

Área Departamental de Engenharia Eletrotécnica de Energia e Automação

Desenvolvimento de Simulador para Orçamentação de Infraestruturas Elétricas

EURICO ANDRÉ MENDES DA SILVA CLEMENTE
Licenciado em Engenharia Eletrotécnica

Dissertação para obtenção do grau de Mestre em
Engenharia Eletrotécnica - Ramo de Energia

Orientadores:

Professora Adjunta Cristina Inês Camus
Equip. Assist. 2º Triénio Eduardo Adelino Mateus Nunes Eusébio

Júri:

Presidente: Professor Luís Manuel dos Santos Redondo

Vogais:

Professor: Cristina Inês Camus
Professor: João Hermínio Ninitas Lagarto

Dezembro de 2015

Dissertação realizada sob orientação de:

Professora Adjunta Doutora Cristina Inês Camus

Área Departamental de Engenharia Eletrotécnica de Energia e Automação

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Equip. Assist. 2º Triénio Mestre Eduardo Adelino Mateus Nunes Eusébio

Área Departamental de Engenharia Eletrotécnica de Energia e Automação

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Agradecimentos

Em primeiro lugar quero agradecer à professora Cristina Inês Camus e ao professor Eduardo Adelino Mateus Nunes Eusébio, da Área Departamental de Engenharia Eletrotécnica de Energia e Automação, por aceitarem a orientação desta dissertação e por toda a disponibilidade demonstrada em ajudar-me sempre que precisei e em responder a todas as minhas dúvidas, mas acima de tudo pelo espírito de excelência que me inculcaram. A vossa disponibilidade, dedicