento e programação de PSVs no cenário da produção *offshore*

brasileira. Já o custo do litro de óleo diesel e do barril de petróleo *Brent* foram obtidos a partir do sistema de levantamento de preços (ANP, 2015a). Tais custos são resumidos na Tabela 3.9.

Tabela 3.9 – Custos

Custos	Valor (USD)
Afretamento diário PSV	30.000,00
Afretamento diário de sonda de perfuração	250.000,00
Petróleo <i>Brent</i> (barril)	45,00
Litro do óleo diesel	0,75

Os custos de falta são calculados a depender se o cluster é constituído por sondas ou UEPs. Para *clusters* formados por sondas, o custo de falta é o custo do afretamento da sonda. Para *clusters* formados por UEPs o custo de falta compreende o custo da remuneração do capital a uma taxa de atratividade de 10% ao ano da receita não gerada devido ao desabastecimento de óleo diesel e a consequente interrupção da produção. Para esse cálculo foi considerado a receita média gerada por hora relativa à produção média (em barris por hora) daquele *cluster*.

Dessa forma, os custos de falta dos *clusters* de UEPs foram calculados com base no histórico da produção diária referentes ao ano de 2015 das UMs pertencentes aos *clusters*. Esses dados também foram obtidos na base de dados BDEP (ANP, 2015b). O custo de falta dos *clusters* compostos por sondas é simplesmente o custo de afretamento diário do total de sondas que compõem o *cluster*. A Tabela 3.10 apresenta os valores base para calcular os custos de falta referentes aos *clusters*.

Tabela 3.10 – Valores base para cálculo de custo de falta

Cluster	Média Produção (bbl/dia)	Receita média diária ou custo diário (USD)	Receita média por hora ou custo por hora (USD)
UEP1	12.866,72	579.002,37	24.125,10
UEP2	40.304,02	1.813.680,75	75.570,03
UEP3	51.779,14	2.330.061,33	97.085,89
UEP4	45.297,67	2.038.395,35	84.933,14
UEP5	21.327,54	959.739,15	39.989,13
UEP6	3.201,91	144.085,73	6.003,57
UEP7	42.896,79	1.930.355,64	80.431,49
UEP8	5.441,84	244.882,86	10.203,45
Sonda1	-	1.250.000,00	52.083,33
Sonda 2	-	1.000.000,00	41.666,67
Sonda 3	-	750.000,00	31.250,00

A Tabela 3.10 aponta os custos que seriam incorridos caso a produção dos *clusters* de UEPs não fosse realizada. Isso significa a perda total da receita que seria gerada durante o período de parada da produção. Porém, na realidade, caso ocorra uma parada, a receita não deixa de ser realizada e sim tem seu exercício atrasado. Uma parada compreende o intervalo de tempo em que um *cluster* atinge nível de estoque zero de óleo diesel e o momento em que esse *cluster* recupera nível de estoque positivo, estabelecendo então o período de parada de suas atividades. Dessa forma, para o caso de *clusters* de UEPs, considera-se como custo de falta o custo do atraso do recebimento da receita relativo à duração da parada, a uma taxa de atratividade do capital de 10% ao ano em regime de juros compostos. Assim, a cada parada ocorrida durante a simulação, é associado um custo que varia em função da duração da parada. Como a duração de uma parada pode estar na escala de horas, optou-se por utilizar regime de juros contínuos. No regime de capitalização contínua, existe pagamento de juros a cada período infinitesimal de tempo. Com isso o capital cresce continuamente no tempo à taxa de juros instantânea (BM&FBOVESPA, 2015). Dessa forma, transforma-se a taxa de juros compostos para juros contínuos de acordo com a Equação (3.2):

$$r = \ln(1 + i) \quad (3.2)$$

onde r é a taxa de juros instantânea e i é a taxa de juros composta.

Considerando a taxa de juros composta como 10% ao ano, a taxa de juros contínua equivalente é dada pela Equação (3.3).

$$r = \ln(1 + 0,1) = 0,0953 \text{ a. a} \quad (3.3)$$

Como a unidade do relógio da simulação é em horas, obteve-se a taxa de juro contínua i equivalente à hora, conforme a Equação (3.4).

$$i = \frac{0,0953}{8760} \cong 0,00001088 \text{ a. h} \quad (3.4)$$

O valor futuro do capital aplicado a uma taxa de juro contínua é dado pela Equação (3.5), onde FV é o valor futuro, PV , o valor presente, i é a taxa de juro e t é o período de capitalização.

$$FV = PV \times e^{it} \quad (3.5)$$

Portanto, a remuneração obtida pela aplicação do capital é dada pela Equação (3.6).

$$\text{Remuneração} = PV \times (e^{it} - 1) \quad (3.6)$$

A Equação (3.12) é utilizada para calcular tanto o custo de estoque, que corresponde ao capital imobilizado quando o custo de falta, relacionado à parada de produção, no caso das UEPs. Estes cálculos são apresentados com detalhe na Seção 5.3.

Os demais dados de entrada são parâmetros passados para o simulador que funcionam como chaves de ativação de código. Esses parâmetros informam o tipo de política que se deseja simular, o experimento (com ou sem custo de exposição ao risco), a utilização ou não do otimizador externo de alocações de PSVs, o tempo de aquecimento, o tempo de simulação e o intervalo de troca de turma.

3.4 Modelagem Conceitual

Para melhor entender e interpretar as regras operacionais é importante elaborar o modelo conceitual do problema estudado. Nos próximos tópicos serão descritos as regras operacionais de cada política e seus fluxogramas.

3.4.1 Política de entregas sob demanda

Na política de entregas sob demanda, o gatilho do sistema é disparado a partir das colocações de pedidos dos *clusters*. Tais pedidos são realizados sempre que o estoque de óleo diesel de algum *cluster* cai para seu ponto de pedido. Quando esse evento ocorre, o *cluster* dispara um pedido de uma certa quantidade de óleo diesel para a central de programação. Por sua vez, a central de programação aloca um PSV oleiro para realizar o atendimento ao *cluster*. Nesta política não existe uma rota pré-definida de atendimento para os PSVs oleiros. A programação dos PSVs é realizada na medida em que estes PSVs se tornam disponíveis na bacia e tenham volume em estoque suficiente para atender aos *clusters*. Pode-se dizer que a função do programador central é alocar PSVs disponíveis para atender aos *clusters* que solicitaram atendimento de

acordo com o estoque dos PSVs, dos volumes pedidos dos *clusters* e do cronograma de troca de turma dos PSVs. O intervalo de troca de turma dos PSVs é de 28 dias.

A Figura 3.5 ilustra o fluxograma das regras operacionais da política sob demanda. O fluxograma foi construído sob a perspectiva individual de um PSV oleeiro.

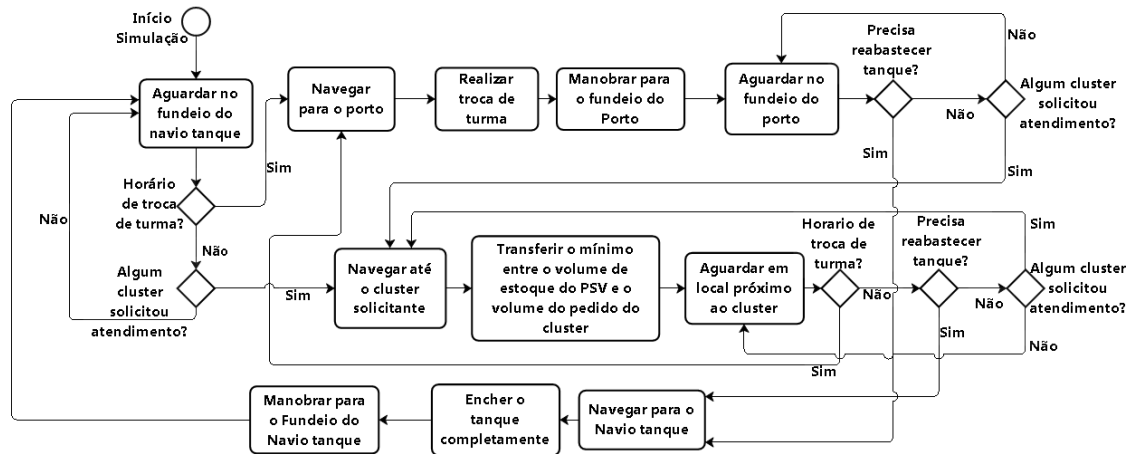


Figura 3.5 – fluxograma política sob demanda

O ponto de pedido do *cluster* i é determinado a partir do *lead-time* médio, do consumo médio do *cluster* i e do seu estoque de segurança, de acordo com a Equação (3.7).

$$Ponto_Pedido_i = (LTM_i * Consumo_Médio_i) + E.Seg_i \quad (3.7)$$

onde LTM_i é o *lead-time* médio do *cluster* i e $E.Seg_i$ compreende o estoque de segurança do *cluster* i .

O *lead-time* médio foi definido a partir da média das diferenças entre as datas de realização dos pedidos e as datas de entrega, oriundas da planilha de registros de atendimentos do período de um ano cedida pela companhia.

3.4.2 Política de entregas programadas

Na política de entregas programadas os atendimentos aos *clusters* ocorrem de acordo com um cronograma pré-estabelecido. Nesse cronograma, a unidade mínima de programação de atendimento é de uma semana, o equivalente a 168 horas. Para se determinar o intervalo ideal de entregas a um *cluster* i , é dada a seguinte Equação:

$$\text{Intervalo_Entre_Visitas}_i = \left\lfloor \frac{\text{Capacidade_Tanque_PSV}}{\text{Unid_Progr} * \text{Consumo_Médio}_i} \right\rfloor * \text{Unid_Prog} \quad (3.8)$$

De acordo com a equação, o intervalo entre visitas para um *cluster* *i* é função da capacidade do PSV, do consumo médio do *cluster* *i* e da unidade de programação mínima. A equação é dada pelo piso do quociente entre a capacidade do tanque do PSV e o consumo médio do *cluster* *i*. Em linguagem matemática, piso é o arredondamento para baixo de um número real x , e sua notação é dada por $\lfloor x \rfloor$. Portanto, o cronograma de atendimentos é planejado de maneira que não ocorra desabastecimento de algum *cluster*, tendo como base de cálculo o consumo esperado de cada *cluster*. Dessa forma, o intervalo entre atendimentos é aquele suficiente para que o *cluster* consuma o equivalente a um tanque cheio de um PSV oleeiro, isto é, 2500 m³.

Considerando a capacidade do tanque do PSV oleeiro de 2500 m³ e o intervalo mínimo de programação de 168 horas, pode-se reescrever a Equação (3.8) como:

$$\text{Intervalo_Entre_Visitas}_i = \left\lfloor \frac{2500}{168 * \text{Consumo_Médio}_i} \right\rfloor * 168 \quad (3.9)$$

O cronograma de atendimento para cada *cluster* pode ser então resumido na Tabela 3.11.

Tabela 3.11 – Cronogramas de atendimento

Cluster	Consumo médio (m ³ /dia)	Consumo médio (m ³ /hora)	Intervalo entre atendimentos (horas)	Intervalo entre atendimentos (semanas)
UEP1	208,21	8,68	168	1
UEP2	309,27	12,89	168	1
UEP3	259,25	10,81	168	1
UEP4	263,39	10,97	168	1
UEP5	145,41	6,06	336	2
UEP6	22,24	0,93	2688	16
UEP7	43,63	1,82	1344	8
UEP8	60,33	2,51	840	5
Sonda1	164,29	6,85	336	2
Sonda2	81,51	3,40	672	4
Sonda3	70,01	2,92	840	5

Na política de entregas programadas, o PSV não realiza viagens entre *clusters*. As viagens sempre são realizadas entre o navio tanque e *cluster* de atendimento. Mesmo que o PSV não tenha transferido todo seu conteúdo em estoque, o mesmo retorna para o navio tanque para reabastecer no fim de cada atendimento.

A Figura 3.6 ilustra o fluxograma das regras operacionais da política de entregas programadas sob a perspectiva do elemento PSV.

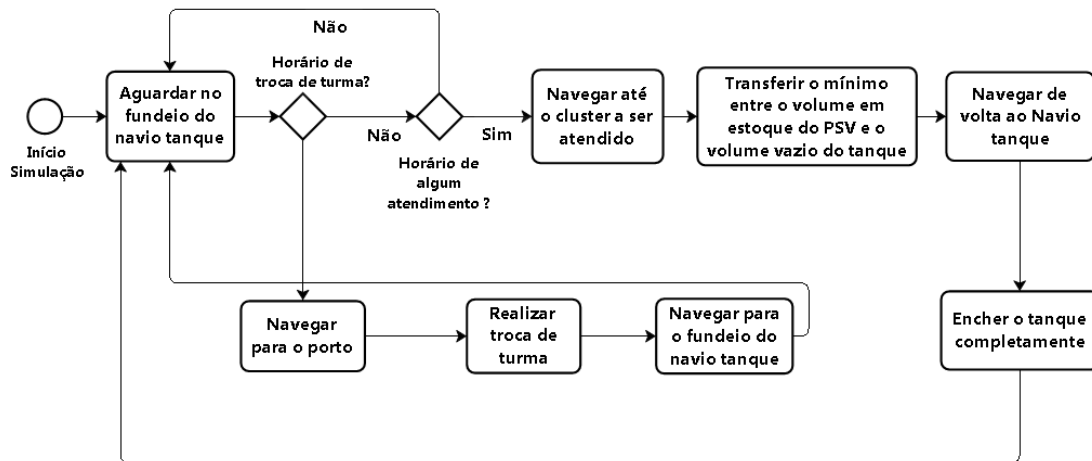


Figura 3.6 – fluxograma política de entregas programadas

3.5 Regras de Alocação de PSVs a *Clusters*

As regras operacionais da política de entregas sob demanda requer a seleção das embarcações que atenderão aos *clusters* solicitantes. No simulador, a regra seguida para atendimento aos *clusters* é a regra FIFO (*first-in-first-out*). Assim, à medida que um PSV oleeiro se torna ocioso ele verifica quais *clusters* dispararam pedidos e atende ao pedido mais antigo. No entanto, essa regra de alocação pode não ser a ótima, pois um *cluster* com pedido mais recente pode possuir uma autonomia menor, ou seja, pode ser prioritário. Dessa forma, foram desenvolvidas rotinas externas ao simulador que utilizam formulação de problema de programação inteira mista para otimizar a alocação dos PSVs aos *clusters*. Essas rotinas de programação são chamadas a cada 24 horas de simulação e exportam as informações do conjunto de PSVs ociosos no momento atual da simulação e o conjunto de *clusters* solicitantes. A rotina externa de programação recebe os dados exportados pelo simulador e retorna a solução ótima de alocação segundo os critérios definidos nas funções objetivo. Foram experimentadas duas formulações diferentes de alocação.

As duas formulações abordadas utilizam os mesmos parâmetros e as mesmas variáveis de decisão. O problema é classificado com um problema de programação inteira mista de alocação e para isso decide sobre dois conjuntos, um de PSVs e outro de *clusters*. Os conjuntos, parâmetros e variáveis de decisão são descritos a seguir.

Conjuntos e índices:

B, i : PSVs
 C, j : *Clusters*

Parâmetros:

EA_j Estoque atual do *cluster* j (m³)
 C_j Taxa média de consumo do *cluster* j (m³/h)
 TN_{ij} Tempo de navegação do PSV i até o *cluster* j (horas)
 S_j Estoque de segurança do *cluster* j (m³)
 Q_j Quantidade demandada do *cluster* j (m³)
 EB_i Estoque atual do PSV i (m³)

Variáveis de decisão:

x_{ij} Variável binária que indica se o PSV i atende o *cluster* j
 DR_j Demanda não atendida do *cluster* j

3.5.1 Formulação de minimização do risco do consumo do estoque de segurança

Esta formulação também pode ser interpretada como minimizar a diferença entre o nível do estoque no momento do atendimento ao *cluster* e seu estoque de segurança. Dessa forma, a regra de alocação dos PSVs aos *clusters* é dada pela seguinte formulação:

$$\text{Min} \sum_{i \in B} \sum_{j \in C} (EA_j + C_j \times TN_{ij} - S_j) \times x_{ij} + \alpha \sum_{j \in C} DR_j \tag{3.10}$$

s.a

$$\sum_{j \in C} (Q_j x_{ij} - DR_j) \leq EB_i \quad \forall i \in B \quad (3.11)$$

$$\sum_{i \in B} (EB_i x_{ij}) \geq Q_j - DR_j \quad \forall j \in C \quad (3.12)$$

$$\sum_{i \in B} x_{ij} \leq 1 \quad \forall j \in C \quad (3.13)$$

$$\sum_{j \in C} x_{ij} \leq 1 \quad \forall i \in B \quad (3.14)$$

$$\sum_{i \in B} \sum_{j \in C} x_{ij} \geq 1 \quad (3.15)$$

$$x_{ij} \in \{0,1\} \quad \forall i \in B, \forall j \in C \quad (3.16)$$

De acordo com formulação, a Função Objetivo (3.10) minimiza o somatório das diferenças entre os níveis de estoque no momento da entrega ao *cluster* e seus respectivos estoques de segurança assim como o somatório das demandas não atendidas dos *clusters*. A parcela $(C_j \times TN_j)$ entra como soma na função objetivo para que seja escolhido o PSV mais próximo do cluster, nos casos em que haja mais de um PSV disponível. Dessa forma, minimiza-se o volume de óleo diesel consumido pelo *cluster* durante o tempo de navegação. O fator α que multiplica o segundo somatório pode ser calibrado de acordo com a penalidade a que se deseja submeter pelo não atendimento das demandas. As Restrições (3.11) garantem que a demanda total alocada a cada PSV não exceda seu respectivo nível de estoque. As restrições (3.12) procuram alocar toda a demanda do *cluster* aos barcos que irão atendê-lo. As Restrições (3.7) garantem que um barco só atenda um único cliente. As restrições (3.8) impõem a condição que um cliente só pode ser atendido por um único barco. As Restrições (3.9) forçam que pelo menos uma alocação seja realizada, impedindo ocorrências de soluções vazias. Por fim, as Restrições (3.10) definem o domínio das variáveis de decisão como binárias.

3.5.1.1 Exemplo numérico

A Figura 3.7 ilustra a variação do nível de estoque dos *clusters* UEP1 e UEP2 ao longo de 8760 horas de simulação do cenário de 10 PSVs. De acordo com a formulação, os atendimentos devem ser realizados para minimizar o risco de consumo do estoque de segurança. Para o cenário de 10 PSVs, o nível de estoque estabiliza em torno do ponto

de pedido e se mantém acima do estoque de segurança, quase 100% do tempo, isto é, $EA_i > S_i$. Nesse cenário procura-se priorizar o atendimento do *cluster* que apresenta um nível de estoque “menos positivo” em relação ao estoque de segurança.

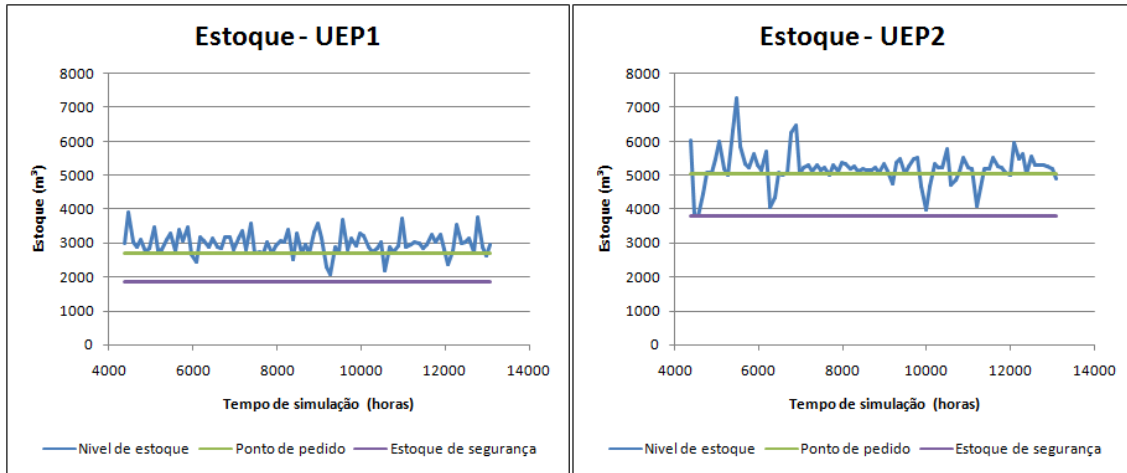


Figura 3.7 – Níveis de estoque acima do estoque de segurança

A Figura 3.8 compara dois *clusters* em uma situação hipotética no momento de pedido. Considerando a hora 0 como o momento do pedido, e a taxa de consumo média por hora dos *clusters* o tempo de navegação TN para cada *cluster* e supondo que os dois *clusters* serão atendidos, mostra a curva do nível de estoque e dos estoques de segurança.

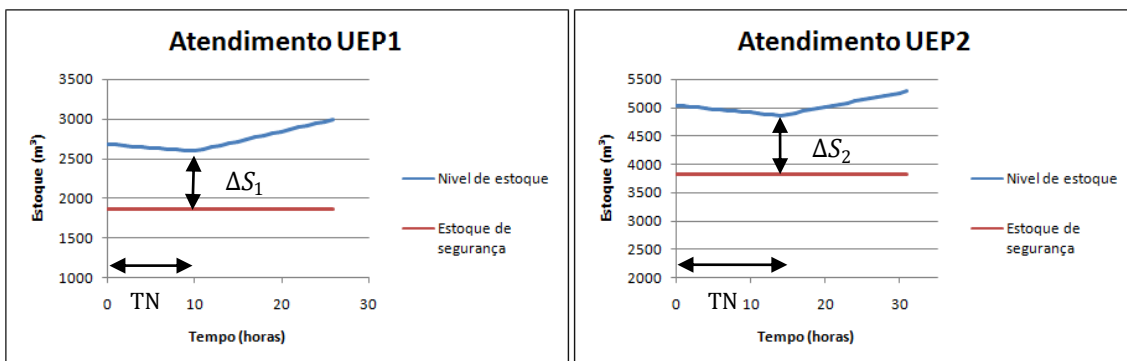


Figura 3.8 – Estado de dois clusters no momento do pedido

No momento da entrega, a diferença entre o nível de estoque e o estoque de segurança do *cluster* UEP1 é $\Delta S_1 = 732 \text{ m}^3$ e a diferença entre o nível de estoque do *cluster* UEP2 é $\Delta S_2 = 1044.4 \text{ m}^3$. Dessa forma $\Delta S_1 < \Delta S_2$, logo o *cluster* que deve ser selecionado para atendimento caso não haja recursos suficiente é o *cluster* UEP1.

Portanto, em um cenário em que os níveis de estoque no momento da entrega são superiores ao estoque de segurança, a formulação prioriza o *cluster* cujo estoque é o menos positivo em relação ao estoque de segurança.

Por outro lado, em um cenário em que os níveis de estoque dos *clusters* são inferiores a seus respectivos estoques de segurança, isto é, $EA_i < S_i$, a formulação visa atender aqueles que estão “mais negativos” em relação ao estoque de segurança. Na Figura 3.9 pode-se observar a variação do nível de estoque dos *clusters* UEP1 e UEP2 ao longo de 3200 horas de simulação para o cenário de 2 PSVs.

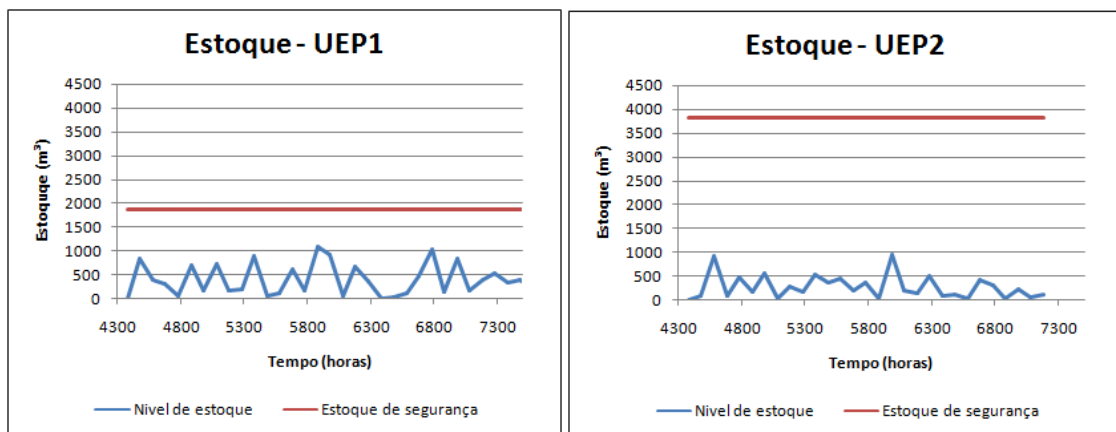


Figura 3.9 – Níveis de estoque abaixo do estoque de segurança

A Figura 3.10 ilustra outra situação hipotética em que os níveis de estoque de dois *clusters* encontram-se abaixo do estoque de segurança. Nesse cenário se houver apenas um barco disponível para atendimento, o *cluster* selecionado será o *cluster* cujo nível de estoque no momento da entrega for mais “negativo” em relação ao estoque de segurança. No exemplo, também considera-se o tempo de navegação do PSV até o cluster UEP1 de 10 horas e o tempo de navegação do PSV até o cluster UEP2 de 15 horas.

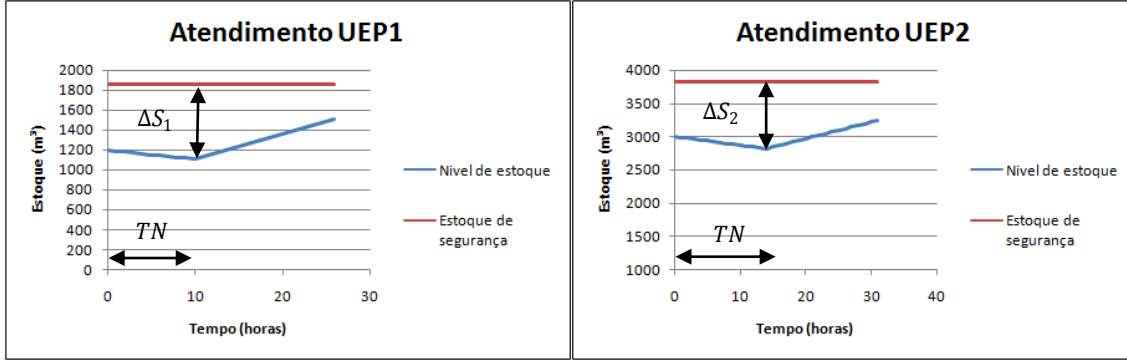


Figura 3.10 – Estado de dois clusters no momento do pedido

Na Figura 3.10 a diferença entre o estoque no momento da entrega e o estoque de segurança do cluster UEP1 é $\Delta S_1 = -748 \text{ m}^3$ e a diferença entre o estoque no momento da entrega do cluster UEP2 é $\Delta S_2 = -995.6 \text{ m}^3$. Logo $\Delta S_2 < \Delta S_1$ e o cluster UEP2 deve ser priorizado caso não haja recursos suficiente.

3.5.2 Formulação de maximização da autonomia dos clusters

Esta formulação propõe priorizar o atendimento aos clusters que possuem menor tempo para operar até esgotarem seus estoques.

$$\text{Min} \sum_{i \in B} \sum_{j \in C} \left(\frac{EA_j + C_j \times TN_{ij}}{C_j} \right) \times x_{ij} + \alpha \sum_{j \in C} \frac{DR_j}{C_j} \quad (3.17)$$

s.a

$$\sum_{j \in C} (Q_j x_{ij} - DR_j) \leq EB_i \quad \forall i \in B \quad (3.18)$$

$$\sum_{i \in B} (EB_i x_{ij}) \geq Q_j - DR_j \quad \forall j \in C \quad (3.19)$$

$$\sum_{i \in B} x_{ij} \leq 1 \quad \forall j \in C \quad (3.20)$$

$$\sum_{j \in C} x_{ij} \leq 1 \quad \forall i \in B \quad (3.21)$$

$$\sum_{i \in B} \sum_{j \in C} x_{ij} \geq 1 \quad (3.22)$$

$$x_{ij} \in \{0,1\} \quad \forall i \in B, \forall j \in C \quad (3.23)$$

De acordo com a formulação, a Função Objetivo (3.17) procura minimizar o somatório da razão entre o nível de estoque atual adicionado do volume consumido até a entrega e o consumo médio do *cluster* a ser atendido. Essa medida é expressa em horas e representa a autonomia do *cluster*, isto é, o tempo restante previsto até o esgotamento de seu estoque. A segunda parcela da função objetivo consiste no somatório do quociente entre a demanda não atendida e o consumo médio do *cluster*, que significa o tempo previsto de consumo da demanda não atendida. Da mesma forma que na primeira formulação, o fator α pode ser calibrado de acordo com a penalidade a que se deseja submeter a função objetivo pelo não atendimento da demanda dos *clusters*. A parcela $(C_j \times TN_{ij})$ entra como soma na função objetivo para que seja escolhido o PSV mais próximo ao cluster, quando há mais de um PSV disponível. Dessa forma, é minimizado o volume consumido de óleo diesel pelo *cluster* durante o tempo de navegação. As Restrições (3.18) garantem que a quantidade total de óleo diesel a ser entregue pelo PSV não exceda o seu estoque. As Restrições (3.19) almejam que a quantidade total de óleo diesel a ser entregue em cada *cluster* seja a mais próxima da quantidade demandada. Essa restrição não garante que toda a demanda seja atendida, pois é subtraída da parcela de demanda não atendida. Porém a demanda não atendida é penalizada na função objetivo, o que força sua minimização. As Restrições (3.20) impõem que cada *cluster* seja atendido por um único PSV e as Restrições (3.21) impõem que cada PSV atenda no máximo um *cluster*. As Restrições (3.22) garantem que ao menos uma alocação seja feita, impedindo soluções vazias e, por fim as Restrições (3.23) definem o domínio das variáveis binárias.

Os resultados obtidos nos experimentos de simulação com utilização das regras de alocação externas de PSVs aos *clusters* descritas nesta seção serão apresentados na Seção 5.6.

4 Implementação Computacional

O passo seguinte à construção do modelo conceitual é a tradução deste em linguagem computacional. Para isso, foi utilizada a simulação a eventos discretos. Tal ferramenta viabiliza o experimento de cenários hipotéticos (nos quais se varia algum parâmetro de entrada) e permite considerar as incertezas inerentes ao sistema que se deseja modelar.

4.1 Software Utilizado

Para traduzir o modelo conceitual para o modelo computacional foi utilizado o *software* Promodel®, que consiste em um *software* de simulação a eventos discretos com uma abordagem de programação por processos. Ele proporciona uma boa flexibilidade ao programador, pois permite criar lógicas de processos para as entidades criadas em cada local construído no modelo. O *software* dispõe de uma linguagem própria de programação e também permite a comunicação com arquivos externos tipo planilhas do MS-Excel® para importação ou exportação de dados. Além disso, oferece uma interface gráfica e permite criar uma biblioteca própria de imagens. Para o modelo, foi escolhido usar uma imagem de fundo azul em referência ao mar e as unidades marítimas compreendem desenhos de plataformas de exploração e produção de petróleo. O Promodel® também possui funções já instaladas capazes de discretizar processos contínuos como operações de consumo e transferência de fluidos através do elemento “tanque”. A Figura 4.1 ilustra a interface gráfica do *software*.

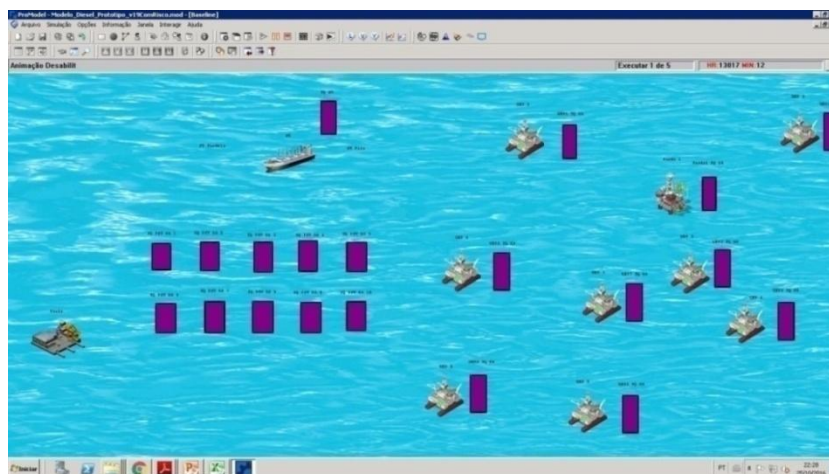


Figura 4.1 – Interface gráfica do promodel

4.2 Estrutura de Modelagem

Os elementos de modelagem compreendem as entidades, os locais, os processos, os recursos, os elementos auxiliares, como variáveis, matrizes, sub-rotinas e macros, além de uma estrutura denominada “rede de caminho” que interliga os locais do modelo. A seguir são descritos todos os elementos utilizados para construir o modelo.

4.2.1 Locais

Os locais no modelo representam regiões físicas por onde as entidades transitam e executam as lógicas de programação. Na construção dos locais é definida sua capacidade – número máximo de entidades que podem ser processadas ao mesmo tempo – assim como a lógica do processo vinculado a eles. A Tabela 4.1 descreve os principais locais construídos no modelo.

Tabela 4.1 – Descrição dos locais

Local	Descrição
Porto	Local onde os PSVs oleeiros realizam a troca de tripulação
NT	Navio tanque onde os PSVs oleeiros reabastecem seus tanques. Possui capacidade de processar até 3 entidades ao mesmo tempo.
NT_Fundeio	Região próxima ao NT em que os PSVs oleeiros permanecem fundeados aguardando alguma ordem de atendimento.
NT_Fila	Região onde os PSVs oleeiros aguardam liberação de capacidade do NT caso o mesmo esteja lotado
UEP_1	Região que representa o <i>cluster</i> UEP1, para onde os PSVs oleeiros navegam caso haja pedido ou viagem programada.
UEP_2	Região que representa o <i>cluster</i> UEP2, para onde os PSVs oleeiros navegam caso haja pedido ou viagem programada.
UEP_3	Região que representa o <i>cluster</i> UEP3, para onde os PSVs oleeiros navegam caso haja pedido ou viagem programada.
UEP_4	Região que representa o <i>cluster</i> UEP4, para onde os PSVs oleeiros navegam caso haja pedido ou viagem programada.
UEP_5	Região que representa o <i>cluster</i> UEP5, para onde os PSVs oleeiros navegam caso haja pedido ou viagem programada.
UEP_6	Região que representa o <i>cluster</i> UEP6, para onde os PSVs oleeiros navegam caso haja pedido ou viagem programada.
UEP_7	Região que representa o <i>cluster</i> UEP7, para onde os PSVs oleeiros navegam caso haja pedido ou viagem programada.
UEP_8	Região que representa o <i>cluster</i> UEP8, para onde os PSVs oleeiros navegam caso haja pedido ou viagem programada.
Sonda_1	Região que representa o <i>cluster</i> Sonda1, para onde os PSVs oleeiros navegam caso haja pedido ou viagem programada.
Sonda_2	Região que representa o <i>cluster</i> Sonda2, para onde os PSVs oleeiros navegam caso haja pedido ou viagem programada.
Sonda_3	Região que representa o <i>cluster</i> Sonda3, para onde os PSVs oleeiros navegam caso haja pedido ou viagem programada.
SetUp	Região imaginária do modelo utilizada para executar processos de segundo plano, como ordem de consumo dos <i>clusters</i> , ordens de pedidos e atendimentos.

4.2.1.1 O elemento “tanque”

O Promodel® dispõe de um elemento de modelagem “tanque” que é responsável por descrever comportamentos de eventos contínuos como consumo e transferência de fluidos. Tais processos são discretizados a partir de funções pré programadas do *software*. O passo de atualização dos processos contínuos de operações com fluidos adotado como configuração padrão é 0.2 horas, o equivalente a 12 minutos, mas este passo pode ser reconfigurado pelo programador. O elemento “tanque” é representado como um local no modelo. Dessa forma, foram construídos 22 locais “tanque”: 11 vinculados aos *clusters*, 1 ao navio tanque e 10 aos PSVs oleiros. A Tabela 4.2 descreve os elementos “tanque” e os respectivos locais a que estão vinculados.

Tabela 4.2 – Descrição do locais “tanque”

Local “tanque”	Descrição
UEP1_TQ_OD	Tanque correspondente ao <i>cluster</i> UEP1
UEP2_TQ_OD	Tanque correspondente ao <i>cluster</i> UEP2
UEP3_TQ_OD	Tanque correspondente ao <i>cluster</i> UEP3
UEP4_TQ_OD	Tanque correspondente ao <i>cluster</i> UEP4
UEP5_TQ_OD	Tanque correspondente ao <i>cluster</i> UEP5
UEP6_TQ_OD	Tanque correspondente ao <i>cluster</i> UEP6
UEP7_TQ_OD	Tanque correspondente ao <i>cluster</i> UEP7
UEP8_TQ_OD	Tanque correspondente ao <i>cluster</i> UEP8
Sonda1_TQ_OD	Tanque correspondente ao <i>cluster</i> Sonda1
Sonda2_TQ_OD	Tanque correspondente ao <i>cluster</i> Sonda2
Sonda3_TQ_OD	Tanque correspondente ao <i>cluster</i> Sonda3
TQ_PSV_OD_1	Tanque correspondente ao PSV 1
TQ_PSV_OD_2	Tanque correspondente ao PSV 2
TQ_PSV_OD_3	Tanque correspondente ao PSV 3
TQ_PSV_OD_4	Tanque correspondente ao PSV 4
TQ_PSV_OD_5	Tanque correspondente ao PSV 5
TQ_PSV_OD_6	Tanque correspondente ao PSV 6
TQ_PSV_OD_7	Tanque correspondente ao PSV 7
TQ_PSV_OD_8	Tanque correspondente ao PSV 8
TQ_PSV_OD_9	Tanque correspondente ao PSV 9
TQ_PSV_OD_10	Tanque correspondente ao PSV 10
TQ_NT	Tanque correspondente ao navio tanque

4.2.2 Entidades

As entidades são elementos individualmente processados no sistema e que mudam de estado durante a simulação. Elas chegam ao sistema através de alguma regra definida e executam os blocos de programação durante o tempo de simulação. As entidades podem ser temporárias, caso existam por um tempo limitado no sistema ou

permanentes, caso nunca saem do sistema até o final do tempo simulado. No modelo existem 4 tipos de entidades “ePSV_OD”, eSetUp_Consumo”, “ePedido_OD” e “eAtendimento”. A Tabela 4.3 apresenta as entidades do modelo e suas descrições.

Tabela 4.3 – Descrição das entidades

Entidade	Tipo	Descrição
ePSV_OD	Permanente	Esta entidade chega ao sistema na hora zero da simulação e representa o elemento PSV oleeiro. Ela executa os códigos dos processos vinculados aos locais do modelo e é transportada através destes locais.
eSetUp_consumo	Temporária	A entidade “eSetUp_consumo” possui infinitas chegadas no modelo em intervalos fixos de uma hora. Elas chegam em grupo de 11 e enviam a ordem de consumo dos tanques para cada <i>cluster</i> .
eAtendimento	Temporária	A chegada da entidade “eAtendimento” no modelo segue o intervalo de tempo definido no cronograma da política de entregas programadas. A chegada ocorre no local “SetUp” e possui a função de indicar a programação de uma nova viagem.
ePedido_OD	Temporária	Esta entidade é criada no local “SetUp” sempre que um <i>cluster</i> realiza um pedido através da sub rotina “sub_geracao_pedidos”, que será apresentada na Seção 4.2.6.4. Possui função de indicar um novo pedido e seu volume.

4.2.3 Recursos

Recursos são elementos usados para transportar entidades através dos locais do modelo. Estes recursos podem ser estáticos, se não possuem uma rede de caminhos associada ou dinâmicos, se possuem uma rede de caminhos associada. A estrutura “rede de caminhos” é explicada na Seção 4.2.5. Além disso, é possível criar regras de paradas programadas e não programadas para os recursos. O modelo conta com um único recurso denominado “rPSV”. Tal recurso é dinâmico e se locomove através da rede de caminho “Rede_Mar” que interliga os locais do modelo. A modelagem das regras de troca de tripulação e inoperância é implementada através de paradas programadas e não programadas do recurso “rPSV”.

4.2.4 Processos

Os processos são blocos de código vinculados aos locais e que instruem a programação a ser seguida pelas entidades que passam naquele local. Eles representam as regras operacionais descritas no modelo conceitual. Nos processos existe uma estrutura complementar chamada de bloco de direcionamento que vincula os possíveis locais para os quais as entidades podem ser direcionadas após a execução da lógica

descrita no local atual. A Figura 4.2 mostra a interface de programação dos processos vinculados aos locais do modelo e as orientações das entidades através dos locais. As setas compreendem os possíveis locais para onde as entidades podem ser conduzidas, a partir do local em que estão. Dessa forma, para cada local do modelo existe um processo e um bloco de direcionamento associado.

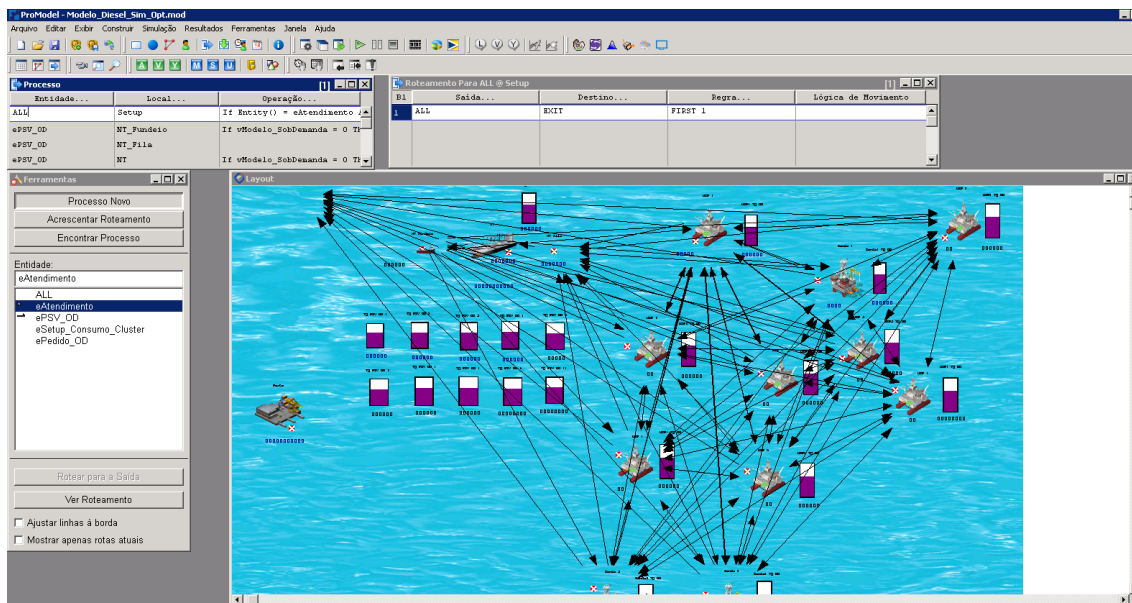


Figura 4.2 – Processos e blocos de direcionamento

As setas representadas na interface gráfica podem ser melhor descritas pela Tabela 31. Nela são apresentados todos os locais onde existe bloco de processo associado, a entidade que executa a lógica do processo e os possíveis locais para os quais as entidades podem ser direcionadas após serem processadas no local vigente. As entidades “eSetup_Consumo”, “eAtendimento” e “ePedido_OD” são temporárias, portanto saem do sistema. Já a entidade “ePSV_OD” permanece no sistema até o fim da simulação. Além disso, vale lembrar que na política de entregas sob demanda, a entidade “ePSV_OD” pode partir de qualquer *cluster* para qualquer outro, já na política de entregas programadas após visitar um *cluster*, a entidade “ePSV_OD” só poderá ser direcionada ao local “NT_Fila”, onde espera para reabastecer.

A Tabela 31 está disposta no Apêndice D – Blocos de Processos e Direcionamento e todos os códigos dos processos descritos nesta tabela podem ser consultados no Apêndice E – Códigos do Simulador.

4.2.5 Rede de caminhos

O Promodel® permite criar redes de caminhos que conectam os locais construídos no modelo. Dessa forma, associa-se um nó a cada local mapeado e interligam-se esses nós por arestas. A rede de caminho então é usada pelos recursos do modelo. Pode-se definir uma rede de caminhos por tempo ou por distância e velocidade. No primeiro caso, atribui-se um tempo que o recurso gasta para percorrer cada aresta da rede. No segundo caso, atribui-se uma distância entre os nós e uma velocidade para os recursos. No modelo foi escolhido utilizar a rede de caminhos orientada por tempos, denominada “Rede_Mar”. Os tempos representam os tempos de navegação, considerando a matriz de distância entre os centros geométricos dos *clusters* e a velocidade média de 15 Km/hora de um PSV oleeiro.

No modelo, o recurso “rPSV” é associado à rede de caminho “Rede_Mar” e transporta as entidades “ePSV_OD” através dos segmentos que conectam os locais. A Figura 4.3 apresenta a interface da rede de caminhos “Rede_Mar”. O nó base da “Rede_Mar” é o local “NT_Fundeio”, que representa o local para onde o recurso “rPSV” deve ser direcionado se estiver ocioso. Na Figura 4.3, os pontos representam os nós e as linhas representam as arestas da rede de caminho.

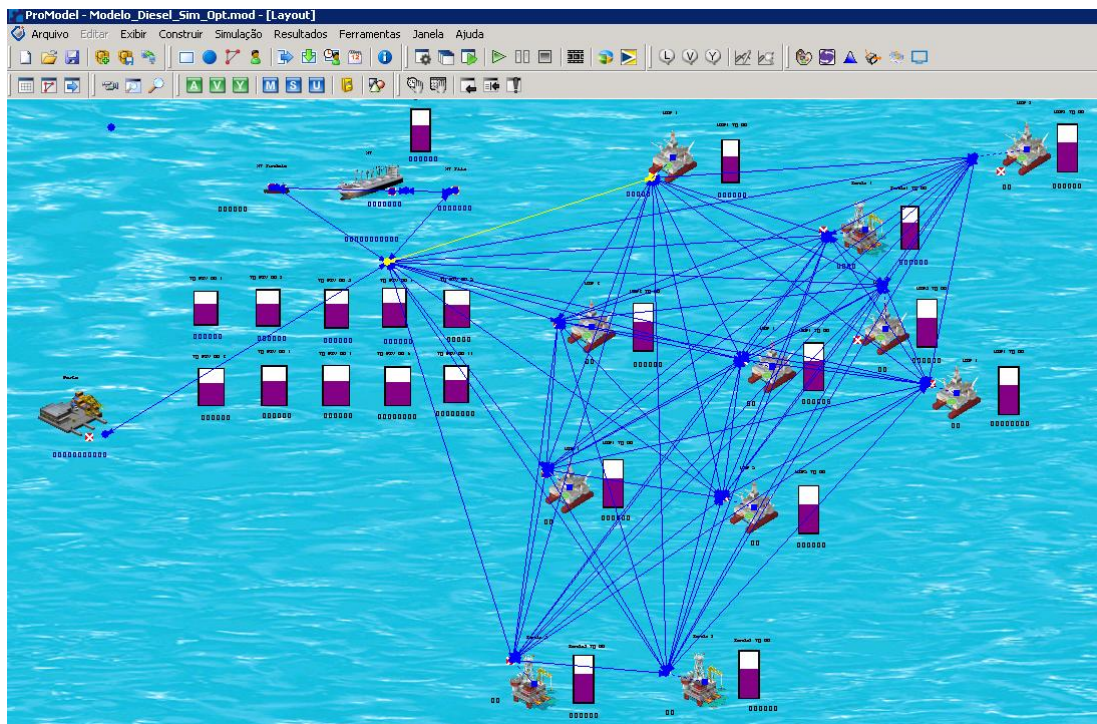


Figura 4.3 – Rede de caminhos “Rede_Mar”

4.2.6 Elementos auxiliares

Atributos, variáveis e matrizes são elementos essenciais para programar o modelo. São essas estruturas que armazenam valores e/ou os atualizam durante a simulação. Variáveis e matrizes atualizáveis possuem seus valores modificados de acordo com os eventos que alteram o estado do sistema. Além desses, as sub rotinas e as macros são estruturas de programação complementares necessárias para construção de funções e lógicas de programação que se repetem ao longo da simulação. As sub-rotinas funcionam como funções e podem retornar ou não algum valor. Já as macros são estruturas de código que podem ser chamadas em qualquer local do modelo, inclusive dentro dos processos.

4.2.6.1 Atributos

Atributos são valores que podem ser associados às entidades ou aos locais. Os atributos podem ter seus valores atualizados durante a simulação e carregam informação dos estados das entidades ou dos locais. A Tabela 4.4 descreve os principais atributos do modelo.

Tabela 4.4 – Atributos

Atributo	Associação	Descrição
aID_Cluster	Entidade	Registra o <i>cluster</i> associado às entidades. Pode ser usado durante o endereçamento das entidades “ePSV_OD” ou na lógica de criação de atendimentos ou pedidos, associados às entidades “eSetUp_Consumo” e “eAtendimento”
aID_PSV_OD	Entidade	Registra o ID do PSV associado às entidades.
aViagem_Complementar	Entidade	Registra se a entidade “ePSV_OD” está realizando uma viagem complementar ao <i>cluster</i> , ou seja, indica se já existe outro PSV no local.
aVolume_a_entregar	Entidade	Registra o volume que deverá ser entregue pela entidade “ePSV_OD” ao chegar no local de destino.
aRetorno_Fundeio	Entidade	Registra se a entidade “ePSV_OD” está chegando pela primeira vez no local “NT_Fundeio” ou se está retornando.
aUltima_Fracao	Entidade	Indica se a entidade “ePSV_OD” está entregando a última fração do pedido ou não. Útil quando são necessárias mais de uma viagem para entregar o mesmo pedido.

4.2.6.2 Variáveis

Variáveis são estruturas de dados que armazenam e/ou atualizam valores ao longo da simulação. Na Tabela 4.5 são apresentadas as principais variáveis de entrada do modelo, aquelas que definem os experimentos que se deseja simular e suas configurações.

Tabela 4.5 – Variáveis

Variável	Descrição
vTempo_Aquecimento	Variável obtida da planilha de entrada do modelo que indica o tempo de aquecimento. O tempo de aquecimento é definido na Seção 5.1.
vTempo_Simulacao	Variável importada da planilha de entrada do modelo que indica o tempo de simulação
vModelo_sobdemanda	Variável obtida da planilha de entrada do modelo que indica se a será executada a lógica da política de entregas sob demanda ou da política de entregas programadas. Assume o valor zero para política de entregas programadas e um para política de entregas sob demanda.
vOtimizador	Variável obtida da planilha de entrada do modelo que indica se serão usadas as regras de alocação externas ou não.
vExperimento	Variável obtida da planilha de entrada do modelo que indica se será rodado o experimento com ou sem custo de exposição ao risco.

4.2.6.3 Matrizes

As matrizes são estruturas de dados que armazenam e/ou atualizam valores durante a simulação. Algumas matrizes são utilizadas para guardar os resultados de interesse do modelo e outras são auxiliares e funcionam como contadores ou *flags* utilizados pelas matrizes principais. Na Tabela 4.6 são destacadas as principais matrizes do modelo.

Tabela 4.6 – Matrizes

Matriz	Descrição
mPedidos_Aguardando[X,Y]	Matriz de dimensão 11x1 que atualiza a quantidade solicitada de óleo diesel de cada <i>cluster</i> em espera para ser entregue. Utilizada na programação da política de entregas sob demanda. As linhas representam os <i>clusters</i> e a coluna, o volume de óleo diesel em espera por atendimento.
mAtendimentos_Aguardando[X,Y]	Matriz de dimensão 11x1 que atualiza a número de viagens programadas para cada <i>cluster</i> no momento presente. Utilizada na programação da política de entregas programadas. As linhas representam os <i>clusters</i> e a coluna o número de viagens programadas ainda e ainda não realizadas.
mPedidos_gerados[X,Y]	Matriz de dimensão 11x1 que acumula os pedidos gerados por cada <i>cluster</i> . Utilizada na programação da política de entregas sob demanda. As linhas representam os <i>clusters</i> e a coluna, a quantidade de pedidos gerados no momento atual.

mEstoque_Transito[X,Y]	Matriz de dimensão 11x1 que atualiza o volume de óleo diesel que está sendo transportado, mas que ainda não foi entregue para cada <i>cluster</i> . As linhas representam os <i>clusters</i> e a coluna, o volume em trânsito.
mStatus_Pedidos[X,Y,Z]	Matriz de 3 dimensões que atualiza a situação atual dos pedidos. Utilizada na programação da política de entregas sob demanda. As linhas representam os pedidos, as colunas as situações dos pedidos (pedido entregue, em transito ou em espera, horário de realização do pedido, horário de entrega e lead-time) e a altura representa os <i>clusters</i> a que se referem os pedidos.
mLead_Time[X,Y]	Matriz de dimensão 10000x11 que grava o tempo decorrido entre a realização do pedido e sua entrega. As linhas representam as entregas e as colunas os <i>clusters</i> em que as entregas foram feitas.
mStatus_Locais[X,Y]	Matriz de dimensão 15x6 que atualiza a situação atual dos locais. As linhas representam os locais, e as colunas a situação (número de PSVs operando no local e número de PSVs ociosos, fundeados no local).
mID_TQ_Cluster[X,Y]	Matriz de dimensão 11x1 carregada na lógica de inicialização que correlaciona os <i>clusters</i> com seus respectivos tanques. As linhas representam o ID do <i>cluster</i> e a coluna, o ID do tanque relacionado.
mID_TQ_PSV_OD[X,Y]	Matriz de dimensão 10x1 carregada na lógica de inicialização que correlaciona os PSVs oleiros com seus respectivos tanques. As linhas representam o ID do PSV e a coluna o ID do tanque relacionado.
mPSV_Selecionado[X,Y]	Matriz de dimensão 10x1 que correlaciona a entidade “ePSV_OD” ao recurso “rPSV”. As linhas representam os IDs dos PSVs oleiros e a coluna representa o ID do recurso “rPSV”.
mVerifica_Barco_TopUp[X,Y]	Matriz de dimensão 11x2 que atualiza para cada <i>cluster</i> a quantidade de PSVs roteados para ele e/ou nele operando e se o tanque do <i>cluster</i> será completamente enchido ou não. As linhas representam os <i>clusters</i> , a primeira coluna indica quantos PSVs oleiros estão operando no local ou navegando para o local e a segunda coluna indica se o tanque será enchido completamente ou não.
mTempo_Ocioso_PSV[X,Y]	Matriz de dimensão 40000x10 que contabiliza o tempo que cada PSV permanece inativo entre duas utilizações consecutivas. As linhas representam as utilizações do PSV e a coluna representa os IDs dos PSVs.
mVolume_Total_entregue[X,Y]	Matriz de dimensão 11x1 que contabiliza o volume total entregue em cada <i>cluster</i> . As linhas representam os <i>clusters</i> e a coluna a soma do volume entregue.
mCont_Entregas[X,Y]	Matriz de dimensão 11x1 que contabiliza o total de entregas efetuadas em cada <i>cluster</i> . As linhas representam os <i>clusters</i> e a coluna representa a soma da quantidade de entregas.
mVolume_Consumido_Cluster[X,Y]	Matriz de dimensão 11x1 que contabiliza o volume

	total de óleo diesel consumido por cada <i>cluster</i> .
mTempo_Ciclo[X,Y]	Matriz de dimensão 10000x10 que grava o tempo de ciclo de cada viagem do PSV. As linhas representam os ciclos e as colunas os PSVs.
mEntrega_PSV[X,Y]	Matriz de dimensão 10x1 que contabiliza a quantidade de entregas realizadas por cada PSV.
mNumero_Visitas_FundeioNT[X,Y]	Matriz de dimensão 10x1 que contabiliza a quantidade de retornos ao local “NT_Fundeio”, que representa número de ciclos realizados por cada PSV.
mIntervalo_Entre_Visitas[X,Y]	Matriz de dimensão 10000x11 que grava o intervalo de tempo entre duas visitas de PSV consecutivas em cada <i>cluster</i> .
mCont_Estoque_Risco[X,Y]	Matriz de dimensão 11x1 que contabiliza o número de vezes em que o nível de estoque de um cluster baixou do nível de segurança.
mCont_StockOut[X,Y]	Matriz de dimensão 11x1 que contabiliza o número de vezes que houve esgotamento total do estoque do <i>cluster</i> .
mDuracao_StockOut[X,Y]	Matriz de dimensão 10000x11 que grava a duração de cada ocorrência de falta de estoque. As linhas representam as ocorrências de falta de estoque e as colunas representam os <i>clusters</i> em que ocorreram as faltas.
mDuracao_Estoque_Risco[X,Y]	Matriz de dimensão 10000x11 que grava a duração correspondente a cada ocorrência de nível de estoque abaixo do estoque de segurança. As linhas representam as ocorrências e as colunas, os <i>clusters</i> onde houve as ocorrências.
mCusto_Operacional[X,Y]	Matriz de dimensão 10x1 que contabiliza o custo operacional dos PSVs. As linhas representam os PSVs e a coluna a soma do custo operacional
mCusto_Falta[X,Y]	Matriz de dimensão 11x1 que contabiliza o custo de falta relativo a cada <i>cluster</i> . As linhas representam os clusters e a coluna a soma do custo de falta.
mCusto_Estoque[X,Y]	Matriz de dimensão 11x1 que contabiliza o custo de estoque relativo a cada <i>cluster</i> . As linhas representam os clusters e a coluna a soma do custo de estoque.
mNivel_servico_operacional[X,Y]	Matriz de dimensão 11x1 que armazena o nível de serviço operacional de cada <i>cluster</i> calculado na lógica de término.
mNivel_servico_seguranca[X,Y]	Matriz de dimensão 11x1 que armazena o nível de serviço de segurança de cada <i>cluster</i> calculado na lógica de término.

4.2.6.4 Sub rotinas

As sub rotinas são blocos de código que podem ser chamados em qualquer processo dentro da simulação ou programados para serem executados em algum intervalo de tempo da simulação. Elas podem receber parâmetro e retornar valores, funcionando como funções ou somente executar um processo de programação. A Tabela 4.7 descreve as principais sub rotinas do modelo.

Tabela 4.7 – Sub rotinas

Sub rotina	Descrição
Sub_seleciona_barco	Sub rotina responsável por executar a lógica de seleção do PSV. Ela é chamada no processo do local “NT_Fundeio” e na macro “mac_processo_geral_sob_demanda”. Faz parte da programação da política de entregas sob demanda.
Sub_seleciona_cluster	Sub rotina responsável por selecionar o <i>cluster</i> a ser atendido pelo PSV oleeiro. Chamada nos processos do local “NT_Fundeio” e nos processos programados nos <i>clusters</i> .
Sub_gera_pedidos	Sub rotina responsável por verificar frequentemente o nível dos tanques dos <i>clusters</i> e realizar pedidos caso esse nível atinja o ponto de pedido. Ativada na programação da política de entregas sob demanda.
Sub_atualiza_consumo	Sub rotina executada de hora em hora, responsável por atualizar a taxa de consumo dos <i>clusters</i> .
Sub_programacao_diaria	Sub rotina responsável por exportar os dados dos <i>clusters</i> solicitantes e dos PSVs oleiros ociosos a cada 24 horas de simulação. Acionada quando é utilizada regra de alocação externa ao simulador.
Sub_exporta_barcos	Sub rotina responsável por exportar os locais dos PSVs e os seus níveis de estoque para arquivo csv. Acionada dentro da sub rotina “sub_programacao_diaria”
Sub_exporta_clusters	Sub rotina responsável por exportar informações dos <i>clusters</i> como nível de estoque, volume do pedido e estoque de segurança para arquivo csv. Acionada dentro da sub rotina “Sub_programacao_diaria”.
Sub_trata_solucão_otimizador	Sub rotina responsável por organizar na estrutura de lógica do simulador a solução importada da regra alocação externa. Acionada quando é utilizada a regra de alocação externa.

Os códigos das principais sub rotinas podem ser consultados no Apêndice E – Códigos do Simulador.

4.2.6.5 Macros

As macros são blocos de código que podem ser acessados em qualquer campo de lógica do modelo. É útil para escrever funções ou códigos que serão utilizados diversas vezes durante a simulação. Basta criar um nome e chamá-las no local em que se deseja processá-las. A Tabela 4.8 apresenta as principais macros do modelo.

Tabela 4.8 – Macros

Macro	Descrição
Mac_inicializacao	Macro responsável por inicializar os valores de matrizes e executar algumas gravações iniciais. É chamada na lógica de inicialização.
Mac_calculo_final	Macro que realiza cálculos de resultados finais como nível de serviço, por exemplo. É chamada na lógica de finalização.
Mac_processo_geral_sob_demanda	Macro que descreve as regras operacionais da política de entregas sob demanda. É chamada nos processos vinculados aos locais que representam os <i>clusters</i> .
Mac_processo_geral_entregas_programadas	Macro que descreve as regras operacionais da política de entregas programadas. É chamada nos processos vinculados aos locais que representam os <i>clusters</i> .
Mac_importar_solução	Macro responsável por importar a solução das alocações externas realizadas por programação matemática. Acionada quando utiliza-se regra de alocação externa.

Os códigos das principais macros encontram-se no Apêndice E – Códigos do Simulador.

4.3 Representação dos Elementos do Modelo

A partir dos elementos de modelagem disponíveis no *software*, definiu-se a maneira mais adequada para representar no modelo os componentes que constituem o sistema. Os elementos físicos do modelo compreendem os PSVs, os *clusters*, o navio tanque, o fundeio do navio tanque, a fila do navio tanque, o porto e os tanques que representam graficamente os níveis dos estoques dos PSVs, dos *clusters* e do navio tanque ao longo da simulação. A Tabela 4.9 descreve a forma como foram representados os elementos do sistema.

Tabela 4.9 – Modelagem elementos físicos

Elementos físicos	Representação no <i>software</i>	Significado para programação
PSV oleeiro	Entidade/Recurso/Tanque	Através do elemento “entidade” é possível programar as regras operacionais da perspectiva do PSV. Através do “recurso” é possível coletar resultados de taxa de utilização, ociosidade e inoperâncias. Através do elemento “tanque” é possível medir e controlar o nível de estoque do PSV.
Cluster	Local/Tanque	Através do elemento “local” é possível programar os processos a serem executados pelas entidades que os visitarem. Através do elemento “tanque” é possível medir e controlar o nível de estoque do <i>cluster</i>
Navio Tanque	Local/Tanque	Através do elemento “local” é possível programar as regras operacionais das entidades que visitarem o Navio tanque. O elemento “tanque” permite controlar o nível de estoque do navio tanque.
Fila do navio tanque	Local	Região onde as entidades “ePSV_OD” aguardam liberação de capacidade do local “navio tanque” para abastecerem.
Porto	Local	No local Porto é possível definir o processo das entidades que o visitarem.
Tanques	Local	Os elementos “tanque” são modelados como locais como procedimento automático do <i>software</i> .

De acordo com a representação dos elementos descrita na Tabela 4.9, foram criados 37 locais que representam respectivamente o porto, o navio tanque, o fundeio do navio tanque, a região da fila do navio tanque, os 11 *clusters* e seus respectivos tanques que contabilizam os níveis de estoque de óleo diesel, 1 tanque que representa o nível de estoque do navio tanque e os 10 tanques que representam os níveis de estoque dos PSVs.

4.4 Lógica de Programação

Após criar os locais e os demais elementos do modelo, foi desenvolvido o código que traduz as regras operacionais descritas para cada política simulada. A programação de cada política é realizada através dos blocos de código que são processados pelas entidades “ePSV_OD” nos locais em que transitam e as demais estruturas de programação como sub rotinas e macros. Alguns blocos de código são comuns às duas políticas. Outros são específicos para cada uma delas. A atualização do consumo é realizada através da sub rotina “sub_atualiza_consumo”. Essa lógica é ativada no início da simulação tanto para política de entregas programadas quanto para política de entregas sob demanda e é alimentada pelos dados de entrada referentes à taxa de consumo diário descritos na Seção 3.3.1.1. Dessa forma, o valor sorteado para a taxa diária de consumo é dividido por 24 para se obter a taxa de consumo horária. A Figura 4.4 mostra o fluxograma da sub rotina de atualização da taxa de consumo.

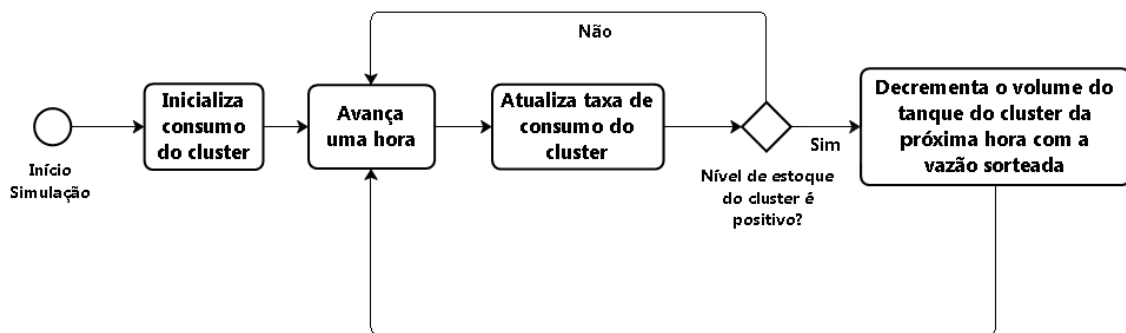


Figura 4.4 – Sub rotina de atualização da taxa de consumo

4.4.1 Política de entregas sob demanda

A lógica de programação da política de entregas sob demanda conta com as sub rotinas “sub_atualização_consumo” e “sub_geração_pedidos” que são responsáveis pela atualização da taxa de consumo dos *clusters* e pela geração de pedidos respectivamente.

A sub rotina “sub_geração_pedidos” inspeciona os níveis de estoque dos *clusters* a cada 12 minutos. O passo do relógio para inspeção do nível de estoque é definido como 12 minutos para acompanhar o passo de discretização do tempo das operações de consumo de fluidos dos elementos “tanque” do simulador. Escolheu-se usar o mesmo

passo de discretização do simulador para garantir uma maior acurácia na medição dos níveis de estoque. Dessa forma, quando os tanques dos *clusters* atingem seus pontos de pedido e o pedido anterior já tiver sido entregue e não houver estoque em trânsito para o *cluster* então uma entidade “ePedido_OD” é criada no local “SetUp”. Nesse local, uma quantidade de pedido é sorteada e gravada na matriz “mPedidos_Aguardando”.

A Figura 4.5 descreve o processo da sub rotina de geração de pedidos.

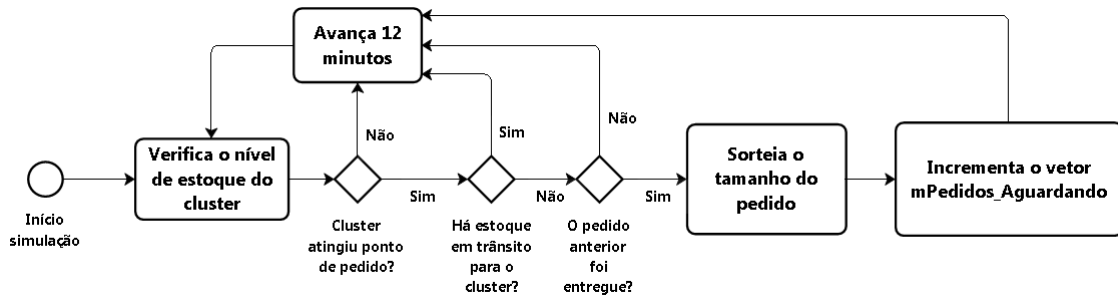


Figura 4.5 – sub rotina de geração de pedidos

A lógica seguida pela entidade “ePSV_OD” na política de entregas sob demanda pode ser descrita em linhas gerais de acordo com a sequência:

1. As entidades “ePSV_OD” chegam ao local “Fundeio_NT” na hora zero da simulação.
2. As entidades “ePSV_OD” aguardam no local em que estão até que ocorra algum pedido, isto é, até que seja gravado um ou mais valores na matriz “mPedidos_Aguardado”. Além disso, verifica-se se já existe algum PSV oleiro navegando para o local do pedido ou operando no local. Se não houver nenhum PSV oleiro navegando ou operando no local ou se houver PSV(s) navegando ou operando no local, mas não irá(ão) encher completamente o tanque do *cluster* então aquele *cluster* é candidato a ser atendido.
3. A partir da lista dos *clusters* candidatos a serem atendidos, a primeira entidade “ePSV_OD” da fila de processos do local seleciona atender o *cluster* que realizou pedido mais cedo.
4. Após selecionar o *cluster* a ser atendido, a programação seguida pela entidade “ePSV_OD” varia de acordo com o local em que a entidade está, havendo dois códigos diferentes: um no processo vinculado ao local “NT_Fundeio” e outro

contido na macro “mac_processo_geral_sob_demanda” que é chamada nos locais que representam os *clusters*.

- 4.1. Caso a entidade “ePSV_OD” se encontre no local “NT_Fundeio”, a mesma aguarda até que haja pelo menos um recurso “rPSV” disponível no local. Assim que um recurso “rPSV” torna-se disponível no local, o mesmo é capturado pela entidade “ePSV_OD” e atualiza-se a posição da matriz “mPedidos_Aguardando” referente ao *cluster* solicitante. Para isso, verifica-se se o estoque do PSV oleiro é suficiente para atender a demanda do *cluster*. No local “NT_Fundeio” todos os recursos “rPSV” encontram-se com tanques cheios, isto é, 2500 m³. Assim, caso o pedido do *cluster* seja inferior a 2500 m³ atualiza-se a posição da matriz “mPedidos_Aguardando” para zero. Caso seja superior então atualiza-se a posição da matriz “mPedidos_Aguardando” com a quantidade remanescente de óleo diesel que não poderá ser entregue.
- 4.2. Caso a entidade “ePSV_OD” se encontre fundeada em algum *cluster*, a mesma verifica quais PSVs oleiros que estão no local possuem em estoque uma quantidade suficiente para suprir a demanda do *cluster*. Se um ou mais PSVs forem capazes de atender a demanda do *cluster*, seleciona-se aquele que tiver o menor volume de óleo diesel em estoque e atualiza-se a posição da matriz “mPedidos_Aguardando” referente ao *cluster* solicitante para zero. Caso nenhum PSV tenha estoque suficiente para atender ao pedido, seleciona-se então o que tiver maior volume em tanque e atualiza-se a posição da matriz “mPedidos_Aguardando” referente ao *cluster* solicitante com a quantidade remanescente de óleo diesel que não pôde ser entregue pelo PSV.
5. A entidade “ePSV_OD”, em posse do recurso “rPSV” navega para o *cluster* solicitante.
6. Ao chegar ao *cluster* a entidade “ePSV_OD” compara o espaço livre do tanque do *cluster* com a quantidade programada a ser entregue e transfere o mínimo entre esses valores.
7. Após a transferência, entidade “ePSV_OD” permanece fundeada no local em que está, em posse do recurso “rPSV” e retorna ao passo (2) até que se esgote o estoque de óleo diesel do tanque do PSV oleiro.
8. Se o estoque estiver esgotado, a entidade “ePSV_OD” navega para o local “NT_fila” onde aguarda para reabastecer.

9. Após reabastecer, a entidade “ePSV_OD” retorna ao local “Fundeio_NT” libera o recurso “rPSV” e volta para o passo (2).

A Figura 4.6 descreve o macro fluxograma do processo da entidade “ePSV_OD” no modelo para política de entregas sob demanda.

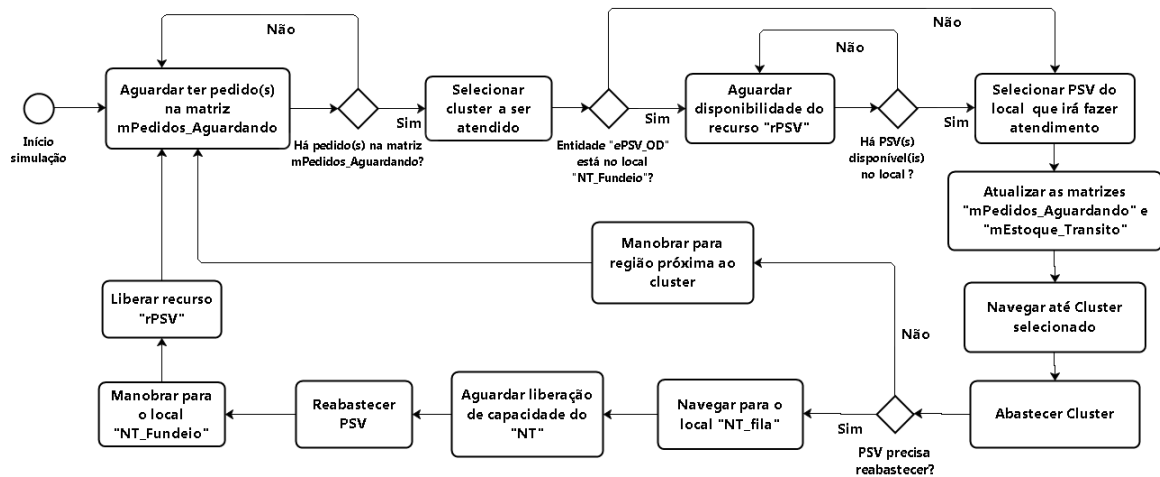


Figura 4.6 – Macro processo da entidade ePSV_OD. Entregas sob demanda

4.4.2 Política de entregas programadas

Nessa política, existem as entidades “eAtendimento” que possui uma chegada programada na frequência de atendimento definida para cada *cluster*. Além disso, também existe a sub rotina “sub_atualização_Consumo”.

A programação dos atendimentos ocorre a partir das chegadas das entidades “eAtendimento” no local “SetUp”. Nesse local as entidades “eAtendimento” recebem um atributo que corresponde ao número do *cluster* e então incrementam em uma unidade a posição da matriz “mAtendimentos_Aguardando” que corresponde ao *cluster* a ser atendido. Essas chegadas se repetem de acordo com o cronograma de viagens estabelecido para os *clusters*. A Figura 4.7 descreve o processo de geração de atendimentos.

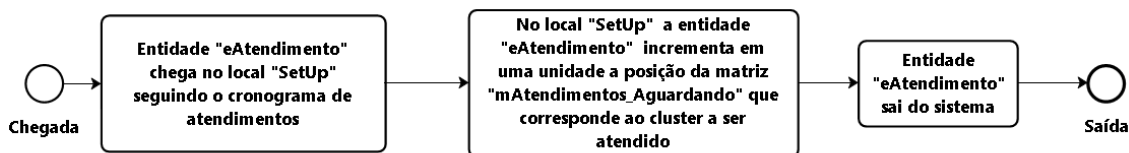


Figura 4.7 – Processo de geração de atendimentos

A lógica seguida pela entidade “ePSV_OD” na política de entregas programadas pode ser descrita em linhas gerais de acordo com a sequência:

1. As entidades “ePSV_OD” chegam no local “Fundeiio_NT”
2. A primeira entidade “ePSV_OD” da fila de processos no local “Fundeiio_NT” aguarda até que um ou mais valores sejam gravados na matriz “mAtendimentos_Aguardando”, indicando uma viagem a ser realizada. Além disso, é verificado se algum PSV oleeiro está navegando para o *cluster* programado ou operando no local. Se não houver nenhum PSV oleeiro navegando ou operando no local ou se houver PSV(s) navegando ou operando no local, mas não irá(ão) encher completamente o tanque do *cluster* então aquele *cluster* é candidato para ser atendido.
3. A partir dos *clusters* candidatos a serem atendidos, a entidade “ePSV_OD” seleciona o *cluster* que possui maior número de atrasos de viagens, isto é, a posição da matriz “mAtendimentos_ aguardando” que possua maior valor. Esta posição corresponde ao número de identificação do *cluster*.
4. A entidade “ePSV_OD” decrementa em uma unidade a posição da matriz “mAtendimentos_Aguardando” correspondente ao *cluster* selecionado para atendimento.
5. Após selecionar o *cluster* a ser atendido, a entidade “ePSV_OD” captura um recurso “rPSV_OD” e navega para o *cluster*.
6. Ao chegar ao local a entidade “ePSV_OD” compara o volume vazio do cluster com o volume em estoque do PSV oleeiro e transfere o mínimo entre esses dois valores.
7. Após realizar a transferência de óleo diesel, a entidade “ePSV_OD” navega para o local “NT_fila” onde aguarda para reabastecer.
8. Após o reabastecimento no local “NT” a entidade “ePSV_OD” retorna para o local “Fundeiio_NT” e libera o recurso “rPSV”.

9. A entidade “ePSV_OD” retorna ao passo (2) onde aguarda a próxima viagem do cronograma.

A Figura 4.8 descreve o macro processo da entidade “ePSV_OD” para política de entregas programadas.

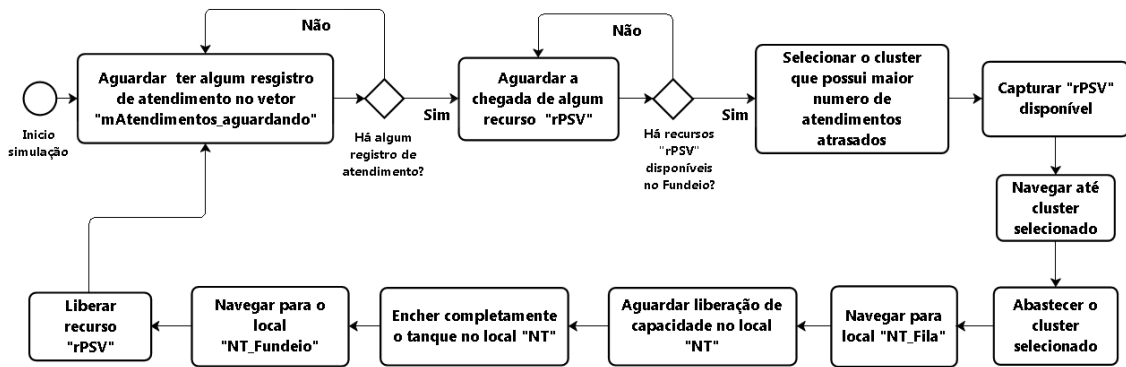


Figura 4.8 – Macro processo da entidade ePSV_OD. Entregas programadas

Além da lógica seguida pela entidade “ePSV_OD”, existe a programação do recurso “rPSV” que é responsável por transportar a entidade pelas rede de caminho construída no modelo. Na Figura 4.9 pode-se verificar a programação do recurso “rPSV”. A programação é exatamente a mesma nas duas políticas e faz parte da modelagem das paradas programada dos recursos, que compreendem as trocas de tripulação e as paradas não programadas que são as falhas ou inoperâncias.

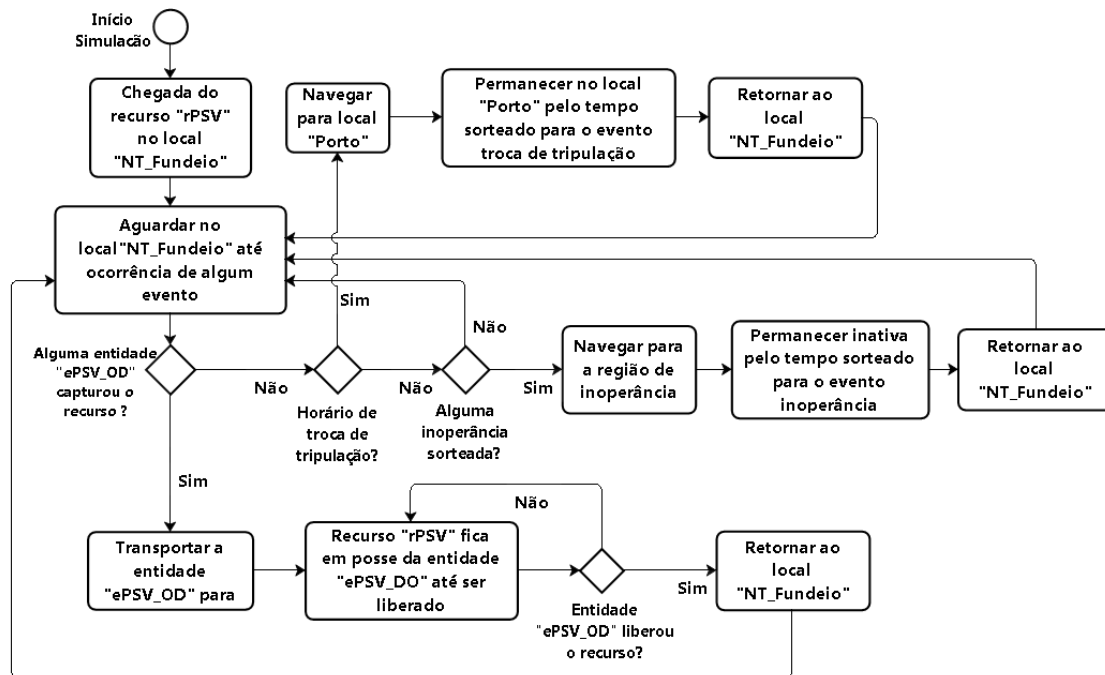


Figura 4.9 – Fluxograma do recurso “rPSV”

4.5 Comunicação entre Simulador e Otimizador

Como descrito na Seção 3.5, pode-se optar por realizar as alocações de PSVs a *clusters* em sub rotinas externas ao simulador, utilizando formulação de programação inteira mista. Nesta seção será descrita a implementação da comunicação entre o simulador e o código que otimiza as alocações diárias dos PSVs.

A execução do código de programação ocorre a cada 24 horas corridas no simulador. Essa rotina de programação é realizada externamente ao simulador, através de uma função do Promodel® de acesso a códigos externos, chamada *XSUB*. Esta função permite que o Promodel acesse outras funções contidas em arquivos do tipo DLL (*Dynamic Link Library*), isto é, uma biblioteca de ligação dinâmica. Assim, através da função *XSUB* são informados os parâmetros necessários para acessar as funções desejadas na DLL mais um parâmetro do tipo *handler* (HWND) da API do sistema Windows® que informa a janela ativa do simulador. O arquivo DLL desenvolvido no estudo foi compilado a partir de um código fonte escrito em C++ através do MS-Visual Studio®. Dessa forma, a função *XSUB* acessa a função *Otimizador* criada dentro da DLL na qual é executado o código de programação inteira

mista desenvolvido na linguagem *Python* e sua biblioteca de programação linear – PuLP. Esta biblioteca utiliza por padrão o solver CBC (COIN-OR *Branch and Cut*), que é um solver *open source* de programação inteira mista desenvolvido na linguagem C++ que utiliza diversos algoritmos como *Branch & Bound*, *Branch and Cut*, Geração de Colunas entre outros, e é utilizado em várias bibliotecas de outros programas. O código fonte da DLL, desenvolvido em C++ assim como o código em *Python* dos dois problemas de programação matemática encontram-se no Apêndice F – Código fonte da DLL e das formulações MIP.

O Promodel® possui funções de escrita e leitura de arquivos tipo “csv”, permitindo exportar e importar dados entre ele e aplicativos externos. Para executar a programação, três arquivos são exportados pelo simulador: o arquivo “data_clusters.csv” informa quais *clusters* realizaram pedidos de óleo diesel e os parâmetros relacionados a eles, o arquivo “data_barcos.csv” informa quais barcos estão disponíveis e os seus parâmetros e o arquivo “data_navegacao.csv” que informa a matriz de tempos de navegação entre os locais do modelo. A partir dos arquivos exportados pelo simulador, o otimizador acessa esses arquivos e então executa a programação. Como resultado, o otimizador devolve para o simulador as alocações dos PSVs aos *clusters* assim como as demandas não atendidas ou demandas restantes de cada *cluster*, que compreendem as variáveis de folga do modelo. Os resultados da otimização são gravados em dois arquivos: o arquivo “Alocacao.csv” e o arquivo “DemandaRestante.csv” que, por sua vez, são acessados pelo simulador.

A Figura 4.10 explica, através de um fluxograma, o processo de comunicação entre o Promodel® e o código externo de programação matemática que realiza as alocações ótimas de PSVs aos *clusters*.

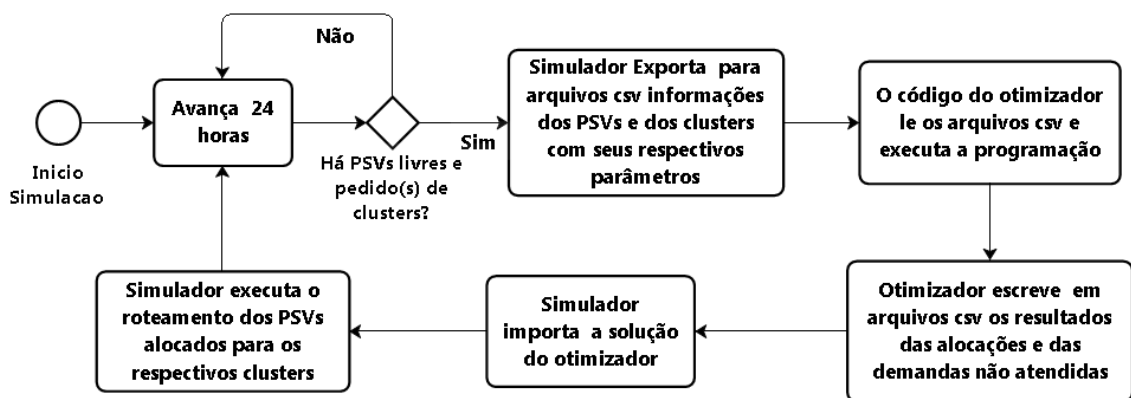


Figura 4.10 – comunicação entre simulador e otimizador

4.6 Resultados de Saída do Modelo

Os resultados de interesse que se deseja medir são relativos aos *clusters* ou aos PSVs. Tais resultados são descritos na Tabela 4.10.

Tabela 4.10 – Resultados do modelo

Variável	Política	Relacionado a	Medida
Volume consumido	Ambas	<i>Cluster</i>	Volume total consumido, em m^3 durante o tempo de simulação.
Volume entregue	Ambas	<i>Cluster</i>	Volume total entregue ao <i>cluster</i> , em m^3 , durante o tempo simulado.
Nível médio de estoque	Ambas	<i>Cluster</i>	Nível médio de estoque contido no tanque do <i>cluster</i> durante o tempo de simulação.
Pedidos realizados	Entregas sob demanda	<i>Cluster</i>	Total de pedidos realizados durante o tempo simulado.
Pedidos entregues	Entregas sob demanda	<i>Cluster</i>	Total de pedidos entregues ao <i>cluster</i> durante o tempo de simulação.
Atendimentos programados	Entregas programadas	<i>Cluster</i>	Total de viagens programadas para o cluster durante o tempo de simulação.
Atendimentos realizados	Entregas programadas	<i>Cluster</i>	Total de atendimentos realizados ao <i>cluster</i> durante o tempo de simulação.
Intervalo médio entre visitas	Ambas	<i>Cluster</i>	Intervalo médio, em horas, entre duas visitas sucessivas ao mesmo <i>cluster</i> .
Nível de serviço operacional	Ambas	<i>Cluster</i>	Porcentagem do tempo em que o estoque do cluster encontra-se positivo.
Nível de serviço de segurança	Ambas	<i>Cluster</i>	Porcentagem do tempo em que o estoque do cluster encontra-se acima do estoque de segurança.
Lead-Time médio	Ambas	<i>Cluster</i>	Tempo médio decorrido entre a realização de um pedido ou horário de uma viagem até a o momento de finalização de atendimento ao <i>cluster</i> .
Custo de estoque	Ambas	<i>Cluster</i>	Custo do capital imobilizado em estoque mantido no <i>cluster</i> durante o tempo de simulação
Custo de falta	Ambas	<i>Cluster</i>	Custo relativo à parada de produção do <i>cluster</i> devido ao desabastecimento.
Custo de exposição ao risco	Ambas	<i>Cluster</i>	Custo atribuído ao tempo que o <i>cluster</i> operou abaixo do estoque de segurança.
Nível médio de estoque no tanque do PSV	Ambas	PSV	Nível médio de estoque contido nos tanques dos PSVs ao longo da simulação.

Taxa de utilização	Ambas	PSV	Relação entre o tempo total de uso do PSV e o tempo total de simulação
Taxa de ociosidade	Ambas	PSV	Relação entre o tempo total em que o PSV encontra-se ocioso e o tempo total de simulação
Taxa de inoperância e troca de turma	Ambas	PSV	Relação entre o tempo total em que o PSV encontra-se inoperante ou trocando tripulação e o tempo total de simulação.
Número de ciclos	Ambas	PSV	Número total de ciclos (tempo decorrido entre uma saída e uma chegada ao fundeio do navio tanque) durante o tempo de simulação
Tempo médio de ciclo	Ambas	PSV	Tempo médio decorrido entre uma saída e um retorno ao fundeio do navio tanque.
Custo operacional	Ambas	PSV	Custo relativo ao tempo em uso do PSV (navegando e operando).
Custo de afretamento	Ambas	PSV	Custo fixo da frota durante o tempo de simulação.

4.7 Verificação e Validação

Antes de experimentar os modelos construídos é preciso verificá-los e validá-los. Para isso, foram realizados testes tanto no modelo simulação quanto no modelo de otimização para certificar que ambos estão programados corretamente e representam a realidade que se deseja analisar. A validação é realizada para o modelo de simulação, onde se compara os resultados das medições geradas pelo simulador e as medições do sistema real, guardando os devidos efeitos das simplificações impostas ao modelo de simulação.

4.7.1 Verificações da Simulação

Após implementar o modelo de simulação, é preciso verificar se o código está funcionando corretamente. Dessa forma, algumas técnicas foram empregadas. Entre elas as verificações das distribuições de consumo e entregas. Foi executada uma simulação com dados determinísticos com as médias das distribuições. Cálculos cruzados foram feitos para verificar os resultados de interesse. Além disso, foi verificado se os dados de saída da simulação condizem com os dados reais, entre esses dados estão os valores de consumo e volume entregue. Para coletar os dados simulados para as duas verificações foi realizada uma replicação piloto de simulação de período de um ano para o cenário de

10 PSVs para a política de entregas sob demanda. A partir do teste piloto, os dados sorteados para os valores de consumo foram guardados em uma matriz e então foram ajustadas distribuições e seus parâmetros usando o mesmo *software* que originou as distribuições dos dados de entrada: o Statfit®

4.7.1.1 Dados de consumo

A verificação dos dados de consumo foi realizada a partir da comparação entre os dados de entrada oriundos da série histórica analisada e os valores gerados pelo simulador. De acordo com a implementação, a taxa de consumo dos *clusters* é atualizada a cada hora. Assim, de hora em hora são sorteados valores de consumo para os 11 *clusters* e estes valores seguem a distribuição de probabilidade de entrada do modelo. Dessa forma, os valores gerados pelo simulador foram coletados e então foi verificado se seguiam as mesmas distribuições e médias dos dados de entrada. A Tabela 4.11 mostra a comparação entre os dados reais (de entrada) e os simulados.

Tabela 4.11 – Verificação dados de consumo

Cluster	Distribuição de origem	Distribuição gerada pelo simulador	Média real	Média simulada
UEP 1	Triangular(172, 281, 172)	Triangular(172, 281, 172)	208.21	208.32
UEP 2	Beta(241, 408, 0.828, 1.21)	Beta(241, 408, 0.847, 1.24)	309.27	308.51
UEP 3	Pearson 6(210, 4.82e+004, 1.43, 1.23e+003)	Pearson 6(210, 1.57e+003, 1.46, 41.8)	259.55	266.22
UEP 4	Weibull(220, 1.53, 49.5)	Weibull(220, 1.55, 49.5)	263.39	264.52
UEP 5	Beta(123, 180, 0.8, 1.18)	Beta(123, 180, 0.774, 1.17)	145.41	145.76
UEP 6	LogLogistic(19., 3.77, 3.07)	LogLogistic(19., 3.87, 3.07)	22.24	22.4
UEP 7	Triangular(33., 64.5, 33.)	Triangular(33., 64.5, 33.)	43.63	43.55
UEP 8	Triangular(54., 70.6, 55.2)	Triangular(54., 70.7, 55.3)	60.33	60.1
Sonda 1	Lognormal(148, 2.4, 0.968)	Lognormal(148, 2.43, 0.964)	164.29	165.8
Sonda 2	Triangular(71., 101, 71.7)	Triangular(71., 101, 71.7)	81.51	81.22
Sonda 3	Uniform(56., 85.)	Uniform(56., 85.)	70.01	70.54

De acordo com a verificação dos dados de volume consumido, pode-se inferir que, de fato, os dados gerados pelo simulador estão correspondendo aos dados de entrada. As distribuições ajustadas para os dados simulados foram exatamente as mesmas ajustadas para os dados de entrada e os parâmetros ajustados de cada distribuição gerada assim como as médias dos dados simulados são muito próximos aos dados de entrada.

4.7.1.2 Dados de volume de pedidos

A verificação dos dados de volume de pedido foi realizada comparando as distribuições oriundas da série histórica analisada e dos valores gerados pelo simulador. Assim, pode-se verificar que as funções de distribuição de probabilidade dos volumes

solicitados pelos *clusters* são as mesmas para os dados de entrada e os dados gerados durante a simulação, com alguma diferença dos parâmetros, apenas. Além disso, a média real – oriundas dos dados de entrada – e a simulada – oriunda dos valores gerados no simulador são bem próximas. A Tabela 4.12 mostra a comparação das distribuições de probabilidade e das médias.

Tabela 4.12 – Verificação dados de volume de pedido

Cluster	Distribuição de origem	Distribuição gerada pelo simulador	Média real	Média simulada
UEP1	Beta(67, 1.97e+003, 0.719, 1.4)	Beta(68, 1.88e+003, 0.641, 1.01)	701.9	797.8
UEP2	Pearson 6(16, 2.44e+003, 2.87, 9.45)	Pearson 6(121, 5.79e+003, 1.84, 18.9)	823.7	730.4
UEP3	Inverse Weibull(3., 1.52, 1.92e-003)	Inverse Weibull(-117, 1.74, 1.38e-003)	930.8	1320.6
UEP4	Beta(100, 3.14e+003, 0.998, 2.92)	Beta(111, 3.33e+003, 1.19, 3.9)	854.9	783.8
UEP5	Gamma(20, 1.84, 353)	Gamma(29, 1.51, 429)	668.2	677.6
UEP6	Uniform(10, 115)	Uniform(10, 114)	62.8	61.6
UEP7	Pearson 6(50, 5.49e+004, 2., 194)	Pearson 6(75, 5.08e+005, 1.78, 1.72e+003)	614.5	603.7
UEP8	Beta(42, 1.18e+003, 0.875, 2.49)	Beta(42, 1.03e+003, 0.922, 2.29)	327.8	326.8
Sonda1	Johnson SB(135, 1.15e+004, 3.83, 1.23)	Johnson SB(167, 5.72e+003, 2.67, 1.1)	756.6	723.0
Sonda2	Weibull(7, 1.74, 565)	Weibull(49, 1.53, 510)	498.0	516.0
Sonda3	Inverse Weibull(200, 1.6, 4.59e-003)	Inverse Weibull(202, 1.62, 4.82e-003)	574.1	656.1

Além das verificações dos dados de consumo e entrega, foram feitas algumas conferências cruzadas para algumas variáveis de saída do modelo, isto é, cálculos feitos a partir de parâmetros diferentes que devem resultar em valores iguais ou próximos. Como exemplo, a variável de saída “taxa de utilização média do PSV” pode ser conferida dividindo-se o produto das variáveis de saída “número de ciclos do PSV” e “Tempo médio de ciclo do PSV” pelo total do tempo simulado. Este valor deve coincidir com a taxa de utilização do PSV, pois é a razão entre o tempo de ocupação total do PSV e o tempo total de simulação. Outro cálculo que pode-se fazer é da capacidade do tanque do PSV, a partir das variáveis de saída “número total de entregas”, “volume médio por entrega” e “número de ciclos do PSV”. Dessa forma, o número de entregas médio por viagem de um PSV – que compreende o quociente entre “número total de entregas” e “número de ciclos do PSV” – multiplicado pelo “volume médio por entrega” deve resultar em um valor próximo da capacidade do tanque do PSV. Assim, a partir desses cálculos o modelo foi verificado.

Por fim, foram rodados alguns cenários com valores médios de consumo e entrega e também com dados de entrada iguais para todos os *clusters*. Com isso, observou-se que as variáveis de saída eram iguais para todos os *clusters*, indicando que o modelo estava respondendo adequadamente à programação.

4.7.2 Validação da Simulação

O processo de validação consiste em confrontar dados e medições gerados pelo simulador com os dados reais registrados. Para isso, foram comparados os dados de consumo total e os dados de volume total entregue. Além disso, também foi comparada a medida de *lead-time*. O processo de validação só pode ser aplicado para a política de entregas sob demanda, pois é aquela praticada pela companhia. Ainda assim, muitas medições não são possíveis de comparação devido às simplificações adotadas no modelo. Para confrontar as medições reais com as simuladas, foi analisado o cenário de 5 PSVs, pois acredita-se que é o tamanho de frota proporcional para atender a demanda de 56 UMs do cenário produtivo da companhia estudada. O volume total consumido pelos *clusters* medido referente aos dados reais referentes ao período de observação um ano foi de 462.981 m³. Para o cenário de simulação de 5 PSVs da política de entregas sob demanda, o volume consumido foi de 580.568 m³ em um ano. O volume total entregue aos *clusters* referente aos dados reais foi de 549.792 m³ para o ano analisado. Para o cenário de 5 PSVs da política de entregas sob demanda, o volume total entregue foi de 571.867 m³ em um ano. Além dessas medidas, o *lead-time* médio medido através dos dados registrados na planilha foi de 4 dias. Para o cenário analisado (5 PSVs), o *lead time* médio medido foi de 3,1 dias. Logo, ainda que reservando os efeitos das simplificações, as medições da simulação apontam uma representação fidedigna à realidade.

4.7.3 Verificações da Otimização

O código da otimização das alocações, desenvolvido com a biblioteca PuLp da linguagem *Python* também foi verificado para certificar que está retornando soluções ótimas de acordo com as formulações testadas. Para realizar as verificações, um exemplo de programação para cada formulação (formulação 1 ou minimização do risco de consumo do estoque se segurança e formulação 2 ou maximização da autonomia) foi realizado para analisar os resultados. Para ambas as formulações foi verificada uma execução de programação para o cenário de 5 PSVs.

4.7.3.1 Formulação de minimização do risco do consumo do estoque de segurança

O exemplo de alocação realizada em um cenário de 5 PSVs para a formulação de minimização do risco do consumo do estoque de segurança foi executada e os arquivos de exportação e resultados foram verificados. Os dados gravados no arquivo

“data_barcos.csv” são apresentados Tabela 4.13 e os dados gravados no arquivo “data_clusters.csv” são apresentados na Tabela 4.14. As tabelas apresentam os dados referentes aos PSVs e aos *clusters* exportados do simulador no momento de uma programação.

Tabela 4.13 – Dados dos PSVs: verificação da formulação 1

PSV	Estoque_Atual (m ³)	Local
2	2500	NT
3	2236,10	UEP2
4	2500	NT

Tabela 4.14 – Dados dos clusters: verificação da formulação 1

Cluster	Estoque_Atual (m ³)	Consumo (m ³ /h)	Qtde_pedida(m ³)	Est_Segurança(m ³)
UEP1	5,45	8,68	154,89	1861
Sonda1	228,69	6,85	385,3	3400
UEP2	2,89	12,89	896,2	3815
UEP4	1422,37	10,97	488,01	4420
UEP5	1281,15	6,06	935,91	3664,23
UEP7	2105,23	1,82	952,47	3555,09
UEP8	0,17	2,51	112,23	1025
Sonda2	321,11	3,4	66,79	1510
Sonda3	1,15	2,92	303,73	1329

De acordo com a Tabela 4.13 e a Tabela 4.14, no momento da programação havia três PSVs disponíveis e nove *clusters* solicitantes no sistema. Portanto, há no modelo de programação 27 variáveis de decisão binárias relativas às alocações e nove variáveis de decisão contínuas relativas às demandas não atendidas. Dois PSVs encontram-se no local “NT_Fundeio” e um PSV encontra-se fundeado no local “UEP2”. Dessa forma, a Tabela 4.15 dispõe a matriz de tempos de navegação em horas relevante para a programação.

Tabela 4.15 – Tempos de navegação em horas: verificação da formulação 1

	UEP1	Sonda1	UEP2	UEP4	UEP5	UEP7	UEP8	Sonda2	Sonda3
UEP2	3,91	2,03	0,5	2,5	1,69	0,98	3,27	3,54	5,11
NT	4,46	1,5	2,2	1,25	1,35	1,95	3,92	1,69	5,14

A partir dos dados apresentados, deseja-se verificar se o resultado da programação está atendendo à formulação 1. Para que a formulação esteja correta, deve-se priorizar atender os clusters que estejam com estoque menos positivos em relação ao estoque de segurança ou aqueles que estejam mais negativos em relação ao estoque de segurança.

De acordo com a Tabela 4.14, todos os *clusters* encontram-se com seus estoques abaixo do nível de segurança no momento da programação. Os três estoques mais negativos, na ordem, são: UEP2, Sonda1 e UEP4.

Após executar programação inteira mista, a solução da alocação é gravada no arquivo “Alocacao.csv” e a demanda não atendida de cada pedido é gravada no arquivo “DemandaRestante.csv”. A solução para o exemplo aloca o PSV2 ao *cluster* Sonda1, o PSV3 ao *cluster* UEP2 e o PSV4 ao *cluster* UEP4 como mostrado na Tabela 4.16. Os PSVs alocados conseguem atender toda a demanda solicitada pelos *clusters*, de forma que não há demanda restante para os *clusters* atendidos. Somente há demanda restante para os *clusters* não atendidos, conforme a Tabela 4.17, e estes *clusters* continuarão solicitando atendimento nas próximas programações.

Tabela 4.16 – Alocações: verificação da formulação 1

PSV	Cluster
2	Sonda1
3	UEP2
4	UEP4

Tabela 4.17 – Demanda não atendida: verificação da formulação 1

Cluster	Demanda_Restante (m ³)
UEP1	154,88
Sonda2	66,79
Sonda3	303,73
Sonda1	0
UEP2	0
UEP4	0
UEP5	935,91
UEP7	952,47
UEP8	112,23

Os dados gravados nos arquivos “Alocacao.csv” e “DemandaRestante.csv” são então acessados pelo simulador que realiza o direcionamento dos PSVs de acordo com a solução ótima encontrada. No exemplo, três dos 9 *clusters* solicitantes serão atendidos. Os resultados obtidos no exemplo da programação permitem concluir que o modelo de otimização está sendo resolvido de forma ótima de acordo com a formulação testada.

4.7.3.2 Formulação de maximização da autonomia

O exemplo de programação para a formulação de maximização da autonomia foi executado para o cenário de 5 PSVs. Da mesma forma que na verificação da formulação 1, foram inspecionados os dados escritos no arquivo “data_clusters.csv” e “data_barcos.csv”. Os dados exportados nesses arquivos são apresentados na Tabela 4.18 e na Tabela 4.19.

Tabela 4.18 – Dados dos PSVs: verificação formulação 2

PSV	Estoque_Atual (m ³)	Local
2	449,61	UEP2
4	750,71	UEP5
5	869,15	UEP3

Tabela 4.19 – Dados dos clusters: verificação formulação 2

Cluster	Estoque_Atual (m ³)	Consumo (m ³ /h)	Qtde_pedida (m ³)	Est_Seguranca (m ³)
UEP1	823,21	8,68	11,31	1861
Sonda1	2574,76	6,85	501,07	3400
UEP2	1245,33	12,89	142,88	3815
UEP3	1599,66	10,81	466,84	6223,40
UEP8	589,42	2,51	438,23	1025
Sonda2	1603,29	3,40	505,42	1510

De acordo com a Tabela 4.18 e a Tabela 4.19, no momento da programação havia três PSVs ociosos e seis *clusters* solicitantes. Dessa forma, há 18 variáveis de decisão binárias relativas às alocações e seis variáveis de decisão contínuas relativas às demandas não atendidas. A partir da localização dos PSVs e dos *clusters* solicitantes, a Tabela 4.20 apresenta os tempos de navegação em horas relevantes para a programação.

Tabela 4.20 – Tempos de navegação: verificação da formulação 2

	UEP1	Sonda1	UEP2	UEP3	UEP8	Sonda2
UEP2	3,91	2,03	0,5	5,55	3,27	3,54
UEP3	8,8	3,6	5,55	0,5	8,15	5,1
UEP5	5,36	0,39	1,69	3,87	4,7	3,38

A partir dos dados apresentados, deseja-se verificar se o resultado da programação está atendendo à formulação 2. Para que a formulação esteja correta, deve-se priorizar atender os *clusters* com menor autonomia. Os três *clusters* com menor autonomia são,

na ordem: UEP1, UEP2 e UEP3. Porém os três *clusters* possuem estoque positivo e, dessa forma, o não atendimento pode gerar valores menores para a função objetivo. De acordo com a Função Objetivo (3.17), quando as duas parcelas são positivas, o atendimento pode gerar valores maiores do que o não atendimento. De acordo com exemplo de programação, foram calculadas autonomias individuais de cada *cluster* e as respectivas parcelas ($DR/Consumo_Médico$) relacionadas ao não atendimento da demanda. A Tabela 4.21 apresenta os *clusters* e suas respectivas parcelas.

Tabela 4.21 – Autonomia e demanda não atendida: verificação formulação 2

Cluster	Estoque_Atual (m ³)	Consumo (m ³ /h)	Qtde_pedida (m ³)	Autonomia (horas)	DR/Consumo (horas)
UEP1	823,21	8,68	11,31	94,89	1,30
Sonda1	2574,76	6,85	501,07	376,13	73,20
UEP2	1245,33	12,89	142,88	96,64	11,09
UEP3	1599,66	10,81	466,84	147,92	43,17
UEP8	589,42	2,51	438,23	234,49	174,34
Sonda2	1603,29	3,40	505,42	472,09	148,82

De acordo com a Tabela 4.21, o atendimento aos três *clusters* com menor autonomia (UEP1, UEP2 e UEP3) gera um custo de $94,89 + 96,64 + 147,92 + 73,20 + 174,34 + 148,82 = 735,81$ para a função objetivo. Por outro lado, o não atendimento de nenhum *cluster* gera um custo de $1,30 + 73,20 + 11,09 + 43,17 + 174,34 + 148,82 = 451,92$ para a função objetivo. Porém, existe a restrição de realizar ao menos um atendimento. Dessa forma, o atendimento que gera menor custo para a função objetivo é o atendimento ao *cluster* UEP8, apresentando um custo para a função objetivo de $234,49 + 1,30 + 73,20 + 11,09 + 43,17 + 148,82 = 512,07$. Logo, a solução para o problema é a solução trivial de atender apenas um *cluster*: o *cluster* UEP8. Todos os PSVs disponíveis no sistema possuem estoques suficientes para atender a demanda do *cluster*, porém, o mais próximo do *cluster* UEP8 é o PSV 2, que encontra-se no *cluster* UEP2. A Tabela 4.22 apresenta o resultado da alocação gravado no arquivo “Alocacao.csv” e a Tabela 4.23 apresenta os resultados das demandas não atendidas gravados no arquivo “DemandaRestante.csv”.

Tabela 4.22 – Alocação: verificação da formulação 2

PSV	Cluster
2	UEP8

Tabela 4.23 – Demandas não atendidas: verificação da formulação 2

Cluster	Demanda_Restante (m ³)
UEP1	11,31
Sonda2	505,42
Sonda1	501,07
UEP2	142,88
UEP3	466,84
UEP8	0,00

Os dados gravados nos arquivos de solução são então acessados pelo simulador que realiza o direcionamento dos PSVs de acordo com a solução ótima encontrada. Os resultados obtidos no exemplo de programação permitem concluir que o modelo de otimização está sendo resolvido de forma ótima, de acordo com a formulação testada.

Além disso, pode-se afirmar que todas as programações são viáveis para as duas formulações apresentadas, pois o problema conta com variáveis de folga e sempre retorna ao menos uma alocação. Além disso, a maior instância do problema ocorre no cenário com 10 PSVs, podendo assumir nesse cenário o número máximo de 110 variáveis de decisão binárias e 11 variáveis de decisão contínuas.

5 Experimentações e Resultados

O estudo foi conduzido através de três experimentos: o Experimento I consiste em aplicar o simulador nas duas políticas, sem calcular o custo de exposição ao risco. O Experimento II também é executado para as duas políticas e inclui nos resultados o custo de exposição ao risco, cujo cálculo provém dos resultados do primeiro experimento. Para cada experimento são analisados 10 cenários, onde varia-se o número de PSVs oleiros de 1 até 10. Ao final de cada experimento, foram coletados os resultados e realizadas as análises comparativas entre as duas políticas. O Experimento III consiste em rodar a política de entregas sob demanda, executando a regra de alocação de PSVs a *clusters* externamente ao simulador, através de duas formulações de programação inteira mista e comparar com os resultados obtidos para a política de entregas sob demanda do Experimento I, ou seja, aquele que não utiliza otimização para o processo de alocação dos PSVs. Para cada replicação de cada experimento, fixou-se o tempo de aquecimento em seis meses e o tempo de simulação em um ano, conforme demonstrado na Seção 5.1. A Tabela 5.1 resume os experimentos realizados e suas principais configurações de entrada.

Tabela 5.1 – Experimentos

Experimento	Política(s) simulada(s)	Custo de Expos. ao risco	Uso de otimizador	Cenários
Experimento I	Ambas	Não	Não	10
Experimento II	Ambas	Sim	Não	10
Experimento III	Entregas sob demanda	Não	Sim	10

Para realizar os experimentos foi utilizado uma máquina equipada com um processador CPU Intel® Core i5 2.27 GHz e 4 GB de memória RAM. O tempo computacional de uma simulação foi de 28 segundos para cada replicação sem uso da regra de alocação externa e de 62 segundos com o uso da regra de alocação externa. Dessa forma o tempo computacional total para executar os Experimentos I e II foi de 5600 segundos cada um, o equivalente a 93 minutos e o tempo computacional para executar o Experimento III foi de 12400 segundos, o equivalente a 3,4 horas.

5.1 Determinação do Tempo de Aquecimento

O tempo de aquecimento foi determinado ao executar o cenário de 10 PSVs do Experimento I para a política de entregas sob demanda por dois anos corridos e observar a variável “nível de estoque do *cluster* UEP7”. Foi escolhido monitorar essa variação porque seu comportamento reflete a aleatoriedade tanto da taxa de consumo quanto do volume do pedido. Além disso, entre os dados de entrada de consumo, o *cluster* que apresenta maior coeficiente de variação é a UEP7. De acordo com a Tabela 3.3, é possível verificar qual *cluster* possui maior coeficiente de variação, isto é, maior razão entre o desvio padrão e a média. Esta informação é traduzida na Tabela 5.2.

Tabela 5.2 – Coeficiente de variação para os dados de consumo

Cluster	Média (m^3)	Desvio-padrão	Coeficiente de variação
UEP1	208,21	27,42	0,1317
UEP2	309,27	51,05	0,1651
UEP3	259,55	37,95	0,1462
UEP4	263,39	29,09	0,1104
UEP5	145,41	18,60	0,1279
UEP6	22,24	2,36	0,1061
UEP7	43,63	8,67	0,1987
UEP8	60,33	3,94	0,0654
Sonda 1	164,29	14,10	0,0858
Sonda 2	81,52	7,73	0,0949
Sonda 3	70,01	9,15	0,1306

Assim, na Figura 5.1, é mostrada a variável “nível de estoque do *cluster* UEP7” para três replicações e sua média para as primeiras 10.000 horas de simulação.

A partir da média do nível de estoque obtida através da simulação das 3 replicações, foram calculadas as médias móveis de tamanhos 5 e 10. As médias móveis suavizam a curva da média dos níveis de estoque, pois descarta um maior número de observações do regime transitório e reduz a influência de picos e vales isolados da curva média. Assim, fica mais fácil a identificação do tempo de aquecimento, isto é, o instante no qual o sistema entra em regime permanente. A Figura 5.2 mostra as curvas da média do nível de estoque do *cluster* UEP7 e das médias móveis para $m = 5$ e $m = 10$.

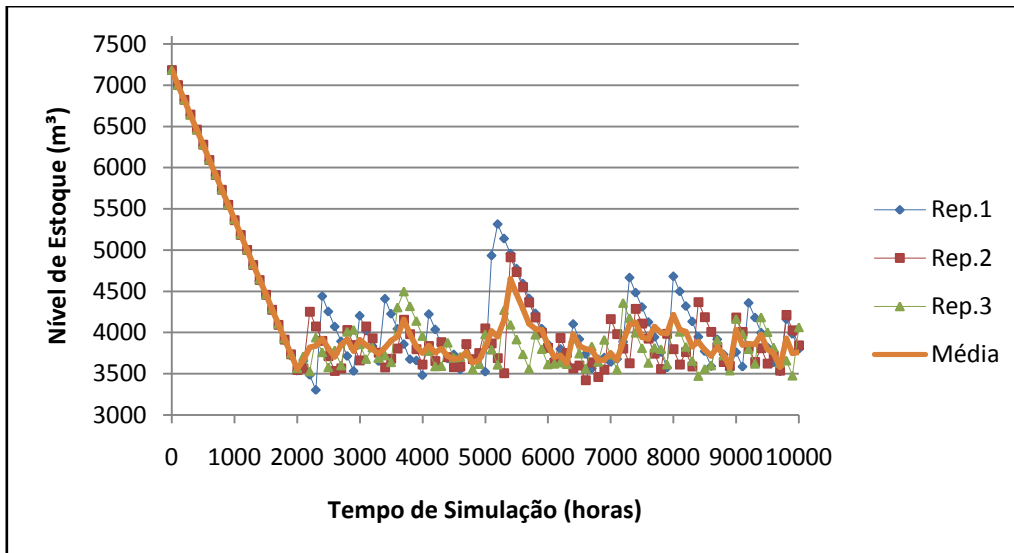


Figura 5.1 – Variação do nível de estoque da UEP7

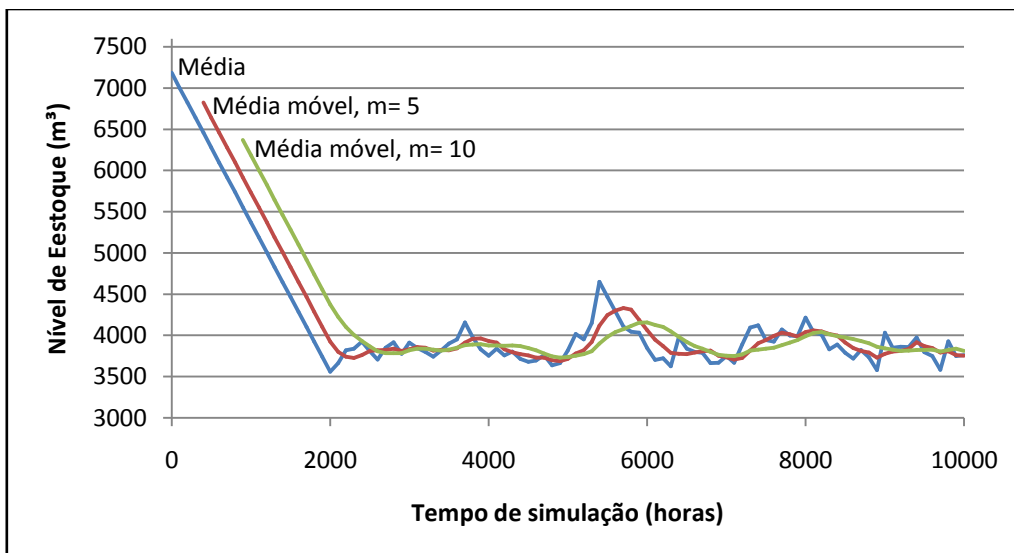


Figura 5.2 – Variação do nível de estoque e médias móveis da UEP7

Na Figura 5.1, pode-se observar que na hora zero da simulação o estoque do *cluster* é igual sua capacidade máxima e a partir da hora 2000, aproximadamente, o comportamento do nível de estoque torna-se estável.

Observando-se as curvas das médias móveis do nível de estoque, através do avanço do tempo de simulação, de acordo com a Figura 5.2, pode-se dizer que o período de aquecimento dura as primeiras 2000 horas, pois a partir deste ponto a curva se estabiliza, não apresentando mais tendências de crescimento ou diminuição. Adotando uma postura conservadora decidiu-se fixar o tempo de aquecimento em 4380 horas, o equivalente a 6 meses. O tempo de simulação foi definido em 8760 horas, o equivalente a um ano.

5.2 Determinação do Número de Replicações

Para determinar o número ideal de replicações a variável escolhida para ser monitorada foi a variável “nível médio de estoque da UEP7”, visto que esta é a que possui maior coeficiente de variação para os dados de consumo. Foi construída uma amostra com 20 replicações para o cenário de 10 PSVs da política de entregas sob demanda. Foi escolhido verificar o resultado “nível de estoque”, pois ele é o que apresenta resultados mais variados, visto que depende diretamente da variável aleatória de entrada taxa de consumo e volume entregue. Além disso, entre os valores de entrada de consumo, o *cluster* que apresenta maior coeficiente de variação é a UEP7, como visto no cálculo do tempo de aquecimento.

Na Tabela 5.3 são mostrados os resultados de uma simulação piloto de 20 replicações para o nível de estoque da UEP7.

Tabela 5.3 – Nível médio de estoque UEP7

Replicação	Nível Médio de estoque
1	0,5476
2	0,5492
3	0,5421
4	0,5409
5	0,5351
6	0,5372
7	0,5263
8	0,5416
9	0,5340
10	0,5433
11	0,5402
12	0,5476
13	0,5441
14	0,5441
15	0,5273
16	0,5271
17	0,5326
18	0,5369
19	0,5443
20	0,5468
Média	0,5394
Desvio-padrão	0,0069

De acordo com Chwif e Medina (2013), pode-se definir o número ótimo de replicações através da construção de um intervalo de confiança para o resultado de interesse que se está medindo. Definindo-se uma precisão do intervalo e uma confiança almejada, determina-se o número ótimo de replicações. A Equação (5.1) define o intervalo de confiança de uma amostra de tamanho n e confiança estatística de $100(1 - \alpha)\%$.

$$h = t_{n-1, \alpha/2} * \frac{s}{\sqrt{n}} \quad (5.1)$$

Onde:

h : Metade do intervalo de confiança;

$t_{n-1, \alpha/2}$: Percentil da distribuição t de Student com $n - 1$ graus de liberdade;

s : Desvio-padrão da amostra

n : Número de dados da amostra.

A partir de h , constrói-se o intervalo de confiança como $[\bar{x} - h, \bar{x} + h]$, onde \bar{x} representa a média amostral.

Dessa forma, para construir um intervalo de confiança para o resultado “nível de estoque da UEP7”, com uma confiança estatística de 95% isto é, $\alpha = 0,05$, substitui-se os valores encontrados para a amostra piloto na Equação (5.1).

Na tabela da distribuição t de Student, $t_{19, 0,025} = 2,093$. Logo, o valor de h é dado por:

$$h = 2,093 * \frac{0,0069}{\sqrt{20}} = 0,003293 \quad (5.2)$$

Portanto, sendo a média da amostra de 20 replicações $\bar{x} = 0,5394$, o intervalo de confiança é $[0,5362; 0,5427]$.

Ainda segundo Chwif e Medina (2013), o número ótimo de replicações n^* desejando uma precisão não menor que h^* , é dada por:

$$n^* = \left\lceil n \left(\frac{h}{h^*} \right)^2 \right\rceil \quad (5.3)$$

Na Equação (5.3), n é o tamanho da amostra, h é a metade do intervalo de confiança obtido para a mostra e h^* é a precisão do intervalo de confiança que se deseja alcançar.

Logo, a partir da amostra piloto de 20 replicações e desejando uma precisão do intervalo de confiança não menor que 0,005, determina-se o número ótimo de replicações, conforme a Expressão (5.4).

$$n^* = \left\lceil 20 * \left(\frac{0,003239}{0,005} \right)^2 \right\rceil = \lceil 8,39 \rceil = 9 \quad (5.4)$$

Assim, o número de replicações necessárias para se obter um intervalo de confiança de 95% e precisão de 0,005 é 9. Dessa forma, para garantir maior confiabilidade nos resultados, optou-se por realizar 10 replicações.

Para ilustrar o comportamento de convergência à média foi realizada uma simulação piloto com 100 replicações, com período de aquecimento de 6 meses e um ano de simulação para política de entregas sob demanda, pois esta política apresenta dados estocásticos tanto de consumo como de entrega. A variável escolhida para coletar as estatísticas foi o nível de estoque da UEP7, pois é a que mais apresentou variação durante a simulação. Para calcular a convergência da série de 100 replicações foi construído o gráfico das observações de cada replicação e da média acumulada dessas observações. Na Figura 5.3 pode-se observar que a série converge rapidamente para média, e mesmo as primeiras replicações apresentam resultados próximos da média. A média acumulada para centésima observação é 0,5427. Este valor de fato está no limite do intervalo de confiança [0,5362; 0,5427], obtido através do experimento da amostra piloto.

O número de replicações necessárias depende da precisão do intervalo de confiança que se deseja obter. Quanto maior o número de replicações menor o intervalo de confiança que contém a média. Dessa, forma, realizando 30 replicações ou mais é possível garantir com quase 100% de certeza que o valor da média das replicações é de fato igual à média real.

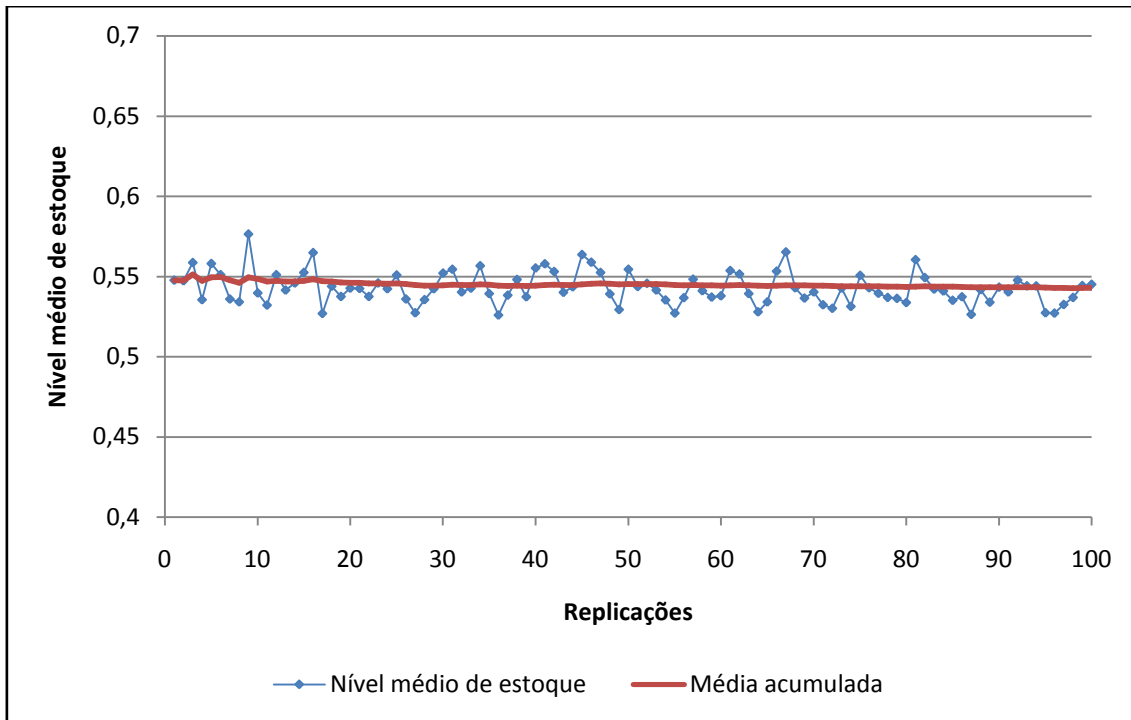


Figura 5.3 – Nível médio de estoque UEP7

Dessa forma, ao se estabelecer um intervalo de confiança para a média das observações, com uma confiança estatística de 95% e uma precisão de 0,5 ponto percentual, isto é, para poder garantir com 95% de certeza que a média real esteja variando em 0,5% para mais ou para menos em relação à média amostral, são necessárias 9 replicações como visto anteriormente. Além disso, pode-se verificar, de acordo com a Tabela 5.3 que a média de 20 replicações foi igual a 0,5394, bem próxima à média acumulada de 100 replicações de 0,5427.

5.3 Resultados de Interesse

Os resultados de interesse dos experimentos compreendem o nível de serviço operacional, o nível de serviço de segurança, o nível médio de estoque nos *clusters*, o *lead-time* médio, a taxa de ocupação e ociosidade dos PSVs e os custos – de falta, de estoque, operacional e de afretamento. Tais resultados são relativos ao sistema durante o tempo total de simulação e, portanto, podem representar as médias das médias, a soma das médias, a soma da soma ou a média das somas. Nessa seção são descritos os cálculos dos resultados de interesse comuns aos Experimentos I, II e III. O custo de

exposição ao risco é calculado apenas no Experimento II, a partir do resultado de custo total do Experimento I e será detalhado no item 5.5.2.

Para facilitar o entendimento, os parâmetros de cálculo, os conjuntos e seus índices serão definidos a seguir:

Conjuntos e índices:

$B, i:$	PSVs
$C, j:$	<i>Clusters</i>
$S, k:$	Ocorrências de <i>stock out</i>

Parâmetros:

$Estoq_Medio_j$	Nível médio de estoque do cluster j durante o tempo de simulação
Cap_j	Capacidade máxima do tanque do cluster j
LTM_j	Lead-time médio do cluster j
$Tacima_Zero_j$	Tempo total em que o estoque do cluster j permanece acima de zero
$Tacima_Seg_j$	Tempo total em que o estoque do cluster j permanece acima do nível de segurança
$Tx_Utiliz_Med_i$	Taxa média de utilização do PSV i
$Tempo_Sim$	Tempo total de simulação
$Custo_Afret_i$	Custo de afretamento do PSV i
$Tempo_Utiliz_i$	Tempo total de utilização do PSV i
$Cust_litro_diesel$	Preço do litro do óleo diesel.
i	Taxa de atratividade do capital
$Tdura_Stkout_{jk}$	Tempo de duração da k -ésima ocorrência de <i>stock out</i> do cluster j
$Prod_Media_j$	Produção média de petróleo <i>brent</i> do cluster j
$Afret_sondas_j$	Custo diário de afretamento das sondas do cluster j
$Total_Clusters$	Número total de <i>clusters</i>
$Total_PSVs$	Número total de PSVs

Nas próximas seções são descritos os cálculos dos resultados de interesse.

5.3.1 Nível médio de estoque

O nível médio de estoque do sistema é definido pela média ponderada dos estoques médios pela capacidade máxima dos tanques dos *clusters* de acordo com a Equação (5.5).

$$\text{Nível médio de estoque} = \frac{\sum_j \text{Estoq_Medio}_j * \text{Cap}_j}{\sum_j \text{Cap}_j} \quad (5.5)$$

5.3.2 Nível de serviço operacional médio

O nível de serviço operacional médio atingido pelo sistema é definido pela média dos níveis de serviço operacional atingidos pelos *clusters*. Esse cálculo pode ser traduzido na Equação (5.6).

$$\text{Nível de serviço operacional médio} = \frac{\sum_j \left(\frac{\text{Tacima_Zero}_j}{\text{Tempo_Sim}} \right)}{\text{Total_Clusters}} \quad (5.6)$$

5.3.3 Nível de serviço de segurança médio

O nível de serviço de segurança médio atingido pelo sistema é definido pela média dos níveis de serviço de segurança atingidos pelos *clusters*. Esse cálculo pode ser traduzido na seguinte Equação:

$$\text{Nível de serviço de segurança médio} = \frac{\sum_j \left(\frac{\text{Tacima_Seg}_j}{\text{Tempo_Sim}} \right)}{\text{Total_Clusters}} \quad (5.7)$$

5.3.4 Taxa média de utilização da frota

A taxa de utilização de um PSV é o cálculo da razão entre o tempo total de uso e o tempo total de simulação, conforme segue:

$$\text{Tx_Utiliz_Med}_i = \frac{\text{Tempo_utiliz}_i}{\text{Tempo_Sim}} \quad (5.8)$$

Logo, a taxa média de utilização da frota total é dada por:

$$\text{Taxa média de utilização da frota} = \frac{\sum_i \text{Tx_Utiliz_Media}_i}{\text{Total_PSVs}} \quad (5.9)$$

5.3.5 Lead-time médio

O *lead-time* médio de entrega é calculado através da média dos *lead-time* médios de atendimento aos *clusters*, conforme definido na Equação (5.10)

$$\text{Lead time médio} = \frac{\sum_j LTM_j}{\text{Total_Clusters}} \quad (5.10)$$

O *lead-time* médio de cada *cluster* j (LTM_j) é calculado na lógica de término da simulação, através dos resultados armazenados na matriz “mLead_Time” para cada entrega registrada. Como um *cluster* representa um conjunto de UMs, o tempo de entrega é calculado após o término da transferência de óleo diesel, o que significa na prática o tempo decorrido até atender a última UM do *cluster*.

5.3.6 Custo de afretamento total

O custo de afretamento de um PSV é definido como 30.000 USD por dia. Portanto, o custo de afretamento total da frota durante o período simulado é dada por:

$$\text{Custo de afretamento total} = \sum_i \text{Custo_Afret}_i \times \text{Tempo_Sim} \quad (5.11)$$

O estudo considera a frota homogênea e os custos de afretamento iguais para todos PSVs.

5.3.7 Custo operacional total da frota

O custo operacional de um PSV é definido como 15% do custo de afretamento. Portanto, calculando o custo de afretamento por hora, obtém-se que o custo operacional por hora de utilização é:

$$\text{Custo operacional total} = \sum_i \text{Tempo_Utiliz}_i * 0,15 * \text{Custo_Afret}_i \quad (5.12)$$

5.3.8 Custo de estoque total

O custo de estoque como visto anteriormente compreende o custo do capital imobilizado a uma taxa de atratividade de 10% ao ano a um regime de juros compostos. Porém, de acordo com o item 3.3.4, optou-se por utilizar o regime de capitalização contínua, realizando-se a conversão da taxa de juro composto para contínuo. Considerando o nível médio de estoque dos *clusters* durante a simulação, em m^3 , i a taxa de atratividade de capital e n o período total de simulação, pode-se calcular o custo de estoque do sistema pela Equação (5.13):

$$\sum_j \text{Estoq_Medio}_j \times 1000 \times \text{Cust_litro_diesel} \times (e^{in} - 1) \quad (5.13)$$

De acordo com a fórmula, o nível médio de estoque em m^3 deve ser multiplicado por 1000 para converter o valor para litros.

5.3.9 Custo de falta total

O custo de falta representa o custo da sonda parada caso a UM seja uma sonda ou o custo de oportunidade da receita não gerada devido à parada de produção, para UMs que são UEPs. Esses custos são associados a cada duração de ocorrência de desabastecimento em cada UM e são somados no final da simulação.

5.3.9.1 Custo de falta para sondas

O custo de falta para *clusters* compostos por sondas compreende o custo da sonda parada, considerando o valor do frete diário de uma sonda como 250.000 USD. Para aumentar a precisão da medida esse custo foi transformado para custo horário e para cada hora parada por falta de óleo diesel, é contabilizado o custo associado a essa parada pelo tempo em que o *cluster* permaneceu desabastecido. Dessa forma, o custo de falta para *clusters* compostos por sondas é calculado através da Expressão (5.14).

$$\sum_j \sum_k Tdura_Stkout_{jk} \times Aftret_sondas_j \quad (5.14)$$

5.3.9.2 Custo de falta para UEPs

O custo de falta para *clusters* compostos por UEPs compreende o custo de oportunidade da receita atrasada devido à parada de produção, considerando uma taxa de atratividade do capital de 10% ao ano, em regime de juros compostos. Como se associa um tempo de duração para cada parada e tal duração pode estar na escala de horas, optou-se por utilizar regime de juros contínuos, conforme apresentado na Seção 3.3.4. Dessa forma, o custo de falta para clusters compostos por UEPs pode ser calculado através da Expressão (5.15).

$$\sum_j \sum_k Tdura_Stkout_{jk} \times Prod_Media_j \times (e^{i \times Tdura_Stkout_{jk}} - 1) \quad (5.15)$$

Foi decidido utilizar taxa de juro contínuo visto que o sistema calcula as durações das paradas em horas. Logo, o custo de oportunidade da receita não gerada é a remuneração deixada de receber decorrente ao atraso da produção, onde a unidade de tempo do relógio da simulação é a hora.

Nas próximas seções são apresentados os experimentos realizados e seus resultados.

5.4 Experimento I

O Experimento I consiste em executar o modelo de simulação pelo período de um ano para as duas políticas variando o cenário de 1 até 10 PSVs. São realizadas 10 replicações para cada cenário para cada política. Portanto, são executadas um total de 200 replicações.

5.4.1 Comparação dos resultados do Experimento I

Após executar o Experimento I, foram analisados os resultados e realizada a análise comparativa entre a política de entregas programadas e a política de entregas sob demanda. Para tanto, foram avaliados os resultados de interesse especificados na Seção 5.3. A partir da coleta dos resultados de interesse foram construídos gráficos que representam a variação dos indicadores de desempenho e os custos de acordo com o cenário simulado para cada política avaliada.

5.4.1.1 Indicadores de desempenho

Os indicadores de desempenho analisados compreendem o nível médio de estoque, a taxa de ocupação média do PSV, o nível de serviço operacional médio e o nível de serviço de segurança médio. O *lead-time* médio também foi medido, porém, para facilitar a leitura, é apresentado em um gráfico à parte. A Figura 5.4 mostra a comparação entre os indicadores de desempenho das duas políticas avaliadas.

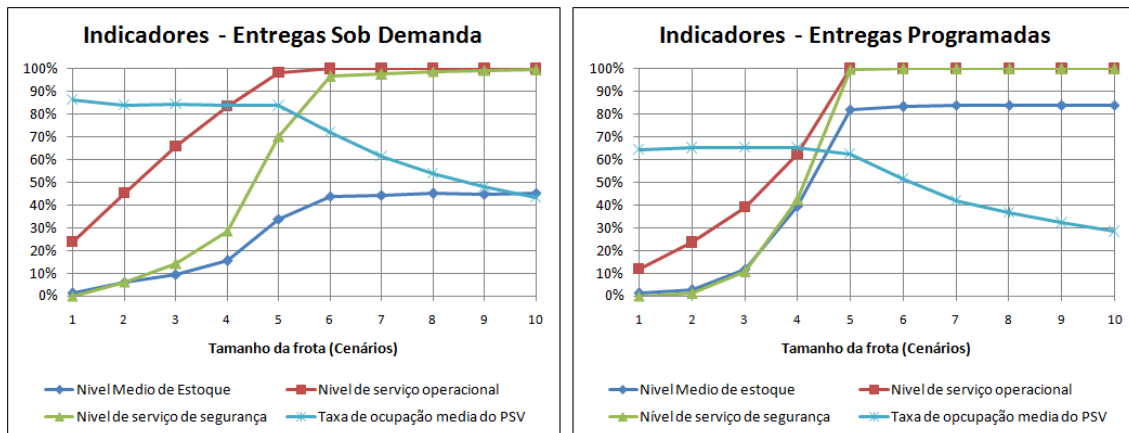


Figura 5.4 – indicadores de desempenho Experimento I

De acordo com a comparação, é possível verificar que para política de entregas sob demanda, o tamanho da frota que garante um nível de serviço operacional próximo a 100% é a partir de 5 PSVs. Porém para atingir um nível de serviço de segurança próximo a 100% são necessários 6 PSVs ou mais. Já para a política de entregas programadas, a partir do cenário com 5 PSVs pode-se garantir tanto um elevado nível de serviço operacional quanto de segurança. Além disso, é possível ver que o nível médio de estoque atinge valores maiores para a política de entregas programadas ultrapassando o nível de 80% a partir do Cenário 5. Por outro lado, na política de entregas sob demanda, o nível de estoque máximo atingido pelo sistema fica em torno de 45%. Por fim, a taxa de ocupação média do PSV é menor para a política de entregas programadas. Essa análise leva a crer que o número ótimo de PSVs está entre 5 e 7 PSVs para ambas as políticas.

Outro indicador que se comparou foi o *lead-time* médio, conforme mencionado anteriormente. A Figura 5.5 apresenta o gráfico comparativo do *lead-time* médio para as duas políticas. Na política de entregas sob demanda, o *lead-time* diminui gradativamente com o aumento do número de PSVs. Isso ocorre porque com mais PSVs

os pedidos são atendidos mais rapidamente. Já para a política de entregas programadas, o *lead-time* médio é baixo para uma frota menor porque muitos *clusters* não chegam a ser atendidos em nenhum momento da simulação, o que não conta como registro. A partir de 5 PSVs o *lead-time* se mantém constante em 3,5 dias. Para os Cenários 5 a 10, o *lead-time* da política de entregas sob demanda é menor, chegando a 1,5 dia para o Cenário 10.

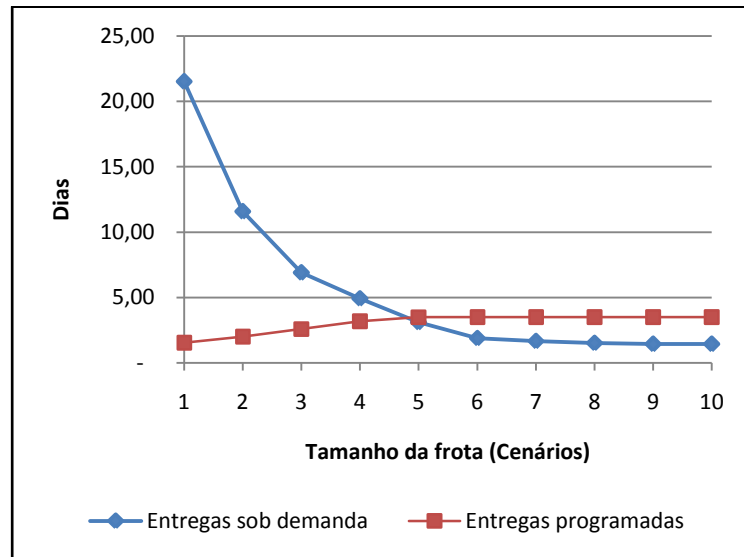


Figura 5.5 – Lead-time médio

5.4.1.2 Custos

Os custos analisados no Experimento I compreendem o custo de afretamento, o custo operacional dos PSVs, o custo de estoque e o custo de falta. Todos os custos foram normalizados na escala de 0 a 1, isto é, dividiram-se os resultados individuais de cada cenário pelo máximo valor observado. Os gráficos de comparação dos custos entre as duas políticas encontram-se na Figura 5.6.

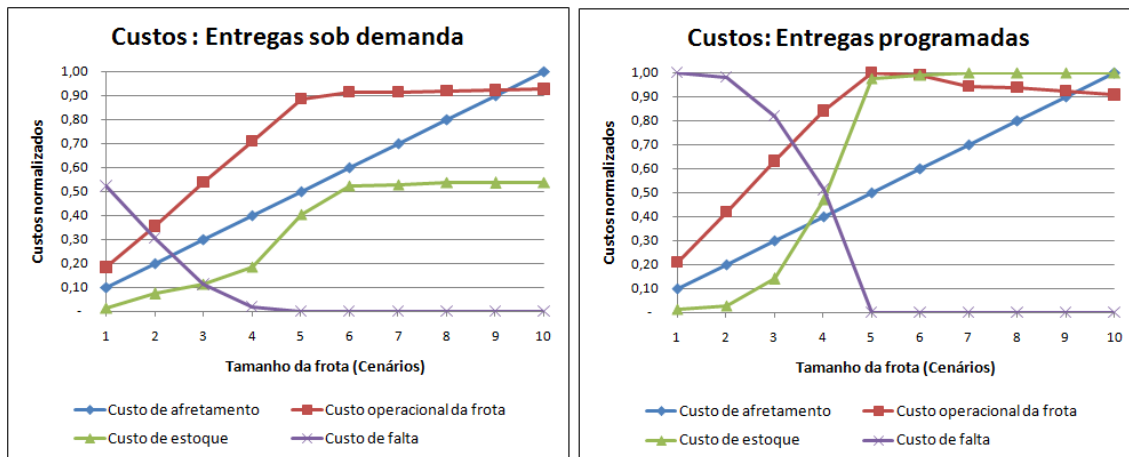


Figura 5.6 – Custos Experimento I

Analisando os gráficos comparativos dos custos, pode-se verificar que o custo de estoque é maior para a política de entregas programadas. Isso é consequência direta dos maiores níveis de estoque mantidos quando se adota essa política. O custo operacional total da frota também é maior para a política de entregas programadas para cenários acima de 5 PSVs. Já o custo de falta é bem maior para a política de entregas programadas para os Cenários de 1 a 4. A partir do Cenário 5, as duas políticas atingem custo de falta próximo a zero. Por fim, o custo de afretamento é exatamente o mesmo para as duas políticas, pois varia linearmente com o tamanho da frota.

A comparação do custo total do sistema para as duas políticas é representado no gráfico da Figura 5.7. Como se pode observar, o custo total da política de entregas programadas é superior ao custo da política de entregas sob demanda. A diferença é muito maior nos Cenários de 1 a 5. Para os Cenários de 5 a 10 PSVs os custos das duas políticas se aproximam, mas a política de entregas programadas ainda é mais cara. A diferença maior para os Cenários de 1 a 5 pode ser explicada devido ao maior custo de falta para política de entregas programadas: em cenários mal dimensionados, a política de entregas programadas prioriza o atendimento dos *clusters* que demandam por uma maior frequência de atendimentos, o que leva alguns *clusters* a nunca serem atendidos. Já a política de entregas sob demanda, em cenários mal dimensionados, consegue atender a todos os *clusters* mesmo ocorrendo faltas. Isso é possível porque o critério de atendimento é FIFO e, dessa forma, mesmo *clusters* que possuem baixa frequência de pedidos são atendidos.

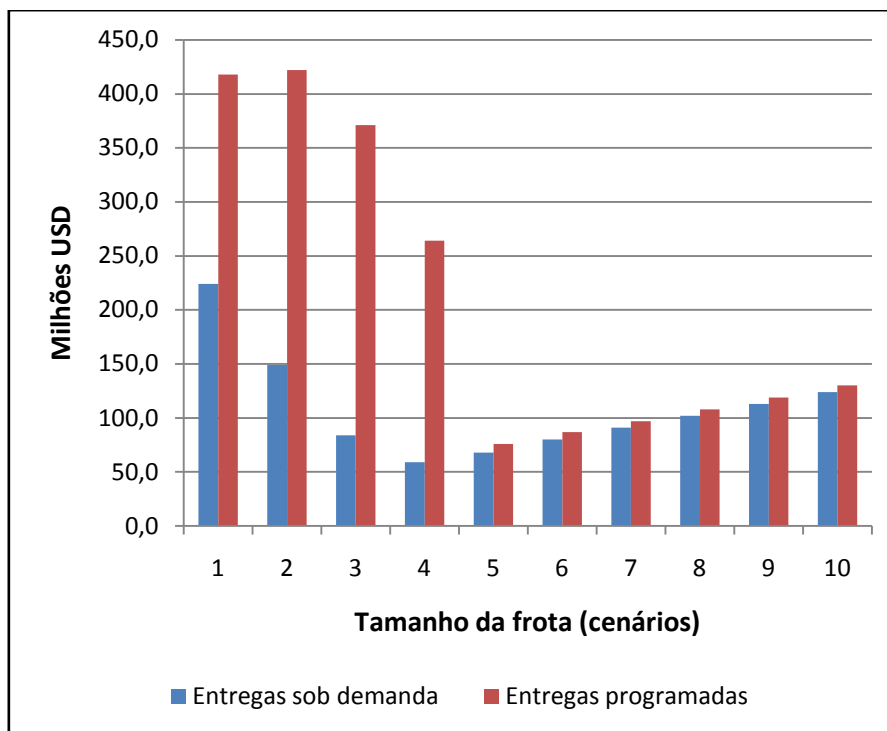


Figura 5.7 – Comparação custo total obtido no Experimento I

A Tabela 5.4 resume os resultados de nível de serviço e custo total para os cenários simulados no Experimento I para a política de entregas sob demanda. De acordo com a tabela é possível observar que para os Cenários de 6 a 10 PSVs o nível de serviço operacional atinge 100%, isto é, não ocorrem desabastecimentos no sistema. Porém o sistema nunca atinge 100% de nível de serviço de segurança, embora chegue perto.

Tabela 5.4 – Resultados política sob demanda: Experimento I

Cenários:	1	2	3	4	5	6	7	8	9	10
Nível de serviço operacional (%)	23,9	45,2	66	83,6	98,3	100	100	100	100	100
Nível serviço de segurança (%)	0	6	14,3	28,5	70	96,7	97,6	98,9	99,2	99,5
Custo Total do Sistema (Milhões USD)	224,0	149,0	84,0	59,0	68,0	80,0	91,0	102,0	113,0	124,0

A Tabela 5.5 resume os resultados de nível de serviço e custo total para os cenários simulados no Experimento I para a política de entregas programadas. De acordo com a tabela pode-se observar que a partir do Cenário 5 não ocorrem desabastecimentos no sistema. Além disso, ao contrário da política de entregas sob demanda, na política de

entregas programadas, os sistema atinge 100% de nível de serviço de segurança a partir do Cenário 7.

Tabela 5.5 – Resultados política de entregas programadas: Experimento I

Cenários:	1	2	3	4	5	6	7	8	9	10
Nível de serviço operacional (%)	11,96	23,56	39,13	62,37	100	100	100	100	100	100
Nível serviço de segurança (%)	0	1,04	10,71	42,38	99,82	99,97	100	100	100	100
Custo Total do Sistema (Milhões USD)	418,0	422,0	371,0	264,0	76,0	87,0	97,0	108,0	119,0	130,0

Comparando as duas tabelas, observa-se que os custos da política de entregas programadas é superior ao custo da política de entregas sob demanda. Porém, na política de entregas programadas é possível atingir 100% de nível de serviço de segurança, evitando operar abaixo do estoque de segurança.

5.5 Experimento II

O Experimento II consiste em rodar 10 replicações para cada cenário para ambas as políticas durante o tempo de simulação de um ano, porém dessa vez considerando o custo de exposição ao risco, isto é, o custo de o sistema operar abaixo do nível de serviço de segurança de 100%. Para mensurar tal custo, foram analisados os resultados do Experimento I para as variáveis de interesse nível de serviço de segurança e custo total. A partir do gráfico de dispersão entre essas duas grandezas foi então ajustada a melhor função que descrevia o custo total adicional do sistema de acordo com a melhoria percentual no nível de serviço de segurança. Com essa função foi calculado o custo do seguro, descrito no próximo tópico.

5.5.1 Custo do seguro

De forma a mensurar o custo de operar abaixo do nível de segurança, foi realizado um estudo sobre o quanto se paga a mais para se obter um sistema com mario nível de serviço de segurança.

A Figura 5.8 mostra os gráficos com as funções ajustadas que relacionam a diferença percentual entre o nível de serviço de segurança atingido com o custo total do

sistema. Para se calcular o custo de seguro, foi feito um gráfico a partir dos resultados do Experimento I.

A função que melhor se ajusta à relação entre o incremento percentual do nível de serviço de segurança e custo total é uma função polinomial de grau um como visto na Figura 5.8.

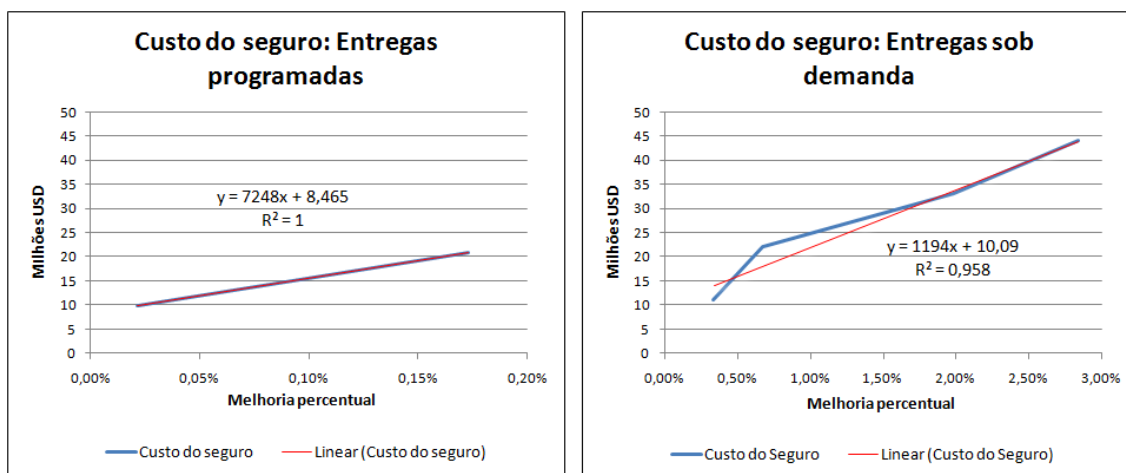


Figura 5.8 – Custo do seguro

De acordo com a Figura 5.8, para a política de entregas programadas, foi ajustada perfeitamente uma reta ($R^2 = 1$) cuja função é dada por $f(x) = 7248x + 8,465$. Para a política de entregas sob demanda, também foi ajustada uma reta ($R^2 = 0,958$) cuja função é dada por $f(x) = 1194x + 10,09$. Nas funções, x é a melhoria percentual no nível de serviço de segurança e $f(x)$ é o custo incremental em milhões USD associado à melhoria.

5.5.2 Custo de exposição ao risco

De forma inversa ao custo de seguro, o custo de exposição ao risco compreende o custo pago pelo sistema ao apresentar um nível de serviço de segurança inferior a 100%. O custo de exposição ao risco é um custo inerente ao sistema e mensura o custo de se operar abaixo de 100% de nível de serviço de segurança. É necessário frisar que somente é calculado o custo de exposição ao risco quando não apresenta nenhuma ocorrência de falta no sistema.

Através das funções ajustadas para o custo do seguro, calcula-se o custo de exposição ao risco na lógica de término da simulação. A Figura 5.9 mostra as curvas ajustadas por funções polinomiais de grau um.

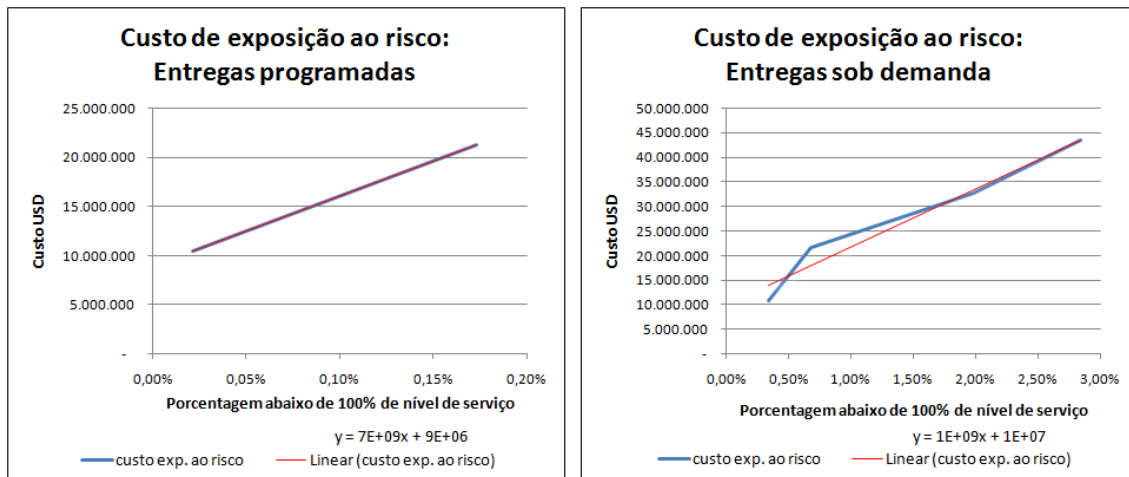


Figura 5.9 – Custo de exposição ao risco

Dessa forma, no final do período de simulação é calculado o custo de exposição ao risco associado a cada política através das Funções (5.16) e (5.17), relacionadas à política de entregas programadas e à política de entregas sob demanda, respectivamente. Nas funções, x é a diferença entre o nível de serviço de segurança 100% e o nível de serviço de segurança atingido na simulação e $f(x)$ é o custo em USD.

$$f(x) = 7 \times 10^9 x + 9 \times 10^6 \quad (5.16)$$

$$f(x) = 10^9 x + 10^7 \quad (5.17)$$

5.5.3 Comparação dos resultados do Experimento II

Após executar os 10 cenários de simulação para cada política, foram analisados e comparados os resultados do Experimento II. A única mudança entre o Experimento I e II é a inserção do custo de exposição ao risco.

5.5.3.1 Indicadores de desempenho

Os indicadores de desempenho avaliados no Experimento II são exatamente os mesmos analisados no Experimento I e são expostos na Figura 5.10. Além disso, os resultados para os indicadores conservam as mesmas características analisadas no Experimento I.

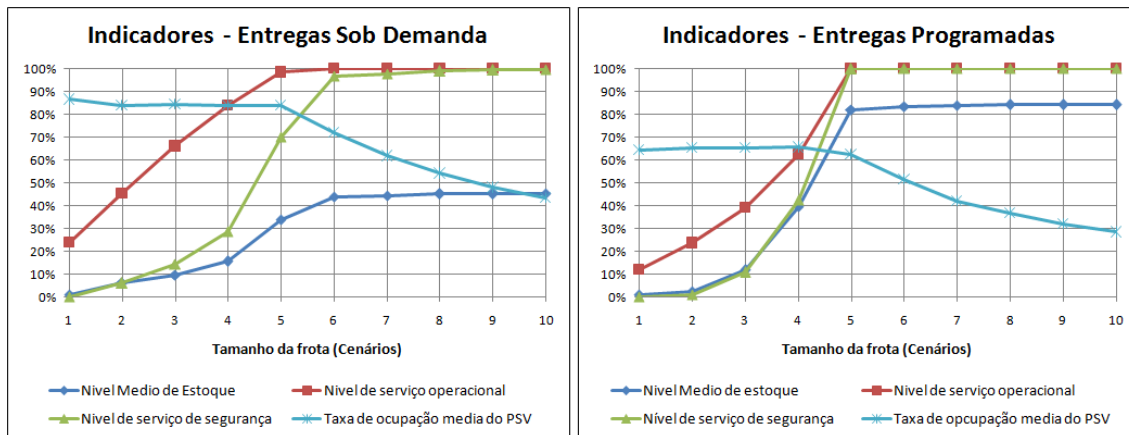


Figura 5.10 – Indicadores de desempenho do Experimento II

Os indicadores do Experimento II são quase idênticos aos do Experimento I. Portanto, podem-se tirar as mesmas conclusões obtidas no Experimento I. Além desses indicadores, o *lead-time* médio também apresenta o mesmo resultado para ambas as políticas. Resultados próximos para os indicadores de desempenho dos Experimentos I e II eram esperados, visto que ambos os experimentos possuem exatamente os mesmos dados de entrada com exceção do custo de exposição ao risco.

5.5.3.2 Custos

Como explicado anteriormente, no Experimento II insere-se o custo de exposição ao risco. O custo de exposição ao risco é um custo inerente ao sistema, ou seja, não é individualizado por *cluster* e sim reflete o quão distante o sistema ficou de 100% do nível de serviço de segurança. Vale ressaltar que o custo de exposição ao risco só faz sentido ser calculado quando não há ocorrências de falta no sistema. Portanto esse custo só será observado em cenários em que o nível de serviço operacional é de 100% mas o nível de serviço de segurança é menor que 100%. Assim, para a política de entregas sob demanda, esse custo é observado nos Cenários 6, 7, 8, 9 e 10. Já para a política de entregas programadas, esse custo é observado nos Cenários 5 e 6 apenas. Os resultados dos custos do Experimento II são mostrados na Figura 5.11.

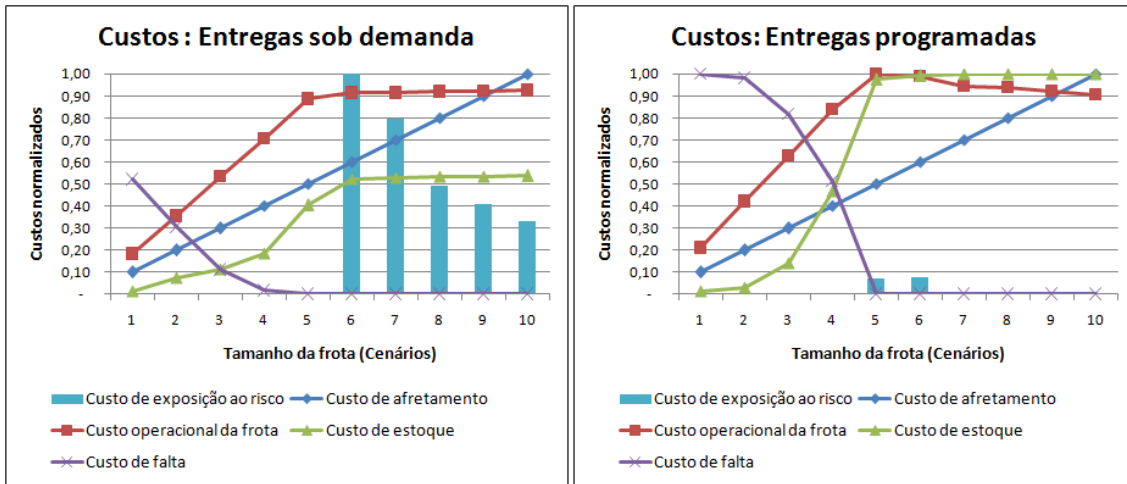


Figura 5.11 – Custos no Experimento II

De acordo com o gráfico, verifica-se que há custo de exposição ao risco para os Cenários de 6 a 10 PSVs na política de entregas sob demanda. Isso ocorre porque, para esta política o sistema atinge 100% de nível de serviço operacional a partir do Cenário 6, porém nunca atinge 100% de nível de serviço de segurança. Já a política de entregas programadas atinge 100% de nível de serviço operacional a partir do Cenário 5. Porém, nessa política só ocorreu consumo do estoque de segurança nos Cenários 5 e 6. Logo, para a política de entregas programadas só existe custo de risco para os Cenários 5 e 6.

A comparação do custo total do sistema entre as duas políticas avaliadas é mostrada no gráfico da Figura 5.12.

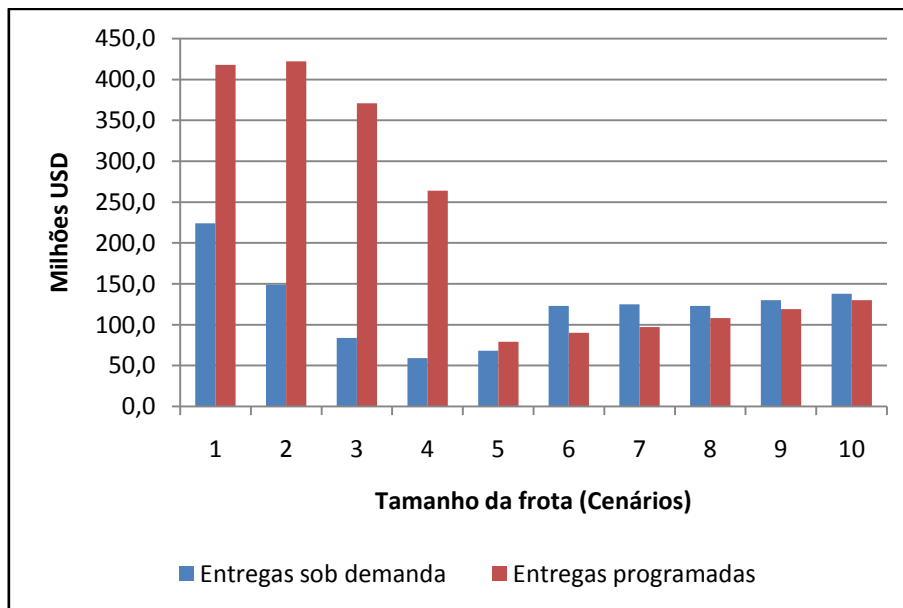


Figura 5.12 – Comparação custo total para o Experimento II

Pode-se observar que o custo total da política de entregas sob demanda torna-se maior do que da política de entregas programadas a partir do Cenário de 6 PSVs. Como visto, nos Cenários de 6 a 10 PSVs existe custo de exposição ao risco para a política de entregas sob demanda. Esse custo é função da diferença entre 100% e o nível de serviço de segurança atingido. Na política de entregas programadas o custo de exposição ao risco só existe nos Cenários 5 e 6 e esse custo é bem baixo, comparado ao custo inerente à política de entregas sob demanda. Isso porque nos Cenários 5 e 6 o sistema chega muito próximo a 100% de nível de serviço de segurança.

A Tabela 5.6 resume os resultados de nível de serviço e custo total para os cenários simulados no Experimento II para política de entregas sob demanda. De acordo com a tabela, é possível observar que o sistema atinge custo mínimo no Cenário de 4 PSVs. Porém, o sistema só atinge 100% de nível de serviço operacional a partir do Cenário 6. Com isso há uma tendência a concluir que o custo de falta compensa o custo adicional de afretamento e o custo operacional maior para um cenário de mais PSVs.

Tabela 5.6 – Resultados política entregas sob demanda: Experimento II

Cenários:	1	2	3	4	5	6	7	8	9	10
Nível de serviço operacional(%)	23,9	45,3	66,0	83,7	98,4	100,0	100,0	100,0	100,0	100,0
Nível serviço de segurança (%)	0,1	6,0	14,4	28,5	70,0	96,8	97,6	98,9	99,3	99,6
Custo total do sistema (Milhões USD)	224,0	150,0	85,0	60,0	68,0	123,0	126,0	123,0	131,0	138,0

A Tabela 5.7 resume os resultados de nível de serviço e custo total para os cenários simulados no Experimento II para a política de entregas programadas. Analisando a tabela, percebe-se que o sistema atinge custo mínimo no Cenário 5, porém o sistema só atinge 100% de nível de segurança a partir do Cenário 7.

Tabela 5.7 – Resultados política entregas programadas: Experimento II

Cenários:	1	2	3	4	5	6	7	8	9	10
Nível de serviço operacional(%)	12,0	23,6	39,1	62,4	100,0	100,0	100,0	100,0	100,0	100,0
Nível serviço de segurança (%)	0,0	1,04	10,71	42,39	99,83	99,98	100,0	100,0	100,0	100,0
Custo total do sistema (Milhões USD)	418,0	423,0	371,0	264,0	80,0	91,0	98,0	109,0	119,0	130,0

Através da comparação entre as duas tabelas de resultados do Experimento II, verifica-se que a política de entregas sob demanda torna-se mais cara do que a política de entregas programadas a partir do Cenário 6, devido à inserção do custo de exposição ao risco. Isso inverte o resultado obtido no Experimento I e permite mensurar o não atendimento de 100% de nível de serviço de segurança na política de entregas sob demanda.

5.6 Experimento III

No Experimento III foram realizadas as simulações da política de entregas sob demanda experimentando-se as duas formulações de programação inteira mista para realizar alocação dos PSVs aos *clusters*. A primeira com a função objetivo de minimizar o risco do consumo do estoque de segurança (ou minimizar a diferença entre o estoque no momento da entrega e o seu estoque de segurança), e a segunda com a função objetivo de maximizar a autonomia do sistema. No Experimento III foram realizadas 10 replicações para os 10 cenários para cada critério de alocação na simulação da política de entregas sob demanda. Nos próximos tópicos são expostas as comparações dos resultados obtidos a partir das simulação com regras de alocação externa de PSVs e a simulação utilizando o critério FIFO de seleção do *cluster* programado internamente na lógica do simulador como no Experimento I. Vale ressaltar que o Experimento III visa fazer comparações somente para a política de entregas sob demanda através da análise das regras de alocação de PSVs aos *clusters*.

5.6.1 Regra FIFO versus minimização do risco de consumo do estoque de segurança

A primeira comparação foi realizada entre a regra de minimização do risco do consumo de segurança (Regra 1) e a regra FIFO interna do simulador. Foram comparados os indicadores obtidos, conforme mostra a Figura 5.13.

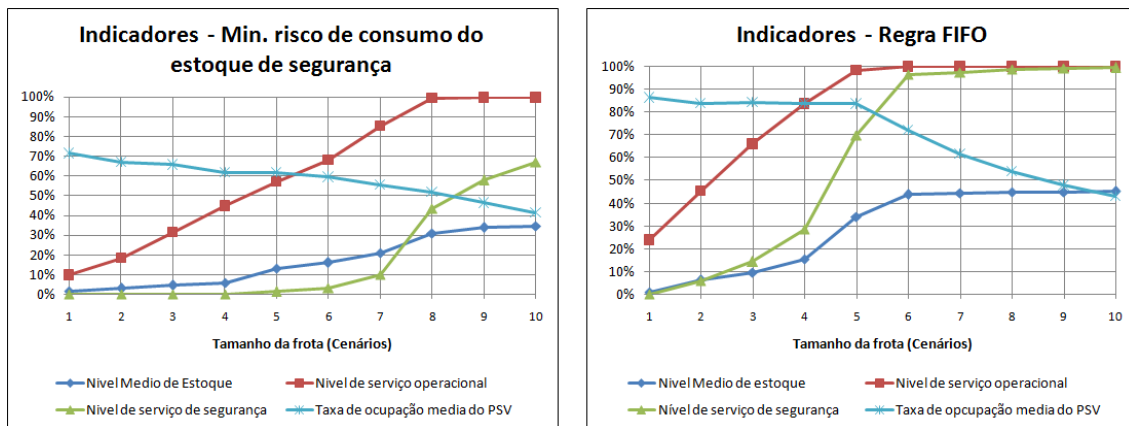


Figura 5.13 – comparação indicadores: Regra 1 do Experimento III

De acordo com os indicadores, verifica-se que os resultados da simulação com a regra de externa de minimização do risco do estoque de segurança são menos satisfatórios que a simulação com regra de seleção FIFO. Porém a taxa de ocupação da frota é menor. Isso sugere que os PSVs permanecem mais tempo ociosos no sistema. A ociosidade é consequência da frequência da chamada da programação externa. Essa programação é realizada a cada 24 horas e não dinamicamente como a regra FIFO realizada internamente ao simulador. Além disso, a formulação implica subutilizações da frota, pois em muitos casos é mais compensador não atender as demandas, visto que na função objetivo o peso da penalidade de não atendimento é igual a 1.

Além dos indicadores de desempenho, foram comparados os custos obtidos, conforme mostra a Figura 5.14.

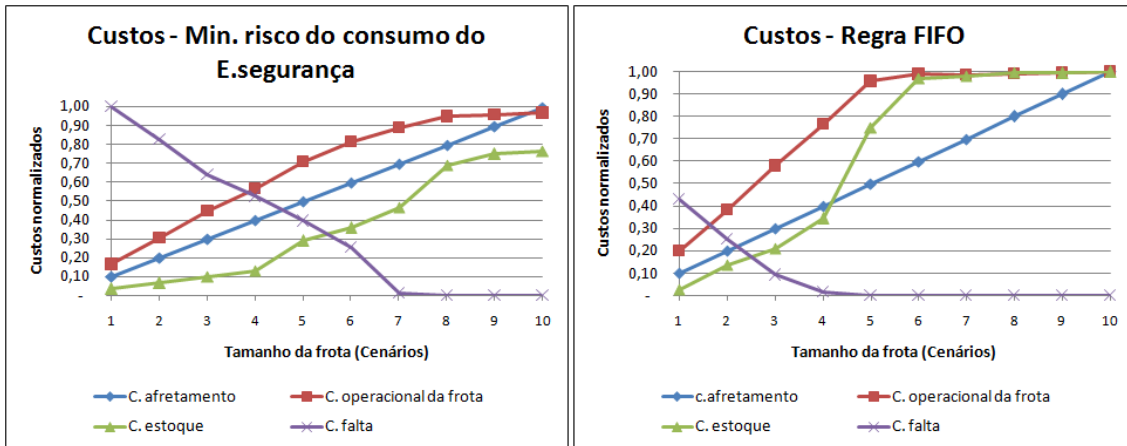


Figura 5.14 – Comparação dos custos: Regra 1 do Experimento III

Através do gráfico, observa-se que o custo de falta é muito maior quando se utiliza a programação externa. Isso se deve, mais uma vez, à frequência de chamada da programação e à não penalização do não atendimento da demanda na função objetivo. A comparação do custo total é mostrada na Figura 5.15.

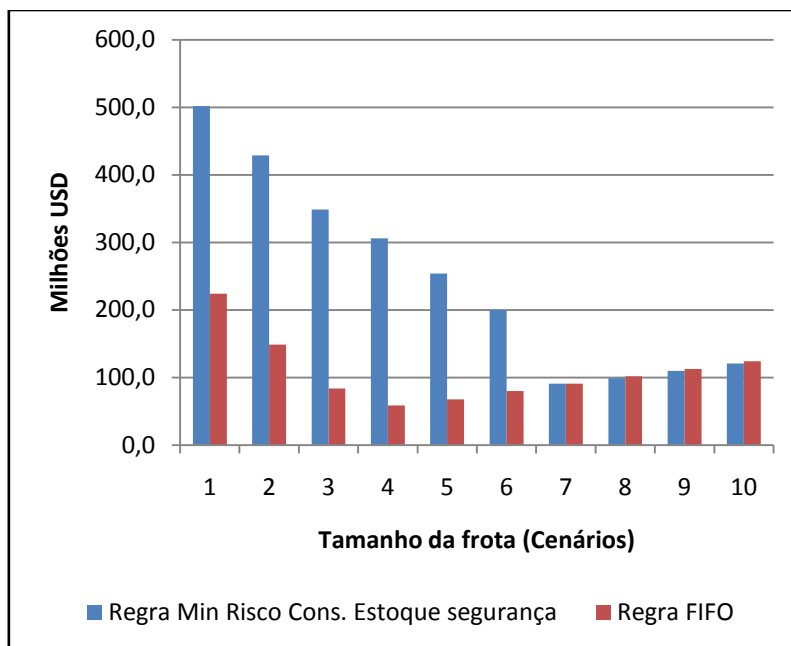


Figura 5.15 – comparação custo total: Regra1 do Experimento III

De acordo com o gráfico, o custo total resultante da simulação da Regra 1 do Experimento III é muito maior que o custo total resultante da simulação que utiliza a regra FIFO de alocação de PSVs para os Cenários de 1 a 6 PSVs. Isso ocorre devido às maiores faltas para esses cenários apresentadas na simulação da Regra 1. A partir do

Cenário 7 esse custos se aproximam e, além disso, o custo mínimo apresentado para a simulação da Regra 1 do Experimento III foi para o Cenário de 7 PSVs.

5.6.2 Regra FIFO versus maximização da autonomia dos clusters

A segunda comparação foi realizada entre a regra de maximização da autonomia (Regra 2) e a regra FIFO interna do simulador. Foram comparados os indicadores obtidos, conforme mostra a Figura 5.16.

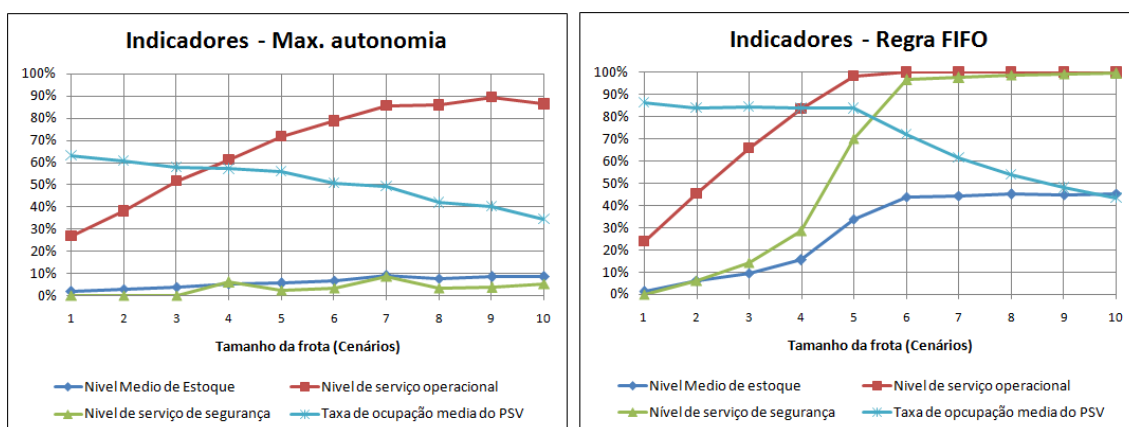


Figura 5.16 – comparação indicadores: Regra 2 do Experimento III

De acordo com os indicadores, mais uma vez verificam-se resultados menos satisfatórios quando se utiliza a programação externa. O nível médio de estoque se mantém muito baixo, assim como o nível de serviço de segurança. Além disso, o sistema nunca atinge 100% de nível de serviço operacional. Somente a taxa de utilização da frota apresenta resultados inferiores que os resultados da simulação com a regra FIFO. De novo, esse fato é explicado devido aos PSVs permanecerem mais tempo ociosos aguardando a chamada da programação. Além disso, também ocorre a subutilização da frota pois quando o nível de estoque é alto é preferível não atender a demanda, pois a penalidade de não atendimento na função objetivo é igual a 1. Logo, os atendimentos começam a ocorrer quando o nível de estoque baixa a um determinado ponto.

Também foram comparados os resultados dos custos da simulação com a programação externa relativa à Regra 2 e a simulação com regra FIFO como mostra a Figura 5.17.

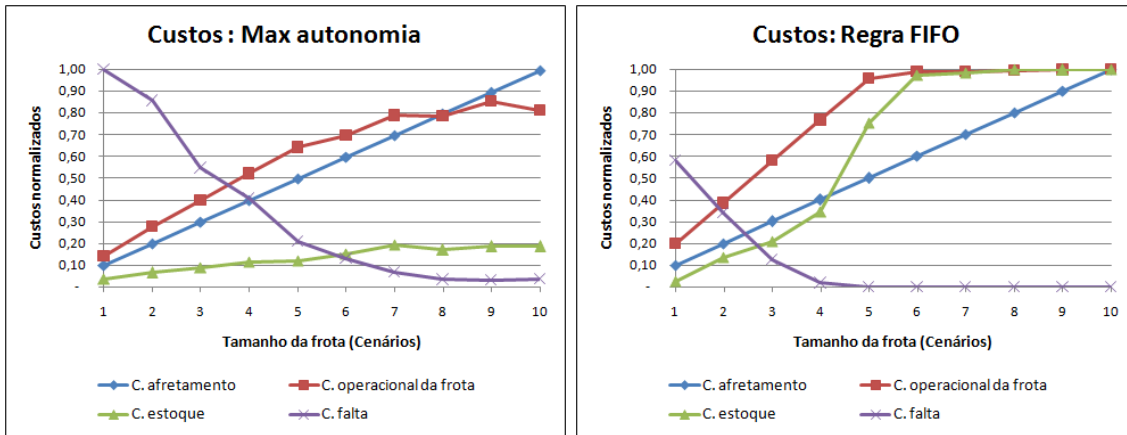


Figura 5.17 – comparação custos: Regra 2 do Experimento III

De acordo com os gráficos, todos os custos são menores na simulação com programação externa, exceto o custo de afretamento e o custo de falta. Mais uma vez, o alto custo de falta é explicado pela preferência da programação de não realizar atendimento quando os estoques dos *clusters* não estão baixos. O custo total é apresentado na Figura 5.18.

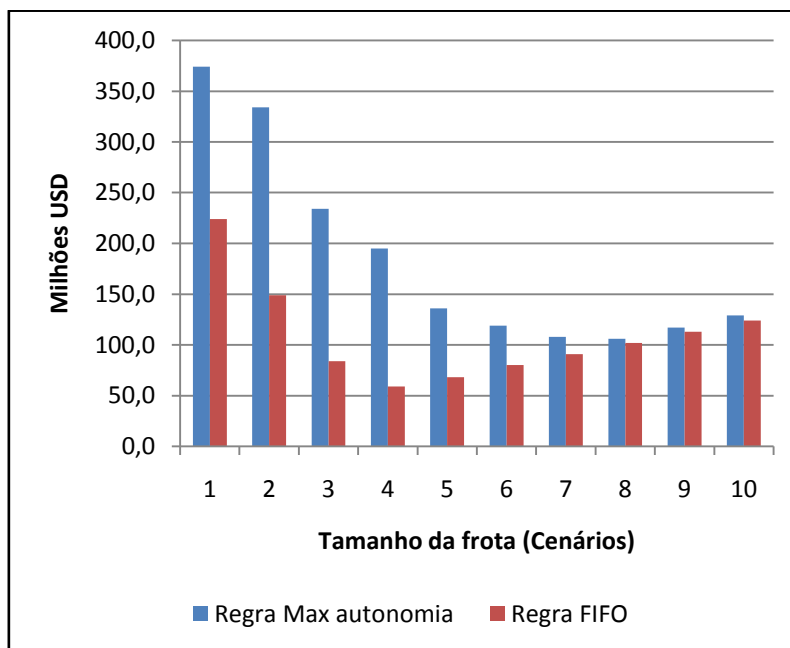


Figura 5.18 – comparação custo total regra2 exp.III

De acordo com o gráfico, o custo total resultante da simulação da Regra 2 do Experimento III é bem maior do que os custos resultantes da simulação com a regra FIFO de alocação de PSVs para os Cenários de 1 a 6 PSVs. Isso ocorre devido às

maiores faltas para esses cenários apresentadas na simulação da Regra 2. A partir do Cenário 7 os resultados de custo total se aproximam, porém, ainda assim, o custo total da simulação com a Regra 2 se apresenta superior ao custo total da simulação que utiliza a regra FIFO de alocação de PSVs.

5.6.3 Comparação dos resultados entre as duas formulações

Além de comparar os resultados obtidos das programações externas com a regra interna FIFO, foi feita a comparação dos resultados das duas regras de programação: a regra de minimização do risco do consumo do estoque de segurança (Regra 1) e a regra de maximização da autonomia (Regra 2). Primeiro foram comparados os indicadores obtidos nas duas simulações utilizando as programações externas conforme mostrado na Figura 5.19.

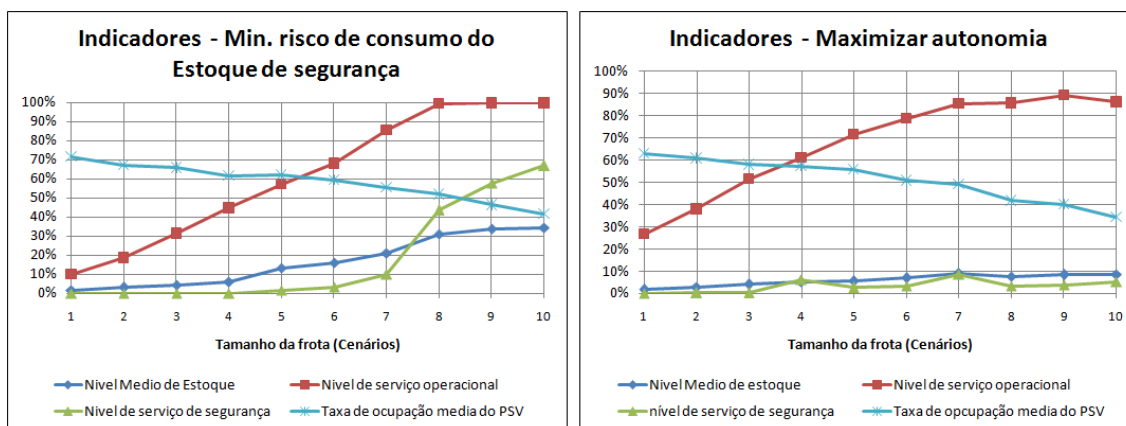


Figura 5.19 – comparação indicadores: Regra 1 e Regra 2 do Experimento III

Observando-se os gráficos é possível verificar que as taxas de utilização da frota são bem próximas para as duas regras de programação, sendo a taxa de ociosidade maior para a regra de maximização da autonomia. Além disso, a regra de minimização do risco de consumo do estoque de segurança apresenta resultados mais favoráveis visto que, com essa regra, atingi-se um nível de serviço operacional próximo a 100% a partir do Cenário 8, assim como um nível de serviço de segurança próximo a 70% no Cenário 10. Já a regra de maximização da autonomia atinge no máximo 90% de nível de serviço operacional e o nível de serviço de segurança se mantém próximo a zero em todos os cenários.

Além dos indicadores, foram comparados os custos obtidos a partir da rodada do simulador com as duas regras de programação, conforme mostrado na Figura 5.20.

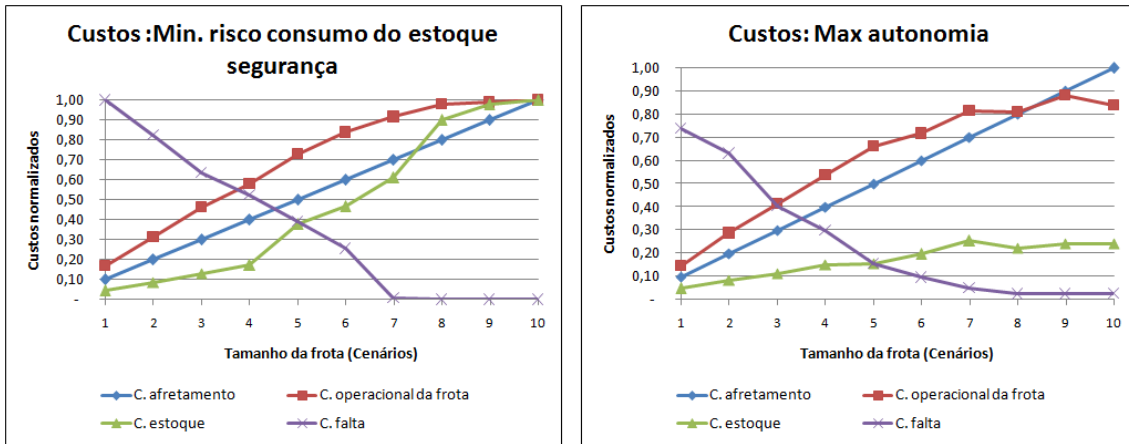


Figura 5.20 – comparação custos: Regra 1 e Regra 2 do Experimento III

De acordo com os gráficos de custos, a regra de minimização do risco de consumo do estoque de segurança apresenta um maior custo de falta nos Cenário de 1 a 7 PSVs. Além disso, o custo de estoque é muito menor na regra de maximização da autonomia. Este fato pode ser explicado pela preferência do não atendimento devido a não penalização na função objetivo. O custo total é mostrado na Figura 5.21.

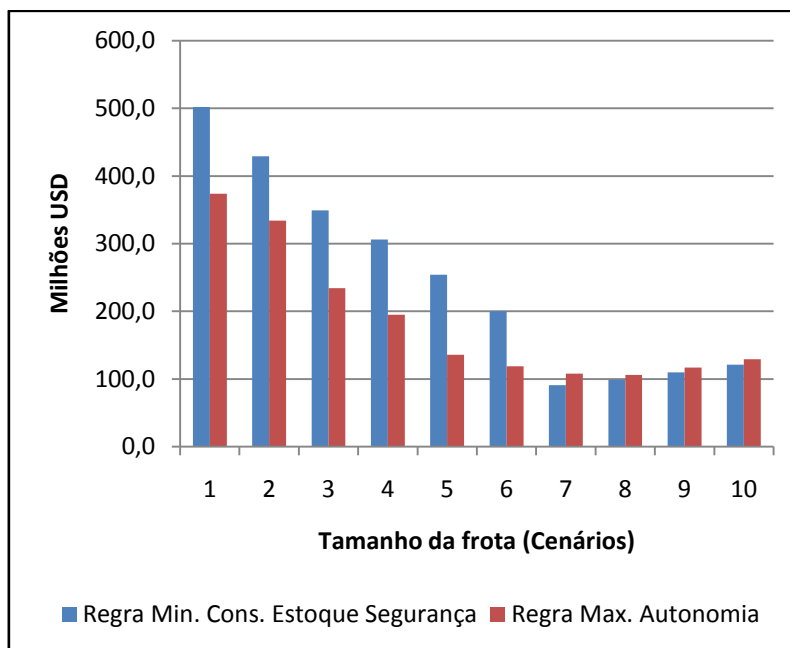


Figura 5.21 – comparação custo total regra 1 e regra 2 exp. III

O maior custo total observado com uso da regra de minimização do risco do consumo de segurança ainda pode ser explicado devido ao maior custo de falta apresentado nos Cenários de 1 a 6 PSVs. É interessante notar que a formulação da

função objetivo, considerando peso igual a 1 em ambas as parcelas, como descrito na Equação (3.10), prefere não atender aos *clusters* se os mesmos estiverem com níveis de estoque acima do estoque de segurança. Isso ocorre porque nessa situação a primeira parcela do somatório sempre será positiva. A partir do momento em que os níveis de estoque baixam do estoque de segurança, a primeira parcela torna a assumir valores negativos, o que faz com que o atendimento das demandas seja mais barato que o não atendimento.

6 Conclusão

A partir dos Experimentos I e II pode-se concluir que a política de entregas programadas é mais segura do ponto de vista de minimizar o risco de paradas. Foi possível mensurar esta segurança através do Experimento II, no qual se inclui o custo de exposição ao risco. Para determinar o tamanho ótimo da frota de PSVs oleiros para cada política analisada são considerados os indicadores de desempenho de nível de serviço e os custos do sistema. Dessa forma, para política de entregas sob demanda o custo mínimo é atingido no Cenário que considera 4 PSVs. Já para a política de entregas programadas, o custo mínimo é atingido no Cenário 5. Porém, desejando-se um nível de serviço operacional de 100%, ou seja, não admitindo faltas no sistema, a política de entregas sob demanda só é satisfatória a partir do Cenário 6, enquanto que a política de entregas programadas já atinge 100% de nível de serviço operacional com 5 PSVs. Analisando a segurança do sistema do ponto de vista do consumo do estoque de segurança, pode-se concluir que a política de entregas programadas atinge um nível de serviço de segurança de 100% a partir de 7 PSVs. Por outro lado, a política de entregas sob demanda atinge, no máximo 99,5% no Cenário de 10 PSVs. Dessa forma, buscando-se operações eficientes e seguras, a política de entregas programadas se revela mais adequada para o cenário produtivo da companhia estudada, considerando as características de consumo analisadas, visto que reúne melhores resultados de custo e nível de serviço. De acordo com o Experimento I, enquanto que na política de entregas programadas, o sistema atinge 100% de nível de serviço de segurança no Cenário 7 com custo total de US\$ 97 Mi, na política de entregas sob demanda o sistema atinge 99,5% de nível de serviço de segurança no Cenário 10 com custo total de US\$ 124 Mi. Considerando o custo de exposição ao risco o custo do Cenário 7 para política de entregas programadas é de US\$ 98 Mi enquanto que para política de entregas sob demanda é de US\$ 126 Mi, portanto mais cara. Além disso, do ponto de vista do planejamento gerencial, a política de entregas programadas é mais recomendada, pois, com ela, é possível seguir um cronograma de atendimento previamente estabelecido.

O Experimento III permite incluir um código externo ao simulador para realizar as alocações. As experiências da simulação com outras regras de alocação, através de aplicação de programação inteira mista, se mostrou eficiente na questão do tempo de execução. Porém, os resultados obtidos com o uso de regras externas de alocação se mostraram menos satisfatórios que os resultados obtidos enquanto as alocações foram

realizadas na lógica de programação interna ao simulador. Isso pode ser explicado devido ao fato de que em alguns casos pode ser mais barato para função objetivo não atender nenhum *cluster*. Isso faz aumentar o custo de falta. Um possível contorno para o problema seria atribuir pesos para as duas parcelas da função objetivo, que tratam as duas variáveis de decisão, evitando a preferência de não atendimentos das demandas. Além disso, adotar a estratégia de formulação minimax para minimizar o máximo das diferenças entre os níveis de estoque dos *clusters* e seus estoques de segurança pode ser válido para evitar grandes faltas de estoque no sistema. Outras abordagens possíveis para o problema de alocação de PSVs são a programação multi-estágio e a otimização estocástica.

Por fim, o estudo se mostrou útil para avaliar políticas de suprimento de óleo diesel a unidades marítimas de exploração e produção de petróleo e determinar a frota ideal de navios supridores. Portanto, o modelo pode ser usado para auxiliar decisões logísticas estratégicas aplicadas em outros cenários de produção *offshore*.

6.1 Trabalhos futuros

O trabalho realizado visa auxiliar a escolha da melhor política de suprimento e o dimensionamento ideal da frota a partir de fatores estocásticos das taxas de consumo e volumes de pedido. Assim, dentro do âmbito proposto do trabalho e considerando a continuação do método de simulação, algumas considerações podem ser feitas sobre expectativas de trabalhos futuros. Uma delas é incluir outros fatores de incertezas como tempos de navegação e operação, visto que esses tempos sofrem influências estocásticas decorrentes das condições climáticas. Além disso, incluir as regras operacionais dos navios tanque e tratar as UMs individualmente pode ampliar a possibilidade de análises e permitir inferências mais precisas. Testar outras distribuições de consumo das UMs pode ser útil para analisar como o aumento da aleatoriedade do sistema influi nos resultados. Por fim, é necessário aprimorar a implementação do código externo de alocação de PSVs, sendo interessante reformular as funções objetivo e a frequência de programação.

7 Referências

AAS, B. **Upstream Logistics in Offshore Petroleum Production & Management of logistics planning**. [s.l.] Molde University College, 2008.

AAS, B.; GRIBKOVSKAIA, I.; ØYVIND, H.; SHLOPAK, A. Routing of supply vessels to petroleum installations. **International Journal of Physical Distribution & Logistics Management**, v. 37, n. 2, p. 164–179, 2007.

ANP. **Sistema de levantamentos de preços**. Disponível em: <http://www.anp.gov.br/preco/prc/Resumo_Por_Municipio_Posto.asp>. Acesso em: 15 jun. 2016a.

ANP. **Dados de E&P**. Disponível em: <<http://www.anp.gov.br/wwwanp/exploracao-e-producao-de-oleo-e-gas/gestao-de-contratos-de-e-p/dados-de-e-p>>. Acesso em: 16 jun. 2016b.

BANKS, J. **Introduction to simulation**. (H. B. FARRINGTON, D. T. NEMBHARD, G. W. EVANS, Eds.) Winter Simulation Conference. **Anais...**1999

BATISTA, B. C. D. **Análise Das Operações Com Embarcações De Apoio Offshore Na Bacia De Campos - Rj**. [s.l.] Universidade Federal do Rio de Janeiro, RJ, 2005.

BM&FBOVESPA. **BM&FBOVESPA**. Disponível em: <http://lojavirtual.bmf.com.br/lojaie/portal/pages/pdf/apostila_pqo_cap_01_v2.pdf>. Acesso em: 10 jun. 2016.

CARSON, J. S. **Introduction to Modeling and Simulation**. Proceedings of the Winter Simulation Conference, 2005. **Anais...**2005 Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-33846666685&partnerID=tZOtx3y1>>

CHWIF, L.; MEDINA, A. C. **Modelagem e simulação de eventos discretos**. 3. ed. Rio de Janeiro: Elsevier, 2013.

FAGERHOLT, K.; LINDSTAD, H. Optimal policies for maintaining a supply service in the Norwegian Sea. **Omega**, v. 28, n. 3, p. 269–275, 2000.

FERREIRA FILHO, V. J. M. **Gestão de Operações e Logística na Produção de Petróleo: fundamentos, metodologia e modelos quantitativos**. 1. ed. Rio de Janeiro: Elsevier, 2016.

FULLER, D. B.; DE ARRUDA, E. F.; FERREIRA FILHO, V. J. M. **Determinação de frota ótima de aliviadores para um conjunto de plataformas**. XLV SBPO. **Anais...**Natal, RN: SOBRAPO, 2013

HALVORSEN-WEARE, E. E.; FAGERHOLT, K.; NONÅS, L. M.; ASBJORNSLETT, B. E. Optimal fleet composition and periodic routing of offshore supply vessels. **European Journal of Operational Research**, v. 223, n. 2, p. 508–517, 2012.

HALVORSEN-WEARE, E. E.; FAGERHOLT, K.; RÖNNQVIST, M. Vessel routing and scheduling under uncertainty in the liquefied natural gas business. **Computers and Industrial Engineering**, v. 64, n. 1, p. 290–301, 2013.

HILLIER, F. S.; LIEBERMAN, G. J. **Introdução à pesquisa operacional**. 8. ed. São Paulo: McGraw-Hill, 2008.

ILOS. **Logística de Apoio Offshore - Integração e sincronização da cadeia de atendimento às unidades marítimas**. Disponível em:

<<http://www.ilos.com.br/web/logistica-de-apoio-offshore-integracao-e-sincronizacao-da-cadeia-de-atendimento-as-unidades-maritimas/>>. Acesso em: 9 fev. 2016.

KAISER, M. J. An integrated systems framework for service vessel forecasting in the Gulf of Mexico. **Energy**, v. 35, n. 7, p. 2777–2795, 2010.

KAISER, M. J.; SNYDER, B. An empirical analysis of offshore service vessel utilization in the US Gulf of Mexico. **International Journal of Energy Sector Management**, v. 4, n. 2, p. 152–182, 2010.

KO, H. J.; KO, C. S.; KIM, T. A hybrid optimization/simulation approach for a distribution network design of 3PLS. **Computers and Industrial Engineering**, v. 50, n. 4, p. 440–449, 2006.

LEITE, R. P. **Maritime transport of deck cargo to Petrobras fields in Campos Basin : an empirical analysis , identification and quantification of improvement points**. [s.l.] PUC, Rio, 2012.

LIMOEIRO, C. D. P.; FULLER, D. B.; PEREIRA, B.; BARBOSA, G. DE A. Processo de dimensionamento por simulação de um sistema integrado de suprimento de petróleo para um conjunto regional de refinarias. **Anais do XL SBPO**, p. 1982–1993, 2008.

LOPES, J. DA S. C.; DA CUNHA, C. B. Um modelo integrado de simulação-otimização para avaliação de um novo negócio de aeronaves de propriedade compartilhada. **Journal of Tansport Literature**, v. 6, p. 8–37, 2012.

MAISIUK, Y.; GRIBKOVSKAIA, I. Fleet sizing for offshore supply vessels with stochastic sailing and service times. **Procedia Computer Science**, v. 31, p. 939–948, 2014.

MELOUK, S. H.; FREEMAN, N. K.; MILLER, D.; DUNNING, M. Simulation optimization-based decision support tool for steel manufacturing. **International Journal of Production Economics**, v. 141, n. 1, p. 269–276, 2013.

PINTO, J. A. **Uso da simulação de sistemas dinâmicos para análise de logística hospitalar: gestão integrada de estoque e capacidade**. [s.l.] UNICAMP, SP, 2007.

RIBEIRO, F. R. **Modelo de simulação para análise operacional de pátio de aeroportos**. [s.l.] Universidade de São Paulo, SP, 2003.

RIBEIRO, L. S.; IACHAN, R. **Dimensionamento Da Frota De Supridores Para Plataformas De Petróleo**. XLI SBPO. **Anais...SOBRAPO**, 2009

SARGENT, R. G. Verification and validation of simulation models. **Proceedings of the 37th conference on Winter simulation**, v. 7, n. 1, p. 130–143, 2005.

SHYSHOU, A.; GRIBKOVSKAIA, I.; BARCELÓ, J. A simulation study of the fleet sizing problem arising in offshore anchor handling operations. **European Journal of Operational Research**, v. 203, n. 1, p. 230–240, 2010.

SILVA, R. C. DOS S. **Análises de tradeoff em sistemas de transporte aquaviário com simulação de eventos discretos**. [s.l.] Universidade de São Paulo, 2012.

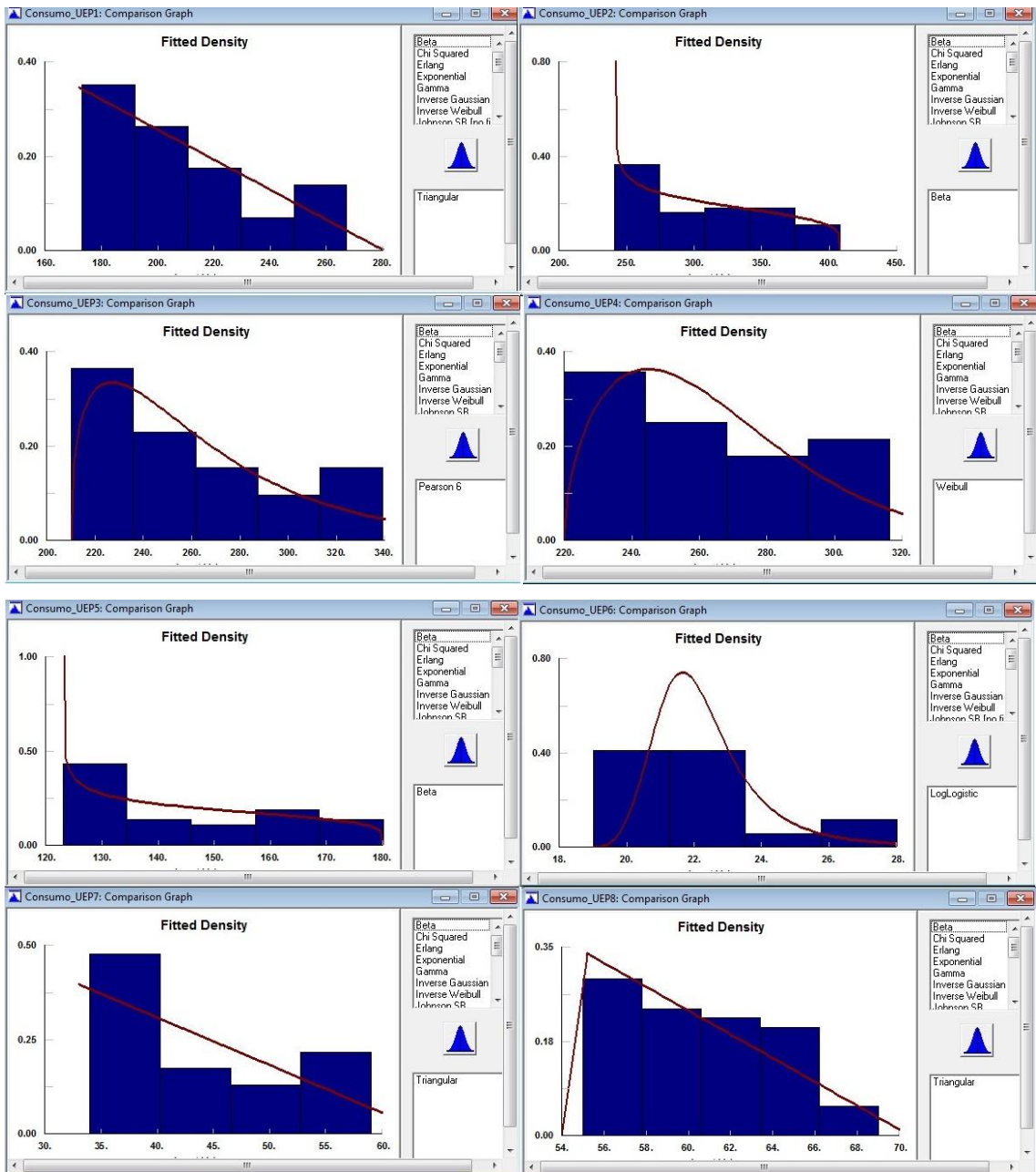
SILVA, R. C. DOS S.; BRITO, T. B.; BOTTER, R. C.; PEREIRA, N. N. **Modeling of a Closed-Loop Maritime Transportation System with Discrete Event Simulation and Multi-Criteria Decision Analysis**. World Congress on Engeneering and computer Science. **Anais...San Francisco, EUA: WCECS**, 2011

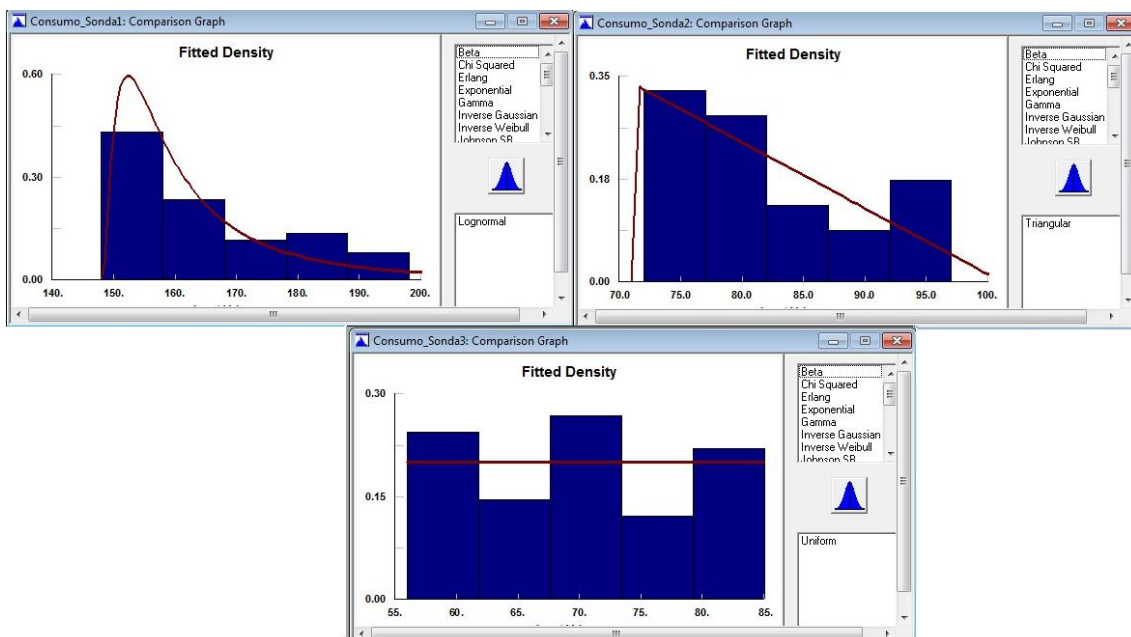
SILVA, V.; LOPES, M.; ARBEX, F.; LOPES, P. **Modelagem e simulação de um sistema portuário: um estudo de caso para o terminal da petrobras na cidade de**

- macaé.** Rio Oil & Gas Expo and Conference. **Anais...**Rio de Janeiro, RJ: IBP, 2010
- SONG, B. L.; WONG, W. K.; FAN, J.; CHAN, S. F. Integration Simulation of Intelligent Real-time Optimization Decision Support System for Assembly Line Balancing. **Wcecs 2008: World Congress on Engineering and Computer Science**, p. 1105–1112, 2008.
- UGLANE, V. T.; FRIEDBERG, D. Ø.; ANDERSSON, H. **Routing and Scheduling of Platform Supply Vessels**. [s.l.] Norwegian University of Science and Technology, 2012.
- ULSTEIN, M. **A simulation-based decision support tool for arctic field logistics**. [s.l.] Norwegian University of Science and Technology, 2014.
- WANKE, P. F. Estratégia de Posicionamento Logístico: Conceitos, Implicações e Análise da Realidade Brasileira. n. May, 2016.

Apêndice A – Curvas de Aderência Consumo

As figuras que seguem representam o histograma dos dados de entrada de consumo e as curvas de distribuição de probabilidade ajustadas pelo *software* Statfit® referentes a cada um dos 11 *clusters*. Na ordem, da direita para esquerda encontram-se representados as curvas dos *clusters* UEP1, UEP2, UEP3, UEP4, UEP5, UEP6, UEP7, UEP8, Sonda1, Sonda 2 e Sonda3.





Apêndice B – Estatísticas Descritivas

A tabela que segue apresenta as estatísticas descritivas dos dados de entrada coletados para os consumos dos *clusters*.

Dados de Consumo

Cluster	Nº de observações	Mínimo	Máximo	Média	Desvio-padrão
UEP1	57	173	267	208.2	27.42
UEP2	55	241	408	309.3	51.05
UEP3	52	210	339	259.6	37.95
UEP4	28	220	316	263.4	29.09
UEP5	37	123	180	145.4	18.60
UEP6	17	19	28	22.2	2.36
UEP7	23	34	59	43.6	8.67
UEP8	55	55	69	60.3	3.94
Sonda1	51	148	198	164.3	14.10
Sonda2	46	72	97	81.5	7.73
Sonda3	41	56	85	70.0	9.15

A tabela que segue apresenta as estatísticas descritivas dos dados de entrada coletados para os volumes de pedido ou entrega referentes a cada *cluster*.

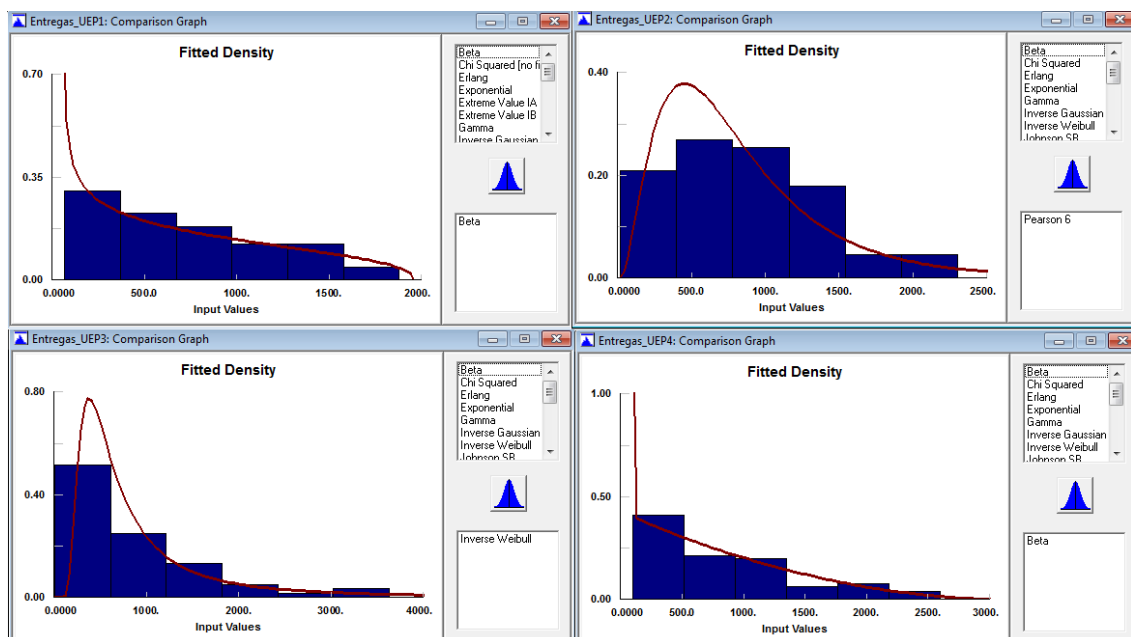
Dados de Entregas

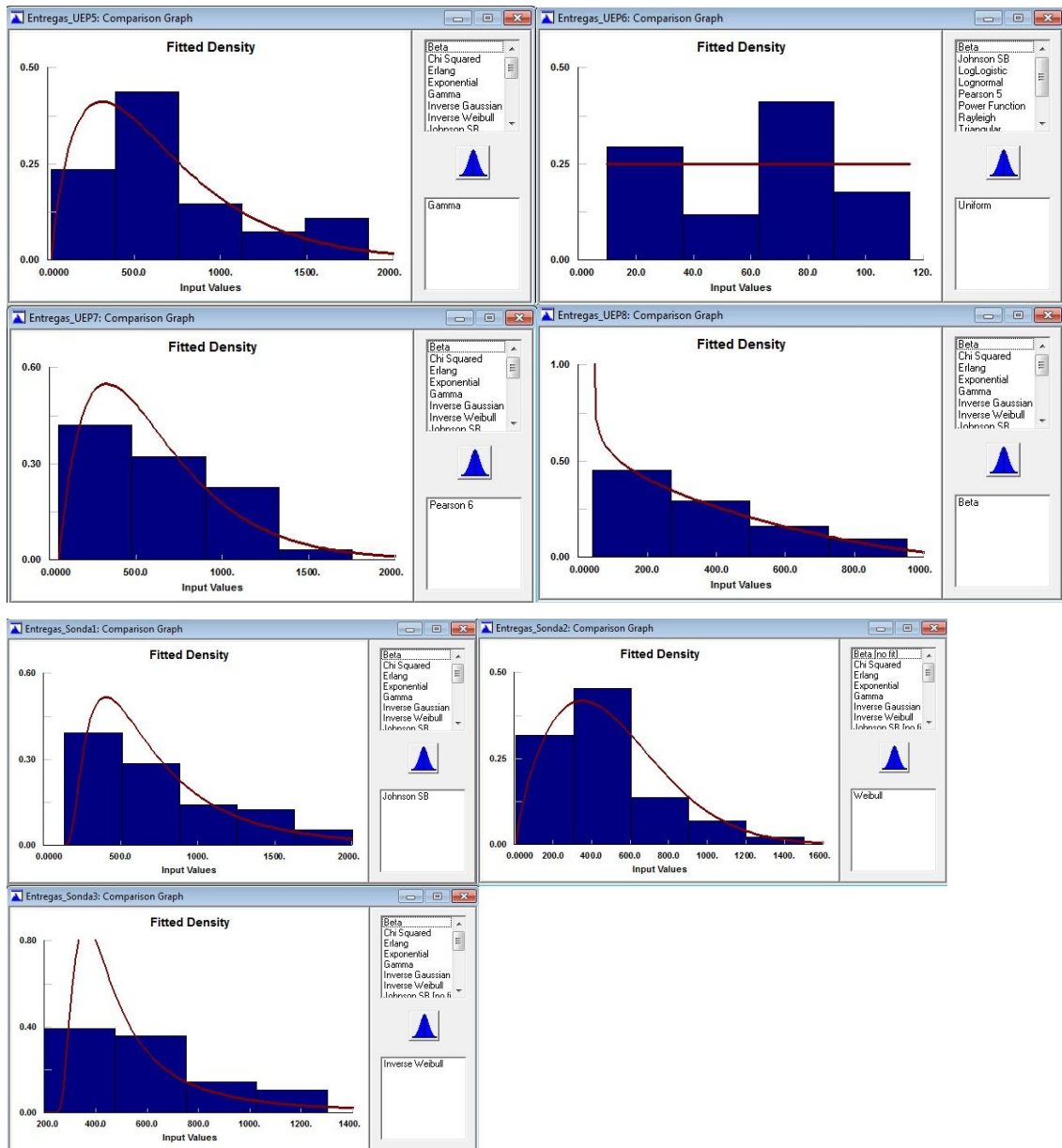
Cluster	Nº de observações	Mínimo	Máximo	Média	Desvio-padrão
UEP1	66	67.2	1877	701.9	508.9
UEP2	67	16	2300	823.7	515.6
UEP3	60	3	3626	930.8	746.1

UEP4	80	100	2598	854.9	587.4
UEP5	55	20	1855	668.2	450.24
UEP6	17	10	115	62.8	32.68
UEP7	31	50	1750	614.5	390.24
UEP8	31	42	950	327.8	242.11
Sonda1	56	135	2000	756.6	443.2
Sonda2	44	7	1500	498.0	310.7
Sonda3	28	200	1300	574.1	295.1

Apêndice C – Curvas de Aderência Entregas

As figuras que seguem representam o histograma dos dados de entrada de volume de pedido/entrega e as curvas de distribuição de probabilidade ajustadas pelo *software* Statfit® referentes a cada um dos 11 *clusters*. Na ordem, da direita para esquerda encontram-se representados as curvas dos *clusters* UEP1, UEP2, UEP3, UEP4, UEP5, UEP6, UEP7, UEP8, Sonda1, Sonda 2 e Sonda3.





Apêndice D – Blocos de Processos e Direcionamento

Tabela 31 – Processos e direcionamento de entidades

Entidade	Local	Processo	Possíveis locais de destino
"eSetUp_Consumo" "eAtendimento" "ePedido_OD"	SetUp	Processo SetUp	Saída
"ePSV_OD"	NT_Fundeio	Processo	UEP1

		NT_Fundeio	UEP2
			UEP3
			UEP4
			UEP5
			UEP6
			UEP7
			UEP8
			Sonda1
			Sonda2
			Sonda3
"ePSV_OD"	NT_Fila	-	NT
"ePSV_OD"	NT	Processo NT	NT_Fundeio
"ePSV_OD"	UEP1	Processo_Cluster	UEP2
			UEP3
			UEP4
			UEP5
			UEP6
			UEP7
			UEP8
			Sonda1
			Sonda2
			Sonda3
			NT_Fila
"ePSV_OD"	UEP2	Processo_Cluster	UEP1
			UEP3
			UEP4
			UEP5
			UEP6
			UEP7
			UEP8
			Sonda1
			Sonda2
			Sonda3
			NT_Fila
"ePSV_OD"	UEP3	Processo_Cluster	UEP1
			UEP2
			UEP4
			UEP5
			UEP6
			UEP7
			UEP8
			Sonda1
			Sonda2
			Sonda3
			NT_Fila

"ePSV_OD"	UEP4	Processo_Cluster	UEP1
			UEP2
			UEP3
			UEP5
			UEP6
			UEP7
			UEP8
			Sonda1
			Sonda2
			Sonda3
			NT_Fila
"ePSV_OD"	UEP5	Processo_Cluster	UEP1
			UEP2
			UEP3
			UEP4
			UEP6
			UEP7
			UEP8
			Sonda1
			Sonda2
			Sonda3
			NT_Fila
"ePSV_OD"	UEP6	Processo_Cluster	UEP1
			UEP2
			UEP3
			UEP4
			UEP5
			UEP7
			UEP8
			Sonda1
			Sonda2
			Sonda3
			NT_Fila
"ePSV_OD"	UEP7	Processo_Cluster	UEP1
			UEP2
			UEP3
			UEP4
			UEP5
			UEP6
			UEP8
			Sonda1
			Sonda2
			Sonda3
			NT_Fila
"ePSV_OD"	UEP8	Processo_Cluster	UEP1

			UEP2
			UEP3
			UEP4
			UEP5
			UEP6
			UEP7
			Sonda1
			Sonda2
			Sonda3
			NT_Fila
"ePSV_OD"	Sonda1	Processo_Cluster	UEP1
"ePSV_OD"	Sonda1	Processo_Cluster	UEP2
"ePSV_OD"	Sonda1	Processo_Cluster	UEP3
"ePSV_OD"	Sonda1	Processo_Cluster	UEP4
"ePSV_OD"	Sonda1	Processo_Cluster	UEP5
"ePSV_OD"	Sonda1	Processo_Cluster	UEP6
"ePSV_OD"	Sonda1	Processo_Cluster	UEP7
"ePSV_OD"	Sonda1	Processo_Cluster	UEP8
"ePSV_OD"	Sonda1	Processo_Cluster	Sonda2
"ePSV_OD"	Sonda1	Processo_Cluster	Sonda3
"ePSV_OD"	Sonda1	Processo_Cluster	NT_Fila
"ePSV_OD"	Sonda2	Processo_Cluster	UEP1
"ePSV_OD"	Sonda2	Processo_Cluster	UEP2
"ePSV_OD"	Sonda2	Processo_Cluster	UEP3
"ePSV_OD"	Sonda2	Processo_Cluster	UEP4
"ePSV_OD"	Sonda2	Processo_Cluster	UEP5
"ePSV_OD"	Sonda2	Processo_Cluster	UEP6
"ePSV_OD"	Sonda2	Processo_Cluster	UEP7
"ePSV_OD"	Sonda2	Processo_Cluster	UEP8
"ePSV_OD"	Sonda2	Processo_Cluster	Sonda1
"ePSV_OD"	Sonda2	Processo_Cluster	Sonda3
"ePSV_OD"	Sonda2	Processo_Cluster	NT_Fila
"ePSV_OD"	Sonda3	Processo_Cluster	UEP1
"ePSV_OD"	Sonda3	Processo_Cluster	UEP2
"ePSV_OD"	Sonda3	Processo_Cluster	UEP3
"ePSV_OD"	Sonda3	Processo_Cluster	UEP4
"ePSV_OD"	Sonda3	Processo_Cluster	UEP5
"ePSV_OD"	Sonda3	Processo_Cluster	UEP6
"ePSV_OD"	Sonda3	Processo_Cluster	UEP7
"ePSV_OD"	Sonda3	Processo_Cluster	UEP8
"ePSV_OD"	Sonda3	Processo_Cluster	Sonda1
"ePSV_OD"	Sonda3	Processo_Cluster	Sonda2
"ePSV_OD"	Sonda3	Processo_Cluster	NT_Fila

Apêndice E – Códigos do Simulador

A seguir são apresentados os principais códigos desenvolvidos para construção do modelo de simulação.

Processo SetUp

```
If Entity() = eAtendimento And vModelo_SobDemanda = 0 Then //Usado para colocar atendimento na
fila de aguardo.
{
  If mAtendimentos_Aguardando[aID_Cluster] = 0 Then mHora_Surge_Atendimento[aID_Cluster] =
clock()
  Inc mAtendimentos_Aguardando[aID_Cluster]
  If clock() >= vTempo_Aquecimento Then Inc mAtendimentos_Programados[aID_Cluster]
  Inc vCont_Entregas_tobe
}
Else If Entity() = eSetup_Consumo_Cluster Then //Usado para começar e atualizar o consumo de diesel e
verificar nível do tanque.
{
  If aID_Cluster = 1 Then
  {
    if Tank_Level[mID_TQ_Cluster[1]] > vc1 Then
    {
      Tank_Empty(UEP1_TQ_OD, vc1, vc1, 0.1)
    }
  }
  Else If aID_Cluster = 2 Then
  {
    If Tank_Level[mID_TQ_Cluster[2]] > vc2 Then
    {
      Tank_Empty(Sonda1_TQ_OD, vc2, vc2, 0.1)
    }
  }
  Else If aID_Cluster = 3 Then
  {
    If Tank_Level[mID_TQ_Cluster[3]] > vc3 Then
    {
      Tank_Empty(UEP2_TQ_OD, vc3, vc3, 0.1)
    }
  }
  Else If aID_Cluster = 4 Then
  {
    if Tank_Level[mID_TQ_Cluster[4]] > vc4 Then
    {
      Tank_Empty(UEP3_TQ_OD, vc4, vc4, 0.1)
    }
  }
  Else If aID_Cluster = 5 Then
  {
    if Tank_Level[mID_TQ_Cluster[5]] > vc5 Then
    {
      Tank_Empty(UEP4_TQ_OD, vc5, vc5, 0.1)
    }
  }
  Else If aID_Cluster = 6 Then
```

```

{
  if Tank_Level[mID_TQ_Cluster[6]] > vc6 Then
  {
    Tank_Empty(UEP5_TQ_OD, vc6, vc6, 0.1)
  }
}
Else If aID_Cluster = 7 Then
{
  if Tank_Level[mID_TQ_Cluster[7]] > vc7 Then
  {
    Tank_Empty(UEP6_TQ_OD, vc7, vc7, 0.1)
  }
}
Else If aID_Cluster = 8 Then
{
  If Tank_Level[mID_TQ_Cluster[8]] > vc8 Then
  {
    Tank_Empty(UEP7_TQ_OD, vc8, vc8, 0.1)
  }
}
Else If aID_Cluster = 9 Then
{
  If Tank_Level[mID_TQ_Cluster[9]] > vc9 Then
  {
    Tank_Empty(UEP8_TQ_OD, vc9, vc9, 0.1)
  }
}
Else If aID_Cluster = 10 Then
{
  If Tank_Level[mID_TQ_Cluster[10]] > vc10 Then
  {
    Tank_Empty(Sonda2_TQ_OD,vc10, vc10, 0.1)
  }
}
Else If aID_Cluster = 11 Then
{
  vCont_Cluster = 0
  If Tank_Level[mID_TQ_Cluster[11]] > vc11 Then
  {
    Tank_Empty(Sonda3_TQ_OD, vc11, vc11, 0.1)
  }
}
}
Else If Entity() = ePedido_OD And vModelo_SobDemanda = 1 Then
{
  Int vlcont = 1
  While vlcont <= 11 Do
  {
    If mCluster_Solicitante[vlcont] > 0 Then
    {
      //Create 1 as ePedido_OD
      vTem_Pedido = 1
      aID_Cluster = vlcont
      If clock() >= vTempo_Aquecimento Then
      {
        Inc vID_Pedido_pos_aquecimento
        Inc mPedidos_Gerados_pos_aquecimento[aID_Cluster]
      }
      Inc vID_Pedido
      Inc mPedidos_Gerados[aID_Cluster]
    }
  }
}

```



```

        mStatus_Pedidos[mPedidos_Gerados[aID_Cluster], vCol_Horario_Pedido_mStatus_Pedidos,
aID_Cluster] = Clock()
        mStatus_Pedidos[mPedidos_Gerados[aID_Cluster], vCol_Status_Pedido_mStatus_Pedidos,
aID_Cluster] = vPedido_Em_Aberto
        mStatus_Pedidos[mPedidos_Gerados[aID_Cluster], vCol_Ultima_Fracao_mStatus_Pedidos,
aID_Cluster] = 0
        mStatus_Pedidos[mPedidos_Gerados[aID_Cluster], vCol_Entrega_Parcial, aID_Cluster] = 0
        aVolume_OD_Pedido = mDistribuicao_Tamanho_Pedido_IN[vlcont]
        Inc mPedidos_Aguardando[aID_Cluster], aVolume_OD_Pedido
        vVolume_Pedido = aVolume_OD_Pedido
        mVerifica_Pedidos[aID_Cluster] = aVolume_OD_Pedido
        Dec mCluster_Solicitante[vlcont]
    }
    Inc vlcont
}
}

```

Processo NT_Fundeio

```

If vModelo_SobDemanda = 0 Then //Lógica modelo Entregas Programdas.
{
    //Ver se tem atendimento e coletar, caso tenha.
    If aRetorno = 1 Then
    {
        Wait Until vPSVs_Fundeio > 0
        mTempo_Livra_PSV[aID_PSV_OD] = clock()
        If clock() >= vTempo_Aquecimento and mHora_Saida_FundeioNT[mEntrega_PSV[aID_PSV_OD],
aID_PSV_OD] >= vTempo_Aquecimento Then
        {
            mTempo_Uso_PSV[mEntrega_PSV[aID_PSV_OD],aID_PSV_OD] = clock() -
mHora_Saida_FundeioNT[mEntrega_PSV[aID_PSV_OD], aID_PSV_OD]
            Inc mTempo_Uso_PSV_Total[aID_PSV_OD],
mTempo_Uso_PSV[mEntrega_PSV[aID_PSV_OD],aID_PSV_OD]
            mTempo_Ciclo[mEntrega_PSV[aID_PSV_OD],aID_PSV_OD] = clock() -
mHora_Saida_FundeioNT[mEntrega_PSV[aID_PSV_OD], aID_PSV_OD]
            Inc mSoma_Tempo_Ciclo[aID_PSV_OD],
mTempo_Ciclo[mEntrega_PSV[aID_PSV_OD],aID_PSV_OD]
            Inc mNumero_Visitas_FundeioNT[aID_PSV_OD]
            Inc vCont_Ciclos
        }
    }
    L2:
    Wait Until vPSVs_Fundeio > 0 And ((mAtendimentos_Aguardando[1] > 0 And
((mVerifica_Barco_TopUp[1,1] > 0 And mVerifica_Barco_TopUp[1,2] = 0) Or
mVerifica_Barco_TopUp[1,1] = 0)) Or
        (mAtendimentos_Aguardando[2] > 0 And ((mVerifica_Barco_TopUp[2,1] > 0
And mVerifica_Barco_TopUp[2,2] = 0) Or mVerifica_Barco_TopUp[2,1] = 0)) Or
        (mAtendimentos_Aguardando[3] > 0 And ((mVerifica_Barco_TopUp[3,1] > 0
And mVerifica_Barco_TopUp[3,2] = 0) Or mVerifica_Barco_TopUp[3,1] = 0)) Or
        (mAtendimentos_Aguardando[4] > 0 And ((mVerifica_Barco_TopUp[4,1] > 0
And mVerifica_Barco_TopUp[4,2] = 0) Or mVerifica_Barco_TopUp[4,1] = 0)) Or
        (mAtendimentos_Aguardando[5] > 0 And ((mVerifica_Barco_TopUp[5,1] > 0
And mVerifica_Barco_TopUp[5,2] = 0) Or mVerifica_Barco_TopUp[5,1] = 0)) Or
        (mAtendimentos_Aguardando[6] > 0 And ((mVerifica_Barco_TopUp[6,1] > 0
And mVerifica_Barco_TopUp[6,2] = 0) Or mVerifica_Barco_TopUp[6,1] = 0)) Or
        (mAtendimentos_Aguardando[7] > 0 And ((mVerifica_Barco_TopUp[7,1] > 0
And mVerifica_Barco_TopUp[7,2] = 0) Or mVerifica_Barco_TopUp[7,1] = 0)) Or
        (mAtendimentos_Aguardando[8] > 0 And ((mVerifica_Barco_TopUp[8,1] > 0
And mVerifica_Barco_TopUp[8,2] = 0) Or mVerifica_Barco_TopUp[8,1] = 0)) Or

```

```

(mAtendimentos_Aguardando[9] > 0 And ((mVerifica_Barco_TopUp[9,1] > 0
And mVerifica_Barco_TopUp[9,2] = 0) Or mVerifica_Barco_TopUp[9,1] = 0)) Or
(mAtendimentos_Aguardando[10] > 0 And ((mVerifica_Barco_TopUp[10,1] >
0 And mVerifica_Barco_TopUp[10,2] = 0) Or mVerifica_Barco_TopUp[10,1] = 0)) Or
(mAtendimentos_Aguardando[11] > 0 And ((mVerifica_Barco_TopUp[11,1] >
0 And mVerifica_Barco_TopUp[11,2] = 0) Or mVerifica_Barco_TopUp[11,1] = 0)))

//Escolha do cluster a ser atendido (método de maior atrasos).
int vlCont = 1
int vlEscolha
Real vlMaior = 0
While vlCont <= 11 Do
{
If mAtendimentos_Aguardando[vlCont] > vlMaior And ((mVerifica_Barco_TopUp[vlCont, 1] > 0
and mVerifica_Barco_TopUp[vlCont,2] = 0) or (mVerifica_Barco_TopUp[vlCont, 1] = 0)) Then
{
vlMaior = mAtendimentos_Aguardando[vlCont]
vlEscolha = vlCont
}
Inc vlCont
}
aID_Cluster = vlEscolha

If mVerifica_Barco_TopUp[aID_Cluster,1] = 0 Or (mVerifica_Barco_TopUp[aID_Cluster,1]>0 And
mVerifica_Barco_TopUp[aID_Cluster,2] = 0) Then
{
Dec vPSVs_OD_Fundeio
Dec mAtendimentos_Aguardando[aID_Cluster]
If mVerifica_Barco_TopUp[aID_Cluster,1] > 0 Then aViagem_Complementar = 1
aID_TQ_Cluster = mID_TQ_Cluster[aID_Cluster]
If subMin(aID_TQ_PSV_OD, aID_TQ_Cluster) = Tank_FreeCap(aID_TQ_Cluster) Then
mVerifica_Barco_TopUp[aID_Cluster,2] = 1
Inc mVerifica_Barco_TopUp[aID_Cluster,1]
Get rPSV_OD
If clock() > vTempo_Aquecimento Then
{
mTempo_Ocioso_PSV[mEntrega_PSV[aID_PSV_OD],aID_PSV_OD] = clock() -
mTempo_Livra_PSV[aID_PSV_OD]
If mTempo_Ocioso_PSV[mEntrega_PSV[aID_PSV_OD], aID_PSV_OD] = 0 Then
mTempo_Ocioso_PSV[mEntrega_PSV[aID_PSV_OD], aID_PSV_OD] = 0.00001
Inc vTempo_Ocioso, mTempo_Ocioso_PSV[mEntrega_PSV[aID_PSV_OD],aID_PSV_OD]
Inc mTempo_Ocioso_PSV_Total[aID_PSV_OD],
mTempo_Ocioso_PSV[mEntrega_PSV[aID_PSV_OD],aID_PSV_OD]
}
Inc mEntrega_PSV[aID_PSV_OD]
mHora_Saida_FundeioNT[mEntrega_PSV[aID_PSV_OD], aID_PSV_OD] = clock()
mTempo_Saida_Atendimento[aID_Cluster] = clock()
Route aID_Cluster
}
Else {DISPLAY "ERRO" GOTO L2}
}
Else If vModelo_SobDemanda = 1 Then //Lógica modelo Sob-Demanda.
{
//Controle do AsIs.
aID_Local = mID_Local[vID_NT_Fundeio]
Int vlaID_local = aID_Local
If aRetorno = 1 Then
{
Inc mPSVs_fundeio[mPSV_Seleccionado[aID_PSV_OD]]
Inc mPSVs_Livres_Fundeio[mPSV_Seleccionado[aID_PSV_OD]]
}
}
}

```

```

aID_Local = mID_Local[vID_NT_Fundeio]
vIaID_local = aID_Local
Inc mStatus_Locais[vID_NT_Fundeio, vCol_Qtde_barcos_mStatus_Locais]
Inc mStatus_Locais[vID_NT_Fundeio, vCol_Qtde_barcos_livres_mStatus_Locais]
mStatus_PSVs[mPSV_Seleccionado[aID_PSV_OD], vCol_Local_mStatus_PSVs] = location()
mStatus_PSVs[mPSV_Seleccionado[aID_PSV_OD], vCol_Status_mStatus_PSVs] = vStatus_Ocioso
Inc mPSVs_OD_Local[vID_NT_Fundeio]
Inc mPSVS_OD_Livres[vID_NT_Fundeio]
mTempo_Livra_PSV[mPSV_Seleccionado[aID_PSV_OD]] = clock()
If clock() >= vTempo_Aquecimento and mTempo_Livra_PSV[mPSV_Seleccionado[aID_PSV_OD]]
> vTempo_Aquecimento and mFlag_Registro[mPSV_Seleccionado[aID_PSV_OD]] = 0 Then
{
    mTempo_Primeiro_Registro[mPSV_Seleccionado[aID_PSV_OD]] = clock()
    mFlag_Registro[mPSV_Seleccionado[aID_PSV_OD]] = 1
}
If clock() >= vTempo_Aquecimento and
mHora_Saida_FundeioNT[mEntrega_PSV[mPSV_Seleccionado[aID_PSV_OD]],
mPSV_Seleccionado[aID_PSV_OD]] >= vTempo_Aquecimento Then
{

mTempo_Ciclo[mEntrega_PSV[mPSV_Seleccionado[aID_PSV_OD]],mPSV_Seleccionado[aID_PSV_OD
]] = clock() -
mHora_Saida_FundeioNT[mEntrega_PSV[mPSV_Seleccionado[aID_PSV_OD]],mPSV_Seleccionado[aID
_PSV_OD]]
    Inc mSoma_Tempo_Ciclo[mPSV_Seleccionado[aID_PSV_OD]],
mTempo_Ciclo[mEntrega_PSV[mPSV_Seleccionado[aID_PSV_OD]],mPSV_Seleccionado[aID_PSV_OD
]]
    Inc mNumero_Visitas_FundeioNT[mPSV_Seleccionado[aID_PSV_OD]]
    Inc vCont_Ciclos
    Inc mTempo_Uso_PSV[mCont_Utilizacao[mPSV_seleccionado[aID_PSV_OD]],
mPSV_Seleccionado[aID_PSV_OD]], (clock() -
mTempo_Inicio_Uso[mPSV_Seleccionado[aID_PSV_OD]])
    Inc mTempo_Uso_PSV_Total[mPSV_Seleccionado[aID_PSV_OD]], (clock() -
mTempo_Inicio_Uso[mPSV_Seleccionado[aID_PSV_OD]])
}
mPSV_Seleccionado[aID_PSV_OD] =0
}
L1:
If vOtimizador = 0 Then
{
    Wait Until mPSVS_OD_Livres[vID_NT_Fundeio] > 0 And ((mPedidos_Aguardando[1] > 0 And
((mVerifica_Barco_TopUp[1,1] > 0 And mVerifica_Barco_TopUp[1,2] = 0) Or
mVerifica_Barco_TopUp[1,1] = 0) And mPSVs_OD_Livres[1] =0) Or
(mPedidos_Aguardando[2] > 0 And ((mVerifica_Barco_TopUp[2,1] > 0 And
mVerifica_Barco_TopUp[2,2] = 0) Or mVerifica_Barco_TopUp[2,1] = 0) And mPSVs_OD_Livres[2]
=0) Or
(mPedidos_Aguardando[3] > 0 And ((mVerifica_Barco_TopUp[3,1] > 0 And
mVerifica_Barco_TopUp[3,2] = 0) Or mVerifica_Barco_TopUp[3,1] = 0)And mPSVs_OD_Livres[3] =0)
Or
(mPedidos_Aguardando[4] > 0 And ((mVerifica_Barco_TopUp[4,1] > 0 And
mVerifica_Barco_TopUp[4,2] = 0) Or mVerifica_Barco_TopUp[4,1] = 0)And mPSVs_OD_Livres[4] =0)
Or
(mPedidos_Aguardando[5] > 0 And ((mVerifica_Barco_TopUp[5,1] > 0 And
mVerifica_Barco_TopUp[5,2] = 0) Or mVerifica_Barco_TopUp[5,1] = 0)And mPSVs_OD_Livres[5] =0)
Or
(mPedidos_Aguardando[6] > 0 And ((mVerifica_Barco_TopUp[6,1] > 0 And
mVerifica_Barco_TopUp[6,2] = 0) Or mVerifica_Barco_TopUp[6,1] = 0)And mPSVs_OD_Livres[6] =0)
Or
(mPedidos_Aguardando[7] > 0 And ((mVerifica_Barco_TopUp[7,1] > 0 And
mVerifica_Barco_TopUp[7,2] = 0) Or mVerifica_Barco_TopUp[7,1] = 0)And mPSVs_OD_Livres[7] =0)
}
}

```

```

Or
    (mPedidos_Aguardando[8] > 0 And ((mVerifica_Barco_TopUp[8,1] > 0 And
mVerifica_Barco_TopUp[8,2] = 0) Or mVerifica_Barco_TopUp[8,1] = 0)And mPSVs_OD_Livres[8] =0)
Or
    (mPedidos_Aguardando[9] > 0 And ((mVerifica_Barco_TopUp[9,1] > 0 And
mVerifica_Barco_TopUp[9,2] = 0) Or mVerifica_Barco_TopUp[9,1] = 0)And mPSVs_OD_Livres[9] =0)
Or
    (mPedidos_Aguardando[10] > 0 And ((mVerifica_Barco_TopUp[10,1] > 0 And
mVerifica_Barco_TopUp[10,2] = 0) Or mVerifica_Barco_TopUp[10,1] = 0)And mPSVs_OD_Livres[10]
=0) Or
    (mPedidos_Aguardando[11] > 0 And ((mVerifica_Barco_TopUp[11,1] > 0 And
mVerifica_Barco_TopUp[11,2] = 0) Or mVerifica_Barco_TopUp[11,1] = 0)And mPSVs_OD_Livres[11]
=0))
    vTem_Pedido = 0
    //Método de Seleção do Cluster
    sub_Selectiona_Cluster()
}
If vOtimizador = 1 Then
{
    Wait 0.001 sec
    Wait Until vTem_Programacao = 1
    //Wait 0.001 sec
    sub_Trata_Solucao_Otimizador()
    Wait 0.001 sec
}
aID_TQ_Cluster = mID_TQ_Cluster[aID_Cluster]

//Seleção do PSV_OD.
If vOtimizador = 0 Then
{
    sub_Selectiona_Barco()
}

If vOtimizador = 0 Then
{
    If mStatus_PSVs[mPSV_Seleccionado[aID_PSV_OD], vCol_Local_mStatus_PSVs] = location()
And mStatus_PSVs[mPSV_Seleccionado[aID_PSV_OD], vCol_Status_mStatus_PSVs] = vStatus_Ocioso
Then
    {
        Dec mPSVs_OD_Local[vID_NT_Fundeio]
        Dec mPSVs_OD_Livres[vID_NT_Fundeio]
        mStatus_PSVs[mPSV_Seleccionado[aID_PSV_OD], vCol_Status_mStatus_PSVs] =
vStatus_Ocupado
        Dec mStatus_Locais[vID_NT_Fundeio, vCol_Qtde_barcos_mStatus_Locais]
        Dec mStatus_Locais[vID_NT_Fundeio, vCol_Qtde_barcos_livres_mStatus_Locais]
        Inc vCont_utilizacao
        Inc mCont_Utilizacao[mPSV_seleccionado[aID_PSV_OD]]
        If Tank_Level[mID_TQ_PSV_OD [mPSV_Seleccionado[aID_PSV_OD]]] >=
mPedidos_Aguardando[aID_Cluster] Then
            {
                aVolume_a_entregar = mPedidos_Aguardando[aID_Cluster]
                Dec mPedidos_Aguardando[aID_Cluster], mPedidos_Aguardando[aID_Cluster]
            }
            Else
            {
                aVolume_a_entregar = Tank_Level[mID_TQ_PSV_OD [mPSV_Seleccionado[aID_PSV_OD]]]
                Dec mPedidos_Aguardando[aID_Cluster], Tank_Level[mID_TQ_PSV_OD
[mPSV_Seleccionado[aID_PSV_OD]]]
            }
            Inc mEstoque_Transito[aID_Cluster], aVolume_a_entregar
    }
}

```

```

If mPedidos_Aguardando[aID_Cluster] = 0 Then
{
  aUltima_Fracao = 1
  mStatus_Pedidos[mPedidos_Gerados[aID_Cluster], vCol_Ultima_Fracao_mStatus_Pedidos,
aID_Cluster] = 1
}
Else
{
  Inc vTem_Demanda_Restante
  Inc mStatus_Pedidos[mPedidos_Gerados[aID_Cluster], vCol_Entrega_Parcial, aID_Cluster]
  aUltima_Fracao =0
}
If mPedidos_Gerados[aID_Cluster] <> 0 Then
{
  mStatus_Pedidos[mPedidos_Gerados[aID_Cluster], vCol_Status_Pedido_mStatus_Pedidos,
aID_Cluster] = vPedido_Em_Transito
}
Inc mEntrega_PSV[mPSV_Selecionado[aID_PSV_OD]]
aOrigem_Local = Location()
If mVerifica_Barco_TopUp[aID_Cluster,1] > 0 Then aViagem_Complementar = 1
If SubMinimo_entre_tres(mID_TQ_PSV_OD[mPSV_Selecionado[aID_PSV_OD]],
aVolume_a_entregar, aID_TQ_Cluster) = Tank_FreeCap(aID_TQ_Cluster) Then
mVerifica_Barco_TopUp[aID_Cluster,2] = 1
Inc mVerifica_Barco_TopUp[aID_Cluster,1]
Get rPSV_OD
If clock() >= vTempo_Aquecimento and
mTempo_Livra_PSV[mPSV_Selecionado[aID_PSV_OD]] >= vTempo_Aquecimento Then
{
  If mFlag_Registro[mPSV_Selecionado[aID_PSV_OD]] = 0 Then
  {
    mTempo_Primeiro_Registro[mPSV_Selecionado[aID_PSV_OD]] = clock()
    mFlag_Registro[mPSV_Selecionado[aID_PSV_OD]] = 1
  }
  mTempo_Ocioso_PSV[vCont_Utilizacao, mPSV_Selecionado[aID_PSV_OD]] = clock() -
mTempo_Livra_PSV[mPSV_Selecionado[aID_PSV_OD]]
  If mTempo_Ocioso_PSV[vCont_Utilizacao, mPSV_Selecionado[aID_PSV_OD]] = 0 Then
mTempo_Ocioso_PSV[vCont_Utilizacao, mPSV_Selecionado[aID_PSV_OD]] = 0.0001
  Inc vTempo_Ocioso,
mTempo_Ocioso_PSV[vCont_Utilizacao, mPSV_Selecionado[aID_PSV_OD]]
  Inc mTempo_Ocioso_PSV_Total[mPSV_Selecionado[aID_PSV_OD]],
mTempo_Ocioso_PSV[vCont_Utilizacao, mPSV_Selecionado[aID_PSV_OD]]
}
  mHora_Saida_FundeioNT[mEntrega_PSV[mPSV_Selecionado[aID_PSV_OD]],
mPSV_Selecionado[aID_PSV_OD]] = clock()
  mTempo_Inicio_Uso[mPSV_Selecionado[aID_PSV_OD]] = clock()
  Route aID_Cluster
}
Else { Wait 0.001 sec GOTO L1 }
}
If vOtimizador = 1 Then
{
  If mPSV_Selecionado[aID_PSV_OD] <> 0 Then
  {
    If aCondicao_Programacao = 1 and mStatus_PSVs[mPSV_Selecionado[aID_PSV_OD],
vCol_Local_mStatus_PSVs] = location() And mStatus_PSVs[mPSV_Selecionado[aID_PSV_OD],
vCol_Status_mStatus_PSVs] = vStatus_Ocioso Then
    {
      Dec mPSVs_OD_Local[vID_NT_Fundeio]
      Dec mPSVs_OD_Livres[vID_NT_Fundeio]
      mStatus_PSVs[mPSV_Selecionado[aID_PSV_OD], vCol_Status_mStatus_PSVs] =

```

```

vStatus_Ocupado
  Dec mStatus_Locais[vID_NT_Fundeio, vCol_Qtde_barcos_mStatus_Locais]
  Dec mStatus_Locais[vID_NT_Fundeio, vCol_Qtde_barcos_livres_mStatus_Locais]
  Inc vCont_utilizacao
  Inc mCont_Utilizacao[mPSV_selecionado[aID_PSV_OD]]
  If Tank_Level[mID_TQ_PSV_OD [mPSV_Selecionado[aID_PSV_OD]]] >=
mPedidos_Aguardando[aID_Cluster] Then
  {
    aVolume_a_entregar = mPedidos_Aguardando[aID_Cluster]
    Dec mPedidos_Aguardando[aID_Cluster], mPedidos_Aguardando[aID_Cluster]
  }
  Else
  {
    aVolume_a_entregar = Tank_Level[mID_TQ_PSV_OD
[mPSV_Selecionado[aID_PSV_OD]]]
    Dec mPedidos_Aguardando[aID_Cluster], Tank_Level[mID_TQ_PSV_OD
[mPSV_Selecionado[aID_PSV_OD]]]
  }
  Inc mEstoque_Transito[aID_Cluster], aVolume_a_entregar
  If mPedidos_Aguardando[aID_Cluster] = 0 Then
  {
    aUltima_Fracao = 1
    mStatus_Pedidos[mPedidos_Gerados[aID_Cluster], vCol_Ultima_Fracao_mStatus_Pedidos,
aID_Cluster] = 1
  }
  Else
  {
    Inc vTem_Demanda_Restante
    Inc mStatus_Pedidos[mPedidos_Gerados[aID_Cluster], vCol_Entrega_Parcial, aID_Cluster]
    aUltima_Fracao = 0
  }
  If mPedidos_Gerados[aID_Cluster] <> 0 Then
  {
    mStatus_Pedidos[mPedidos_Gerados[aID_Cluster], vCol_Status_Pedido_mStatus_Pedidos,
aID_Cluster] = vPedido_Em_Transito
  }
  Inc mEntrega_PSV[mPSV_Selecionado[aID_PSV_OD]]
  aOrigem_Local = Location()
  If mVerifica_Barco_TopUp[aID_Cluster,1] > 0 Then aViagem_Complementar = 1
  If SubMinimo_entre_tres(mID_TQ_PSV_OD[mPSV_Selecionado[aID_PSV_OD]],
aVolume_a_entregar, aID_TQ_Cluster) = Tank_FreeCap(aID_TQ_Cluster) Then
mVerifica_Barco_TopUp[aID_Cluster,2] = 1
  Inc mVerifica_Barco_TopUp[aID_Cluster,1]
  Get rPSV_OD
  If clock() >= vTempo_Aquecimento and
mTempo_Livra_PSV[mPSV_Selecionado[aID_PSV_OD]] >= vTempo_Aquecimento Then
  {
    If mFlag_Registro[mPSV_Selecionado[aID_PSV_OD]] = 0 Then
    {
      mTempo_Primeiro_Registro[mPSV_Selecionado[aID_PSV_OD]] = clock()
      mFlag_Registro[mPSV_Selecionado[aID_PSV_OD]] = 1
    }
    mTempo_Ocioso_PSV[vCont_Utilizacao, mPSV_Selecionado[aID_PSV_OD]] = clock() -
mTempo_Livra_PSV[mPSV_Selecionado[aID_PSV_OD]]
    If mTempo_Ocioso_PSV[vCont_Utilizacao, mPSV_Selecionado[aID_PSV_OD]] = 0 Then
mTempo_Ocioso_PSV[vCont_Utilizacao, mPSV_Selecionado[aID_PSV_OD]] = 0.0001
    Inc vTempo_Ocioso,
mTempo_Ocioso_PSV[vCont_Utilizacao, mPSV_Selecionado[aID_PSV_OD]]
    Inc mTempo_Ocioso_PSV_Total[mPSV_Selecionado[aID_PSV_OD]],
mTempo_Ocioso_PSV[vCont_Utilizacao, mPSV_Selecionado[aID_PSV_OD]]
  }

```

```

    }
    mHora_Saida_FundeioNT[mEntrega_PSV[mPSV_Selecionado[aID_PSV_OD]],
mPSV_Selecionado[aID_PSV_OD]] = clock()
    mTempo_Inicio_Uso[mPSV_Selecionado[aID_PSV_OD]] = clock()
    aCondicao_Programacao =0
    Route aID_Cluster
    }
    Else { wait 0.001 sec GOTO L1}
}
Else { Wait 0.001 sec GOTO L1}
}
}
}

```

Processo NT

```

If vModelo_SobDemanda = 0 Then
{
    Tank_Transfer(TQ_NT, aID_TQ_PSV_OD, Tank_FreeCap(aID_TQ_PSV_OD), 60, 0, 0)
    Tank_SetLevel(TQ_NT, Tank_Cap(TQ_NT))
    Free All
    Inc vPSVs_OD_Fundeio
    aRetorno= 1
}
If vModelo_SobDemanda = 1 Then
{
    Tank_Transfer(TQ_NT, mID_TQ_PSV_OD[mPSV_Selecionado[aID_PSV_OD]],
Tank_FreeCap(mID_TQ_PSV_OD[mPSV_Selecionado[aID_PSV_OD]]), 60, 0, 0)
    Tank_SetLevel(TQ_NT, Tank_Cap(TQ_NT))
    Free All
    Inc vPSVs_OD_Fundeio
    Inc vPSVs_OD_Livres_Fundeio
    aRetorno = 1
}
}

```

Processo Clusters

```

If vModelo_SobDemanda = 0 Then //Entra aqui para o modelo de entregas programadas.
{
    macProcesso_Geral_Entregas_Programadas
}
Else If vModelo_SobDemanda = 1 Then //Entra aqui para o modelo de entregas Sob Demanda.
{
    macProcesso_Geral_Sob_Demanda
}
}

```

Macro macProcesso_Geral_Entregas_Sob_Demanda

```

L3:
Int vlaID_Cluster = aID_Cluster
aDestino = 1
Inc mIndice_Visita_Cluster[aID_Cluster]
If clock() >= vTempo_Aquecimento Then
{
    mIntervalo_entre_Visitas[mIndice_Visita_Cluster[aID_Cluster], aID_Cluster] = clock() -
mUltima_Visita_Cluster[aID_Cluster]
}
}

```

```

    Inc mSoma_Intervalo_entre_Visitas[aID_Cluster],
    mIntervalo_entre_Visitas[mIndice_Visita_Cluster[aID_Cluster], aID_Cluster]
    Inc mIndice_Visita_Cluster_aquecimento[aID_Cluster]
}
mUltima_Visita_Cluster[aID_Cluster] = clock()
Inc mPSVs_OD_Local[aID_Cluster]
mStatus_PSVs[mPSV_Seleccionado[aID_PSV_OD], vCol_Local_mStatus_PSVs] = location()
Inc mStatus_Locais[aID_Cluster, vCol_Qtde_barcos_mStatus_Locais]
If SubMinimo_entre_tres(mID_TQ_PSV_OD[mPSV_Seleccionado[aID_PSV_OD]],
aVolume_a_entregar, aID_TQ_Cluster) = Tank_FreeCap(aID_TQ_Cluster) Or
aVolume_Faltante_Topar_TQ_Cluster < aVolume_a_entregar Then
mVerifica_Barco_TopUp[aID_Cluster,2] = 1
If aViagem_Complementar = 1 Or mVerifica_Barco_TopUp[aID_Cluster, 1] > 1 Then
{
    vITQ_Volume_Para_Transferir = SubMinimo(aVolume_a_entregar,
aVolume_Faltante_Topar_TQ_Cluster)
    aVolume_Faltante_Topar_TQ_Cluster =
SubMinimo(SubPiso_Zero(aVolume_Faltante_Topar_TQ_Cluster -
SubMinimo_entre_tres(mID_TQ_PSV_OD[mPSV_Seleccionado[aID_PSV_OD]], aVolume_a_entregar,
aID_TQ_Cluster)), aVolume_Faltante_Topar_TQ_Cluster)
}
Else
{
    vITQ_Volume_Para_Transferir =
SubMinimo_entre_tres(mID_TQ_PSV_OD[mPSV_Seleccionado[aID_PSV_OD]],aVolume_a_entregar,
aID_TQ_Cluster)
    aVolume_Faltante_Topar_TQ_Cluster = Tank_FreeCap(aID_TQ_Cluster) -
SubMinimo_entre_tres(mID_TQ_PSV_OD[mPSV_Seleccionado[aID_PSV_OD]], aVolume_a_entregar,
aID_TQ_Cluster)
}
vIQtde_de_fato_transferida =
SubMinimo_entre_tres(mID_TQ_PSV_OD[mPSV_Seleccionado[aID_PSV_OD]],vITQ_Volume_Para_Tr
ansferir, aID_TQ_Cluster)

Tank_Transfer(mID_TQ_PSV_OD[mPSV_Seleccionado[aID_PSV_OD]],aID_TQ_Cluster,
vIQtde_de_fato_transferida, 25, 0,0)
If mStatus_Pedidos[mPedidos_Gerados[aID_Cluster], vCol_Entrega_Parcial, aID_Cluster] > 0 Then
{
    Dec mStatus_Pedidos[mPedidos_Gerados[aID_Cluster], vCol_Entrega_Parcial, aID_Cluster]
}
Dec mEstoque_Transito[aID_Cluster], aVolume_a_entregar
Inc ventrega
mStatus_PSVs[mPSV_Seleccionado[aID_PSV_OD], vCol_Status_mStatus_PSVs] = vStatus_Ocioso
If aUltima_Fracao =1 Then
{
    mStatus_Pedidos[mPedidos_Gerados[aID_Cluster], vCol_Status_Pedido_mStatus_Pedidos,
aID_Cluster] = vPedido_Entregue
}
If clock() >= vTempo_Aquecimento and mStatus_Pedidos[mPedidos_Gerados[aID_Cluster],
vCol_Horario_Pedido_mStatus_Pedidos, aID_Cluster] >= vTempo_Aquecimento and
mPedidos_Gerados_pos_aquecimento[aID_Cluster] > 0 Then
{
    Inc mVolume_Total_Entregue_Sob_Demanda[aID_Cluster], vIQtde_de_fato_transferida
    mLead_Time_pos_Aquecimento[mPedidos_Gerados_pos_aquecimento[aID_Cluster], aID_Cluster] =
clock() - mhora_ultimo_pedido[aID_Cluster]
    If aUltima_Fracao =1 Then
    {
        Inc mCont_Entregas_Sob_Demanda [aID_Cluster]
        aUltima_Fracao = 0
    }
}

```



```

}
mStatus_Pedidos[mPedidos_Gerados[aID_Cluster], vCol_Horario_Entrega_mStatus_Pedidos,
aID_Cluster] = clock()
mStatus_Pedidos[mPedidos_Gerados[aID_Cluster], vCol_Lead_Time_mStatus_Pedidos, aID_Cluster] =
clock() - mhora_ultimo_pedido[aID_Cluster]
mLead_Time[mPedidos_Gerados[aID_Cluster], aID_Cluster] = clock() -
mhora_ultimo_pedido[aID_Cluster]
Inc mSoma_Tempo_Lead_Time[aID_Cluster], mLead_Time[mPedidos_Gerados[aID_Cluster],
aID_Cluster]
mLead_Time_Esperado[aID_Cluster] = mSoma_Tempo_Lead_Time[aID_Cluster]/
mPedidos_Gerados[aID_Cluster]
Inc mPSVS_OD_Livres[aID_Cluster]
Inc mStatus_Locais[aID_Cluster, vCol_Qtde_barcos_livres_mStatus_Locais]
mTempo_Livra_PSV[mPSV_Seleccionado[aID_PSV_OD]] = clock()
If clock() >= vTempo_Aquecimento and mTempo_Inicio_Uso[mPSV_Seleccionado[aID_PSV_OD]] >=
vTempo_Aquecimento Then
{
  If mFlag_Registro[mPSV_Seleccionado[aID_PSV_OD]] = 0 Then
  {
    mTempo_Primeiro_Registro[mPSV_Seleccionado[aID_PSV_OD]] =clock()
    mFlag_Registro[mPSV_Seleccionado[aID_PSV_OD]] = 1
  }
  mTempo_Uso_PSV[mCont_Utilizacao[mPSV_seleccionado[aID_PSV_OD]],
mPSV_Seleccionado[aID_PSV_OD]] = clock() -
mTempo_Inicio_Uso[mPSV_Seleccionado[aID_PSV_OD]]
  Inc mTempo_Uso_PSV_Total[mPSV_Seleccionado[aID_PSV_OD]],
mTempo_Uso_PSV[mCont_Utilizacao[mPSV_seleccionado[aID_PSV_OD]],
mPSV_Seleccionado[aID_PSV_OD]]
}
aViagem_Complementar = 0
Dec mVerifica_Barco_TopUp[aID_Cluster,1]
mVerifica_Barco_TopUp[aID_Cluster,2] = 0
//Esperar até surgir novo pedido e o tanque atingir nível mínimo.
L1:
While Tank_Level[mID_TQ_PSV_OD[mPSV_Seleccionado[aID_PSV_OD]]] >
mEstoques_PSVs_OD_IN[mPSV_Seleccionado[aID_PSV_OD], 1] Do
{
  aDestino = 1
  If vOtimizador = 0 Then
  {
    Wait Until mPSVS_OD_Livres[aID_Cluster] >0 And ((mPedidos_Aguardando[1] > 0 And
((mVerifica_Barco_TopUp[1,1] > 0 And mVerifica_Barco_TopUp[1,2] = 0) Or
mVerifica_Barco_TopUp[1,1] = 0) ) Or
(mPedidos_Aguardando[2] > 0 And ((mVerifica_Barco_TopUp[2,1] > 0 And
mVerifica_Barco_TopUp[2,2] = 0) Or mVerifica_Barco_TopUp[2,1] = 0) ) Or
(mPedidos_Aguardando[3] > 0 And ((mVerifica_Barco_TopUp[3,1] > 0 And
mVerifica_Barco_TopUp[3,2] = 0) Or mVerifica_Barco_TopUp[3,1] = 0) ) Or
(mPedidos_Aguardando[4] > 0 And ((mVerifica_Barco_TopUp[4,1] > 0 And
mVerifica_Barco_TopUp[4,2] = 0) Or mVerifica_Barco_TopUp[4,1] = 0) ) Or
(mPedidos_Aguardando[5] > 0 And ((mVerifica_Barco_TopUp[5,1] > 0 And
mVerifica_Barco_TopUp[5,2] = 0) Or mVerifica_Barco_TopUp[5,1] = 0) ) Or
(mPedidos_Aguardando[6] > 0 And ((mVerifica_Barco_TopUp[6,1] > 0 And
mVerifica_Barco_TopUp[6,2] = 0) Or mVerifica_Barco_TopUp[6,1] = 0) ) Or
(mPedidos_Aguardando[7] > 0 And ((mVerifica_Barco_TopUp[7,1] > 0 And
mVerifica_Barco_TopUp[7,2] = 0) Or mVerifica_Barco_TopUp[7,1] = 0) ) Or
(mPedidos_Aguardando[8] > 0 And ((mVerifica_Barco_TopUp[8,1] > 0 And
mVerifica_Barco_TopUp[8,2] = 0) Or mVerifica_Barco_TopUp[8,1] = 0) ) Or
(mPedidos_Aguardando[9] > 0 And ((mVerifica_Barco_TopUp[9,1] > 0 And
mVerifica_Barco_TopUp[9,2] = 0) Or mVerifica_Barco_TopUp[9,1] = 0) ) Or
(mPedidos_Aguardando[10] > 0 And ((mVerifica_Barco_TopUp[10,1] > 0 And

```

```

mVerifica_Barco_TopUp[10,2] = 0) Or mVerifica_Barco_TopUp[10,1] = 0) ) Or
      (mPedidos_Aguardando[11] > 0 And ((mVerifica_Barco_TopUp[11,1] > 0 And
mVerifica_Barco_TopUp[11,2] = 0) Or mVerifica_Barco_TopUp[11,1] = 0) ))
  vTem_Pedido = 0
  //Método de seleção do Cluster.
  sub_Seleciona_Cluster()
}
If vOtimizador = 1 Then
{
  Wait 0.001 sec
  Wait Until vTem_Programacao = 1
  Wait 0.001 sec
  sub_Trata_Solucao_Otimizador()
  Wait 0.001 sec
}
aID_TQ_Cluster = mID_TQ_Cluster[aID_Cluster]
If vOtimizador = 0 Then
{
  //Selecionar PSV
  sub_Seleciona_Barco()
}
//Verifica se o barco escolhido está no local atual e está livre.
If vOtimizador = 0 Then
{
  If mStatus_PSVs[mPSV_Selecionado[aID_PSV_OD], vCol_Local_mStatus_PSVs] = location()
And mStatus_PSVs[mPSV_Selecionado[aID_PSV_OD],vCol_Status_mStatus_PSVs] = vStatus_Ocioso
Then
  {
    mStatus_PSVs[mPSV_Selecionado[aID_PSV_OD], vCol_Status_mStatus_PSVs] =
vStatus_Ocupado
    Dec mPSVS_OD_Livres[vlaID_Cluster]
    Dec mPSVs_OD_Local[vlaID_Cluster]
    Dec mStatus_Locais[vlaID_Cluster, vCol_Qtde_barcos_mStatus_Locais]
    Dec mStatus_Locais[vlaID_Cluster, vCol_Qtde_barcos_livres_mStatus_Locais]
    Inc vCont_utilizacao
    Inc mCont_Utilizacao[mPSV_selecionado[aID_PSV_OD]]
    mTempo_Inicio_Uso[mPSV_Selecionado[aID_PSV_OD]] = clock()
    If clock() >= vTempo_Aquecimento and
mTempo_Livra_PSV[mPSV_Selecionado[aID_PSV_OD]]>= vTempo_Aquecimento Then
    {
      If mFlag_Registro[mPSV_Selecionado[aID_PSV_OD]] = 0 Then
      {
        mTempo_Primeiro_Registro[mPSV_Selecionado[aID_PSV_OD]] =clock()
        mFlag_Registro[mPSV_Selecionado[aID_PSV_OD]] = 1
      }
      mTempo_Ocioso_PSV[vCont_Utilizacao,mPSV_Selecionado[aID_PSV_OD]] = clock() -
mTempo_Livra_PSV[mPSV_Selecionado[aID_PSV_OD]]
      If mTempo_Ocioso_PSV[vCont_Utilizacao,mPSV_Selecionado[aID_PSV_OD]] = 0 Then
mTempo_Ocioso_PSV[vCont_Utilizacao,mPSV_Selecionado[aID_PSV_OD]] = 0.00001
      Inc vTempo_Ocioso,
mTempo_Ocioso_PSV[vCont_Utilizacao,mPSV_Selecionado[aID_PSV_OD]]
      Inc mTempo_Ocioso_PSV_Total[mPSV_Selecionado[aID_PSV_OD]],
mTempo_Ocioso_PSV[vCont_Utilizacao,mPSV_Selecionado[aID_PSV_OD]]
    }
    If Tank_Level[mID_TQ_PSV_OD [mPSV_Selecionado[aID_PSV_OD]]] >=
mPedidos_Aguardando[aID_Cluster] Then
    {
      aVolume_a_entregar = mPedidos_Aguardando[aID_Cluster]
      Dec mPedidos_Aguardando[aID_Cluster], mPedidos_Aguardando[aID_Cluster]
    }
  }
}
}

```

```

Else
{
  aVolume_a_entregar = Tank_Level[mID_TQ_PSV_OD [mPSV_Selecionado[aID_PSV_OD]]]
  Dec mPedidos_Aguardando[aID_Cluster], Tank_Level[mID_TQ_PSV_OD
[mPSV_Selecionado[aID_PSV_OD]]]
}
Inc mEstoque_Transito[aID_Cluster], aVolume_a_entregar
If mPedidos_Aguardando[aID_Cluster] = 0 Then
{
  aUltima_Fracao = 1
  mStatus_Pedidos[mPedidos_Gerados[aID_Cluster], vCol_Ultima_Fracao_mStatus_Pedidos,
aID_Cluster] = 1
}
Else
{
  Inc vTem_Demanda_Restante
  Inc mStatus_Pedidos[mPedidos_Gerados[aID_Cluster], vCol_Entrega_Parcial, aID_Cluster]
  aUltima_Fracao = 0
}
If mPedidos_Gerados[aID_Cluster] <> 0 Then
{
  mStatus_Pedidos[mPedidos_Gerados[aID_Cluster], vCol_Status_Pedido_mStatus_Pedidos,
aID_Cluster] = vPedido_Em_Transito
}
If mVerifica_Barco_TopUp[aID_Cluster,1] > 0 Then aViagem_Complementar = 1
If SubMinimo_entre_tres(mID_TQ_PSV_OD[mPSV_Selecionado[aID_PSV_OD]],
aVolume_a_entregar, aID_TQ_Cluster) = Tank_FreeCap(aID_TQ_Cluster) Then
mVerifica_Barco_TopUp[aID_Cluster,2] = 1
Inc mVerifica_Barco_TopUp[aID_Cluster,1]
//Calcular tempo de navegação entre UMs.
If aID_Local = location() Then
{
  Wait T(20,30,40) min
  GOTO L3
}
Else
{
  aDestino = aID_Cluster + 1
  Break
}
}
Else GOTO L1
}
If vOtimizador = 1 Then
{
  If aCondicao_Programacao = 1 and mStatus_PSVs[mPSV_Selecionado[aID_PSV_OD],
vCol_Local_mStatus_PSVs] = location() And
mStatus_PSVs[mPSV_Selecionado[aID_PSV_OD],vCol_Status_mStatus_PSVs] = vStatus_Ocioso Then
{
  mStatus_PSVs[mPSV_Selecionado[aID_PSV_OD], vCol_Status_mStatus_PSVs] =
vStatus_Ocupado
  Dec mPSVS_OD_Livres[vlaID_Cluster]
  Dec mPSVs_OD_Local[vlaID_Cluster]
  Dec mStatus_Locais[vlaID_Cluster, vCol_Qtde_barcos_mStatus_Locais]
  Dec mStatus_Locais[vlaID_Cluster, vCol_Qtde_barcos_livres_mStatus_Locais]
  Inc vCont_utilizacao
  Inc mCont_Utilizacao[mPSV_selecionado[aID_PSV_OD]]
  mTempo_Inicio_Uso[mPSV_Selecionado[aID_PSV_OD]] = clock()
  If clock() >= vTempo_Aquecimento and

```

```

mTempo_Livra_PSV[mPSV_Seleccionado[aID_PSV_OD]]>= vTempo_Aquecimento Then
{
  If mFlag_Registro[mPSV_Seleccionado[aID_PSV_OD]] = 0 Then
  {
    mTempo_Primeiro_Registro[mPSV_Seleccionado[aID_PSV_OD]] =clock()
    mFlag_Registro[mPSV_Seleccionado[aID_PSV_OD]] = 1
  }
  mTempo_Ocioso_PSV[vCont_Utilizacao,mPSV_Seleccionado[aID_PSV_OD]] = clock() -
mTempo_Livra_PSV[mPSV_Seleccionado[aID_PSV_OD]]
  If mTempo_Ocioso_PSV[vCont_Utilizacao,mPSV_Seleccionado[aID_PSV_OD]] = 0 Then
mTempo_Ocioso_PSV[vCont_Utilizacao,mPSV_Seleccionado[aID_PSV_OD]] = 0.00001
  Inc vTempo_Ocioso,
mTempo_Ocioso_PSV[vCont_Utilizacao,mPSV_Seleccionado[aID_PSV_OD]]
  Inc mTempo_Ocioso_PSV_Total[mPSV_Seleccionado[aID_PSV_OD]],
mTempo_Ocioso_PSV[vCont_Utilizacao,mPSV_Seleccionado[aID_PSV_OD]]
}
  If Tank_Level[mID_TQ_PSV_OD [mPSV_Seleccionado[aID_PSV_OD]]] >=
mPedidos_Aguardando[aID_Cluster] Then
  {
    aVolume_a_entregar = mPedidos_Aguardando[aID_Cluster]
    Dec mPedidos_Aguardando[aID_Cluster], mPedidos_Aguardando[aID_Cluster]
  }
  Else
  {
    aVolume_a_entregar = Tank_Level[mID_TQ_PSV_OD [mPSV_Seleccionado[aID_PSV_OD]]]
    Dec mPedidos_Aguardando[aID_Cluster], Tank_Level[mID_TQ_PSV_OD
[mPSV_Seleccionado[aID_PSV_OD]]]
  }
  Inc mEstoque_Transito[aID_Cluster], aVolume_a_entregar
  If mPedidos_Aguardando[aID_Cluster] = 0 Then
  {
    aUltima_Fracao = 1
    mStatus_Pedidos[mPedidos_Gerados[aID_Cluster], vCol_Ultima_Fracao_mStatus_Pedidos,
aID_Cluster] = 1
  }
  Else
  {
    Inc vTem_Demanda_Restante
    Inc mStatus_Pedidos[mPedidos_Gerados[aID_Cluster], vCol_Entrega_Parcial, aID_Cluster]
    aUltima_Fracao =0
  }
  If mPedidos_Gerados[aID_Cluster] <> 0 Then
  {
    mStatus_Pedidos[mPedidos_Gerados[aID_Cluster], vCol_Status_Pedido_mStatus_Pedidos,
aID_Cluster] = vPedido_Em_Transito
  }
  If mVerifica_Barco_TopUp[aID_Cluster,1] > 0 Then aViagem_Complementar = 1
  If SubMinimo_entre_tres(mID_TQ_PSV_OD[mPSV_Seleccionado[aID_PSV_OD]],
aVolume_a_entregar, aID_TQ_Cluster) = Tank_FreeCap(aID_TQ_Cluster) Then
mVerifica_Barco_TopUp[aID_Cluster,2] = 1
  Inc mVerifica_Barco_TopUp[aID_Cluster,1]
  //Calcular tempo de navegaco entre UMs.
  aCondicao_Programacao = 0
  If aID_Local = location() Then
  {
    Wait T(20,30,40) min
    GOTO L3
  }
  Else
  {

```

```

        aDestino = aID_Cluster + 1
        Break
    }
}
Else { Wait 0.001 sec GOTO L1 }
}
}
}
aOrigem_Local = Location()
If aDestino = 1 Then
{
Dec mStatus_Locais[vlaID_Cluster, vCol_Qtde_barcos_livres_mStatus_Locais]
Dec mStatus_Locais[vlaID_Cluster, vCol_Qtde_barcos_mStatus_Locais]
Dec mPSVS_OD_Livres[vlaID_Cluster]
Dec mPSVs_OD_Local[vlaID_Cluster]
mStatus_PSVs[mPSV_Seleccionado[aID_PSV_OD], vCol_Status_mStatus_PSVs] = vStatus_Ocupado
mTempo_Inicio_Uso[mPSV_Seleccionado[aID_PSV_OD]] = clock()
}
Route aDestino

```

Macro macProcesso_Geral_Entregas_Programadas

```

Inc mIndice_Visita_Cluster[aID_Cluster]
If clock() >= vTempo_Aquecimento Then
{
    mIntervalo_entre_Visitas[mIndice_Visita_Cluster[aID_Cluster], aID_Cluster] = clock() -
    mUltima_Visita_Cluster[aID_Cluster]
    Inc
    mSoma_Intervalo_entre_Visitas[aID_Cluster], mIntervalo_entre_Visitas[mIndice_Visita_Cluster[aID_Cluster], aID_Cluster]
    Inc mIndice_Visita_Cluster_aquecimento[aID_Cluster]
    Inc mIndice_Visita_Cluster_pos_aquecimento[aID_Cluster]
}
mUltima_Visita_Cluster[aID_Cluster] = clock()
If subMin(aID_TQ_PSV_OD, aID_TQ_Cluster) = Tank_FreeCap(aID_TQ_Cluster) Or
aVolume_Faltante_Topar_TQ_Cluster < subMin(aID_TQ_PSV_OD, aID_TQ_Cluster) Then
mVerifica_Barco_TopUp[aID_Cluster, 2] = 1
Else aVolume_Faltante_Topar_TQ_Cluster = Tank_FreeCap(aID_TQ_Cluster) -
subMin(aID_TQ_PSV_OD, aID_TQ_Cluster)
Real vITQ_Volume_Para_Transferir
If aViagem_Complementar = 1 Or mVerifica_Barco_TopUp[aID_Cluster, 1] > 1 Then
vITQ_Volume_Para_Transferir = aVolume_Faltante_Topar_TQ_Cluster
Else vITQ_Volume_Para_Transferir = subMin(aID_TQ_PSV_OD, aID_TQ_Cluster)
Real vIQtde_de_fato_Transferida = subMinimo_entre_tres(aID_TQ_PSV_OD,
vITQ_Volume_Para_Transferir, aID_TQ_Cluster)
Tank_Transfer(aID_TQ_PSV_OD, aID_TQ_Cluster, vIQtde_de_fato_Transferida, 25, 0, 0)
mLead_Time[mIndice_Visita_Cluster[aID_Cluster], aID_Cluster] = clock() -
mTempo_Saida_Atendimento[aID_Cluster]
If clock() >= vTempo_Aquecimento and mTempo_Saida_Atendimento[aID_Cluster] >=
vTempo_Aquecimento Then
{
    Inc mVolume_Entregue_Entregas_Programadas[aID_Cluster], vIQtde_de_fato_Transferida
    Inc mCont_Entregas_Entregas_Programadas [aID_Cluster]
    Inc ventrega
    mLead_Time_pos_Aquecimento[mIndice_Visita_Cluster_pos_aquecimento[aID_Cluster],
aID_Cluster] = clock() - mTempo_Saida_Atendimento[aID_Cluster]
}
}
Route 1

```

Sub Selecciona_Cluster

```
int i = 1
Real vlPedido_mais_Antigo = Clock(hr) + 1
While i <= ArrayDimSize(mPedidos_Aguardando,1) Do
{
    If mPedidos_Aguardando[i] > 0 And ((mVerifica_Barco_TopUp[i, 1] > 0 And
mVerifica_Barco_TopUp[i,2] = 0) Or (mVerifica_Barco_TopUp[i,1]= 0)) And mhora_ultimo_pedido[i]
< vlPedido_mais_Antigo Then
    {
        vlPedido_mais_Antigo = mhora_ultimo_pedido[i]
        aID_Cluster = i
        aID_Local = mID_Local[i]
    }
    Inc i
}
}
```

Sub Selecciona_Barco

```
//Verifica quais barcos livres do local são capazes.
int vlaID_Local = aID_Cluster
If location() = vloc_Fundeio_NT Then vlaID_Local = vID_NT_Fundeio
int z = 1
Int cont = 0
While z <= ArrayDimSize(mStatus_PSVs,1) Do
{
    If mStatus_PSVs[z, vCol_Local_mStatus_PSVs] = location() And mStatus_PSVs[z,
vCol_Status_mStatus_PSVs] = vStatus_Ocioso Then
    {
        If Tank_Level[mID_TQ_PSV_OD[z]] >= mPedidos_Aguardando[aID_Cluster] Then
        {
            mPSVs_Capazes[z, vlaID_Local] = 1
            cont= cont+1
        }
    }
    Inc z
}
//Caso nenhum barco seja capaz de carregar o pedido, escolher o barco livre que tiver maior volume em
tanque.
If cont = 0 Then
{
    Real vlmax = 0
    Int Maior = 1
    While Maior <= ArrayDimSize(mStatus_PSVs,1) Do
    {
        If mStatus_PSVs[Maior, vCol_Local_mStatus_PSVs] = location() And mStatus_PSVs[Maior,
vCol_Status_mStatus_PSVs] = vStatus_Ocioso and Tank_Level[mID_TQ_PSV_OD [Maior]] > vlmax
Then
        {
            vlmax = Tank_Level[mID_TQ_PSV_OD [Maior]]
            mPSV_Seleccionado[aID_PSV_OD] = Maior
        }
    }
}
```

```

    Inc Maior
  }
}
//Caso haja barco capaz, seleciona o barco de menor volume em tanque dentre todos os barcos capazes do
local.
Else
{
  Int k = 1
  Real vlmin = 3000
  While k <= ArrayDimSize(mStatus_PSVs,1) Do
  {
    If mPSVs_Capazes[k, vlaID_Local] > 0 Then
    {
      If Tank_Level[mID_TQ_PSV_OD [k]] < vlmin Then
      {
        vlmin = Tank_Level[mID_TQ_PSV_OD [k]]
        mPSV_Selecionado[aID_PSV_OD] = k
      }
    }
    Inc k
  }
}
//zerar vetor mPSVs_Capazes_Fundeio
int y = 1
While y <= ArrayDimSize(mStatus_PSVs, 1) Do
{
  mPSVs_Capazes[y, vlaID_Local] = 0
  inc y
}

```

Sub Atualiza_consumo

```

int vlteste = 0
While 1 Do
{
  WAIT 1 hr
  vc1 = mDistribuicao_Consumo_IN[1]/24
  mVerifica_Consumo[1] = mDistribuicao_Consumo_IN[1]
  If clock() >= vTempo_Aquecimento and Tank_Level[mID_TQ_Cluster[1]] > vc1 Then INC
  mVolume_Consumido_Cluster[1],vc1
  vc2 = mDistribuicao_Consumo_IN[2]/24
  mVerifica_Consumo[2] = mDistribuicao_Consumo_IN[2]
  If clock() >= vTempo_Aquecimento and Tank_Level[mID_TQ_Cluster[2]] > vc2 Then INC
  mVolume_Consumido_Cluster[2],vc2
  vc3 = mDistribuicao_Consumo_IN[3]/24
  mVerifica_Consumo[3] = mDistribuicao_Consumo_IN[3]
  If clock() >= vTempo_Aquecimento and Tank_Level[mID_TQ_Cluster[3]] > vc3 Then INC
  mVolume_Consumido_Cluster[3],vc3
  vc4 = mDistribuicao_Consumo_IN[4]/24
  mVerifica_Consumo[4] = mDistribuicao_Consumo_IN[4]
  If clock() >= vTempo_Aquecimento and Tank_Level[mID_TQ_Cluster[4]] > vc4 Then INC
  mVolume_Consumido_Cluster[4],vc4
  vc5 = mDistribuicao_Consumo_IN[5]/24
  mVerifica_Consumo[5] = mDistribuicao_Consumo_IN[5]
  If clock() >= vTempo_Aquecimento and Tank_Level[mID_TQ_Cluster[5]] > vc5 Then INC
  mVolume_Consumido_Cluster[5],vc5
  vc6 = mDistribuicao_Consumo_IN[6]/24
  mVerifica_Consumo[6] = mDistribuicao_Consumo_IN[6]
  If clock() >= vTempo_Aquecimento and Tank_Level[mID_TQ_Cluster[6]] > vc6 Then INC
  mVolume_Consumido_Cluster[6],vc6
}

```

```

vc7 = mDistribuicao_Consumo_IN[7]/24
mVerifica_Consumo[7] = mDistribuicao_Consumo_IN[7]
If clock() >= vTempo_Aquecimento and Tank_Level[mID_TQ_Cluster[7]] > vc7 Then INC
mVolume_Consumido_Cluster[7],vc7
vc8 = mDistribuicao_Consumo_IN[8]/24
mVerifica_Consumo[8] = mDistribuicao_Consumo_IN[8]
If clock() >= vTempo_Aquecimento and Tank_Level[mID_TQ_Cluster[8]] > vc8 Then INC
mVolume_Consumido_Cluster[8],vc8
vc9 = mDistribuicao_Consumo_IN[9]/24
mVerifica_Consumo[9] = mDistribuicao_Consumo_IN[9]
If clock() >= vTempo_Aquecimento and Tank_Level[mID_TQ_Cluster[9]] > vc9 Then INC
mVolume_Consumido_Cluster[9],vc9
vc10 = mDistribuicao_Consumo_IN[10]/24
mVerifica_Consumo[10] = mDistribuicao_Consumo_IN[10]
If clock() >= vTempo_Aquecimento and Tank_Level[mID_TQ_Cluster[10]] > vc10 Then INC
mVolume_Consumido_Cluster[10],vc10
vc11 = mDistribuicao_Consumo_IN[11]/24
mVerifica_Consumo[11] = mDistribuicao_Consumo_IN[11]
If clock() >= vTempo_Aquecimento and Tank_Level[mID_TQ_Cluster[11]] > vc11 Then INC
mVolume_Consumido_Cluster[11],vc11
}

```

Sub gera_pedidos

```

While 1 Do
{
  Wait 0.2 hr
  Int c = 1
  Int vlcont = 0
  While c <= 11 Do
  {
    If mPedidos_Gerados[c] <> 0 Then
    {
      If Tank_Level[mID_TQ_Cluster[c]] < mPonto_Pedido_Cluster_IN[c] and mEstoque_Transito[c]
= 0 and mPedidos_Aguardando[c] = 0 and mStatus_Pedidos[mPedidos_Gerados[c],
vCol_Status_Pedido_mStatus_Pedidos, c] = vPedido_Entregue Then
      {
        Inc mCluster_Solicitante[c]
        Inc vlcont
        mhora_ultimo_pedido[c] = clock()
      }
    }
    Else
    {
      If Tank_Level[mID_TQ_Cluster[c]] < mPonto_Pedido_Cluster_IN[c] and mEstoque_Transito[c]
= 0 and mPedidos_Aguardando[c] = 0 Then
      {
        Inc mCluster_Solicitante[c]
        Inc vlcont
        mhora_ultimo_pedido[c] = clock()
      }
    }
  }
  Inc c
}
If vlcont > 0 Then Order 1 ePedido_OD To Setup
}

```

Sub programação_diaria

```
While 1 Do
{
  Wait 1 day
  Int i = 1
  Int vlCont_Ocorrencias_Barcos = 0
  Int vlCont_Ocorrencias_Clusters = 0
  Real vlRetorno
  While i <= vCont_PSV_OD Do
  {
    If mStatus_PSVs[i, vCol_Status_mStatus_PSVs] = vStatus_Ocioso Then
    {
      Inc vlCont_Ocorrencias_Barcos
    }
    Inc i
  }
  i = 1
  While i <= 11 Do
  {
    If mPedidos_Aguardando[i] > 0 Then Inc vlCont_Ocorrencias_Clusters
    Inc i
  }
  If vlCont_Ocorrencias_Clusters > 0 And vlCont_Ocorrencias_Barcos > 0 Then
  {
    sub_Exporta_Barcos()
    sub_Exporta_Clusters()
    vlRetorno = Xsub(OtimizadorDiesel, 1, 0)
    macImportar_Solucao
    vTem_Programacao = 1
  }
}
}
```

Macro macImporta_Solução

```
Int ii = 1
While ii <= ArrayDimSize(mDemanda_Restante_Otimizador,1) Do
{
  Read Demanda_Restante, mDemanda_Restante_Otimizador[ii,1]
  Read Demanda_Restante, mDemanda_Restante_Otimizador[ii,2]
  Inc ii
}
Reset Demanda_Restante
Close Demanda_Restante
Int jj = 1
While jj <= ArrayDimSize(mAlocacao,1) Do
{
  Read Alocacao, mAlocacao[jj,1]
  Read Alocacao, mAlocacao[jj,2]
  Inc jj
}
Reset Alocacao
Close Alocacao
```

Macro macCalculo_Final

```
Int i = 1
Int j = 1
While i <= 10 Do
```

```

{
  If mPSVs_Existentes_Sistema[i] > 0 Then
  {
    mCusto_Operacional[i] = ((vTempo_Simulacao-mTempo_Ocioso_PSV_Total[i]) *187.5)
    Inc vCusto_Operacional_PSV, ((vTempo_Simulacao-mTempo_Ocioso_PSV_Total[i]) *187.5)
    mCusto_Fixo_PSV[i] = 1250 * vTempo_Simulacao
  }
  mVolume_Medio_PSV[i] =
  ((Tank_Statistics[mID_TQ_PSV_OD[i],3])/vTempo_Simulacao)/mEstoques_PSVs_OD_IN[i,2]
  If mSoma_Tempo_Ciclo[i]> 0 and mNumero_Visitas_FundeioNT[i] >0 Then
  {
    mMedia_Tempo_Ciclo[i] = mSoma_Tempo_Ciclo[i]/mNumero_Visitas_FundeioNT[i]
  }
  If mPSVs_Existentes_Sistema[i] >0 Then
  {
    mTempo_Total_Contagem[i] = clock() - mTempo_Primeiro_Registro[i]
    mTempo_Medio_Ocioso_PSV[i] = mTempo_Ocioso_PSV_Total[i]/mTempo_Total_Contagem[i]
    mTempo_Medio_Uso_PSV[i] = mTempo_Uso_PSV_Total[i]/mTempo_Total_Contagem[i]
  }
  Inc i
}
While j <=11 Do
{
  If mFlag_Cluster[j] = 0 Then
  {
    mDuracao_StockOut[mCont_StockOut[j],j] = clock()- mHora_StockOut[mCont_StockOut[j], j]
    Inc mTotal_StockOut_Cluster[j], mDuracao_StockOut[mCont_StockOut[j],j]
    If j = 2 Or j = 10 Or j = 11 Then
    {
      Inc mCusto_Falta[j], mDuracao_StockOut[mCont_StockOut[j],j]* mCusto_Falta_IN[j]
    }
    Else
    {
      Inc mCusto_Falta[j],
      (mDuracao_StockOut[mCont_StockOut[j],j]*mCusto_Falta_IN[j])*(exp(0.00001088*mDuracao_StockOut[mCont_StockOut[j],j]) -1)
    }
  }
  If mFlag_Acima_Seguranca[j] = 0 Then
  {
    mDuracao_Estoque_Risco[mCont_Estoque_Risco[j],j] = clock()-
    mHora_Entra_Risco[mCont_Estoque_Risco[j], j]
    Inc mTotal_Risco_Cluster[j], mDuracao_Estoque_Risco[mCont_Estoque_Risco[j],j]
  }
  mCusto_Estoque[j] = ((Tank_Statistics[mID_TQ_Cluster[j],3])/vTempo_Simulacao)*(0.75)*1000)*
  (exp(0.2) -1)
  Inc vCusto_Estoque, ((Tank_Statistics[mID_TQ_Cluster[j],3])/vTempo_Simulacao)*(0.75)*1000)*
  (exp(0.2) -1)
  mVolume_Medio_Cluster[j] =
  ((Tank_Statistics[mID_TQ_Cluster[j],3])/vTempo_Simulacao)/mEstoques_Clusters_IN[j,2]
  If mSoma_Intervalo_entre_Visitas[j] >0 And mIndice_Visita_Cluster_aquecimento[j] >0 Then
  {
    mMedia_Intervalo_entre_Visitas[j] = mSoma_Intervalo_entre_Visitas[j]/
    mIndice_Visita_Cluster_aquecimento[j]
  }
  mNivel_servico_StockOut[j] = (vTempo_Simulacao -
  mTotal_StockOut_Cluster[j])/vTempo_Simulacao
  mNivel_servico_Risco[j] = (vTempo_Simulacao - mTotal_Risco_Cluster[j])/vTempo_Simulacao
  Inc j
}
}

```

```

If vExperimento = 2 Then
{
  If (mCont_StockOut[1] = 0 And mCont_StockOut[2] = 0 And mCont_StockOut[3] = 0 and
mCont_StockOut[4] = 0 and mCont_StockOut[5] = 0 and mCont_StockOut[6] = 0 And
mCont_StockOut[7] = 0 And mCont_StockOut[8] = 0 And mCont_StockOut[9] = 0 And
mCont_StockOut[10] = 0 and mCont_StockOut[11] = 0) Then
  {
    j = 1
    While j <= 11 Do
    {
      If mCont_StockOut[j] = 0 Then
      {
        Inc vSoma_Nivel_Servico_Minimo, mNivel_servico_Risco[j]
        Inc vOcorrencia_Nivel_Servico_Minimo
      }
      Inc j
    }
    If vOcorrencia_Nivel_Servico_Minimo > 0 Then vMedia_Nivel_Servico_Minimo =
vSoma_Nivel_Servico_Minimo/vOcorrencia_Nivel_Servico_Minimo
    If vModelo_SobDemanda = 1 and vOcorrencia_Nivel_Servico_Minimo >0 Then
    {
      mCusto_Risco[11] = 10**9*(1 - vMedia_Nivel_Servico_Minimo)+ 10**7
      If mCusto_Risco[11] < 0 Then mCusto_Risco[11] = 0
      Inc vcusto_risco, mCusto_Risco[11]
    }
    If vModelo_SobDemanda = 0 and vOcorrencia_Nivel_Servico_Minimo >0 and
vMedia_Nivel_Servico_Minimo <1 Then
    {
      mCusto_Risco[11] = 7*10**9*(1 - vMedia_Nivel_Servico_Minimo)+9*10**6
      If mCusto_Risco[11] < 0 Then mCusto_Risco[11] = 0
      Inc vcusto_risco, mCusto_Risco[11]
    }
  }
}
mac_Lead_Time_Medio
mac_Lead_Time_Medio_pos_Aquecimento
mac_Planificacao
macNivel_Final

```

Apêndice F – Código fonte da DLL e das formulações MIP

A seguir são apresentados o código fonte em C++ do arquivo DLL desenvolvido para execução da otimização dentro do simulador e os códigos das formulações de programação inteira mista (MIP), desenvolvido na linguagem *Python*.

Código fonte DLL

```

#include "cabecalho_dll.h"
#ifdef __MINGW32__
  #ifndef MINGW_DLL_H__
  #define MINGW_DLL_H__
  #endif
#endif

```

```

#endif
typedef struct PARAM {
    int msgbox;
    #ifdef _WIN32
        HWND hWndFrame;
    #endif
} Param;
using namespace std;
void MensagemNoPromodel(Param* Parametros, string Titulo, string Mensagem);
string GetPath();

#ifdef _WIN32
    __declspec(dllexport) double __cdecl Otimizador(void* p)
#else
    double Otimizador(void* p)
#endif
{
    // --- Variáveis Locais --- //
    PyObject *main_module, *dict;
    PyObject *otimizador_python, *resultado_py;
    Param* p_parametros
    string path_base;                // Diretório onde estão os arquivos .py, .csv e .dll
    string path_python;              // Path do arquivo .py
    string mensagem;                // Mensagem a ser impressa
    int status;                      // Status retornado pelo otimizador
    string str_status;              // String que indica o status do otimizador
    FILE *python;

    // --- Inicialização --- //
    setlocale(LC_ALL, "");
    path_base = GetPath();
    path_python = path_base + "SimOpt.py";
    p_parametros = (Param*)p; // Structure overlay (usando typecast)

    // -- Python -- //
    Py_Initialize();
    main_module = PyImport_AddModule("__main__");
    dict = PyModule_GetDict(main_module);

    python = fopen(path_python.c_str(), "r");

    // -- Abrir o arquivo do script em Python -- //
    if (python == NULL) {
        mensagem = "O arquivo `" + path_python + "` não foi encontrado.";
        MensagemNoPromodel(p_parametros, "Erro do arquivo", mensagem);
        return (double) 0;
    }
    PyRun_SimpleFile(python, path_python.c_str());

    otimizador_python = PyDict_GetItemString(dict, "Otimizador");
    resultado_py = PyObject_CallFunction(otimizador_python, "s", path_base.c_str());

    status = PyLong_AsLong(resultado_py);
    if (status == 0)
        str_status = "Problema não resolvido.";
    else if (status == 1)
        str_status = "Solução ótima encontrada.";
    else if (status == -1)
        str_status = "Problema inviável.";
    else if (status == -2)

```

```

    str_status = "Problema ilimitado.";
else if (status == -3)
    str_status = "Problema indefinido.";
else if (status == -4)
    str_status = "Os arquivos `.csv` não foram encontrados.";
else if (status == -5)
    str_status = "O pacote Pulp não está instalado.";
else if (status == -6)
    str_status = "O pacote Pandas não está instalado.";

fclose(python);
Py_Finalize();

// -- Mensagem -- //
if (p_parametros->msgbox == 1 || status != 1) {
    mensagem = "Status do Otimizador: " + str_status;
    // mensagem = ponteiro_file + path_python + "\n\n" + path_csv + "\n\n" + to_string(status);
    MensagemNoPromodel(p_parametros, "Programação das Embarcações", mensagem);
}
return (double) status;
}
void MensagemNoPromodel(Param* Parametros, string Titulo, string Mensagem)
{
    // ----- Variáveis Locais ----- //
#ifdef _WIN32
    LPWSTR w_mensagem, w_titulo;
    // Mensagem
    w_mensagem = new wchar_t[Mensagem.size()];
    MultiByteToWideChar(CP_UTF8, 0, Mensagem.c_str(), -1, w_mensagem,
        Mensagem.size());
    // Título
    w_titulo = new wchar_t[Titulo.size()];
    MultiByteToWideChar(CP_UTF8, 0, Titulo.c_str(), -1, w_titulo,
        Titulo.size());

    MessageBoxW(GetTopWindow(Parametros->hWndFrame), w_mensagem, w_titulo,
MB_OK);
#else
    cout << Titulo << ":" << endl << "  " << Mensagem << endl;
#endif
}
string GetPath()
{
#ifdef _WIN32
    char buffer[MAX_PATH];
    GetModuleFileName( NULL, buffer, MAX_PATH );
    string::size_type pos = string( buffer ).find_last_of( "\\\" );
    //return string( buffer ).substr( 0, pos ) + "\\\";
    return "C:\\Promodel\\";
#endif

#ifdef __APPLE__
    int ret;
    pid_t pid;
    char pathbuf[PROC_PIDPATHINFO_MAXSIZE];
    string path;
    pid = getpid();
    ret = proc_pidpath (pid, pathbuf, sizeof(pathbuf));
    if (ret <= 0)
        cerr << "Erro na identificação do path (Funcao proc_pidpath)." << endl;
#endif
}

```

```

path = string(pathbuf);
path = path.substr(0, path.find_last_of('/') + 1);
return path;
#else
cerr << "Erro: O sistema utilizado não é nem Mac OS X nem Microsoft \
Windows. Erros podem acontecer." << endl;
return "";
#endif
}

```

Formulação MIP – Min. Risco consumo E.Segurança

```

def Otimizador(path_base="C:\\Promodel"):
    try:
        import pulp as plp
    except:
        return -5
    try:
        import pandas as pd
    except:
        return -6
    try:
        ClustersData = pd.read_csv(path_base + "\\data_clusters.csv",sep=";")
        BarcosData = pd.read_csv(path_base + "\\data_barcos.csv",sep=";")
        NavegacaoData = pd.read_csv(path_base + "\\data_navegacao.csv",sep=";")
    except:
        return -4

    Clusters = list(ClustersData["Cluster"].dropna())
    Barcos = list(BarcosData["PSV"].dropna())
    prob = plp.LpProblem("Aloc_barco_cluster", plp.LpMinimize)
    x = plp.LpVariable.dicts("x",(Barcos, Clusters),0, None,plp.LpBinary)
    dr = plp.LpVariable.dicts("dr",(Clusters),0, None, plp.LpContinuous)

    prob+= plp.lpSum([(ClustersData['Estoque_Atual'][Clusters.index(i)]-
(ClustersData['Consumo'][Clusters.index(i)]*(-
NavegacaoData.loc[int(BarcosData.loc[Barcos.index(j)][2]-1)][i])) -
ClustersData['Estoque_Seguranca'][Clusters.index(i)]*x[j][i] for j in Barcos for i in Clusters])+
plp.lpSum([dr[i] for i in Clusters])

    prob+=plp.lpSum(x[j][i] for j in Barcos for i in Clusters) >= 1
    for j in Barcos:
        prob+= plp.lpSum ([x[j][i]*ClustersData['Qtde_pedida'][Clusters.index(i)]- dr[i] for i in
Clusters]) <= BarcosData['Estoque_Atual'][Barcos.index(j)]

    for i in Clusters:
        prob+= plp.lpSum([x[j][i]* BarcosData['Estoque_Atual'][Barcos.index(j)] for j in Barcos])>=
(ClustersData['Qtde_pedida'][Clusters.index(i)] - dr[i])

    for j in Barcos:
        prob+= plp.lpSum([x[j][i] for i in Clusters]) <= 1

    for i in Clusters:
        prob+= plp.lpSum([x[j][i] for j in Barcos]) <= 1
    prob.solve()

```

```

columns1 = ["Cluster", "DR"]
columns2 = ["Cluster", "PSV"]
sollist1 = []
sollist2 = []
for v in prob.variables():
    if v.name.split("_")[0] == "dr":
        dr = v.varValue
        soldict1 = {"Cluster": "", "DR": ""}
        soldict1["Cluster"] = v.name.split("_")[1]
        soldict1["DR"] = dr
        sollist1.append(soldict1)
for v in prob.variables():
    if v.name.split("_")[0] == "x" and v.varValue != 0 :
        soldict2 = {"Cluster": "", "PSV": ""}
        soldict2["Cluster"] = v.name.split("_")[2]
        soldict2["PSV"] = v.name.split("_")[1]
        sollist2.append(soldict2)
datasolDR = pd.DataFrame(sollist1,columns = columns1)
datasolAloc = pd.DataFrame(sollist2,columns = columns2)
datasolDR.to_csv(path_base + "\DemandaRestante.csv", sep=";", index=False, encoding='ascii')
datasolAloc.to_csv(path_base + "\Alocacao.csv", sep=";", index=False, encoding='ascii')
return prob.status

```

Formulação MIP – Max. Autonomia

```

def Otimizador(path_base="C:\Promodel"):
    try:
        import pulp as plp
    except:
        return -5
    try:
        import pandas as pd
    except:
        return -6
    try:
        ClustersData = pd.read_csv(path_base + "\data_clusters.csv",sep=";")
        BarcosData = pd.read_csv(path_base + "\data_barcos.csv",sep=";")
        NavegacaoData = pd.read_csv(path_base + "\data_navegacao.csv",sep=";")
    except:
        return -4

    Clusters = list(ClustersData["Cluster"].dropna())
    Barcos = list(BarcosData["PSV"].dropna())

    prob = plp.LpProblem("Aloc_barco_cluster", plp.LpMinimize)
    x = plp.LpVariable.dicts("x",(Barcos, Clusters),0, None,plp.LpBinary)
    dr = plp.LpVariable.dicts("dr",(Clusters),0, None, plp.LpContinuous)

    prob+=
    plp.lpSum([((ClustersData['Estoque_Atual'][Clusters.index(i)]+ClustersData['Consumo'][Clusters.index(i)
    ])*NavegacaoData.loc[int(BarcosData.loc[Barcos.index(j)])[2]-
    1])[i])/ClustersData['Consumo'][Clusters.index(i)]*x[j][i] for j in Barcos for i in Clusters])+
    plp.lpSum([dr[i]*(1/ClustersData['Consumo'][Clusters.index(i)]) for i in Clusters])

    prob+=plp.lpSum(x[j][i] for j in Barcos for i in Clusters) >= 1

    for j in Barcos:

```

```

prob+= plp.lpSum ([x[j][i]*ClustersData['Qtde_pedida'][Clusters.index(i)] - dr[i] for i in
Clusters]) <= BarcosData['Estoque_Atual'][Barcos.index(j)]

for i in Clusters:
    prob+= plp.lpSum([x[j][i]* BarcosData['Estoque_Atual'][Barcos.index(j)] for j in Barcos])>=
(ClustersData['Qtde_pedida'][Clusters.index(i)] - dr[i])

for j in Barcos:
    prob+= plp.lpSum([x[j][i] for i in Clusters]) <= 1

for i in Clusters:
    prob+= plp.lpSum([x[j][i] for j in Barcos]) <= 1

prob.solve()
columns1 = ["Cluster", "DR"]
columns2 = ["Cluster", "PSV"]
sollist1 = []
sollist2 = []
for v in prob.variables():
    if v.name.split("_")[0] == "dr":
        dr = v.varValue
        soldict1 = {"Cluster": "", "DR": "" }
        soldict1["Cluster"] = v.name.split("_")[1]
        soldict1["DR"] = dr
        sollist1.append(soldict1)
for v in prob.variables():
    if v.name.split("_")[0] == "x" and v.varValue != 0 :
        soldict2 = {"Cluster": "", "PSV": ""}
        soldict2["Cluster"] = v.name.split("_")[2]
        soldict2["PSV"] = v.name.split("_")[1]
        sollist2.append(soldict2)
datasolDR = pd.DataFrame(sollist1,columns = columns1)
datasolAloc = pd.DataFrame(sollist2,columns = columns2)
datasolDR.to_csv(path_base + "\DemandaRestante.csv", sep=";", index=False, encoding='ascii')
datasolAloc.to_csv(path_base + "\Alocacao.csv", sep=";", index=False, encoding='ascii')
return prob.status

```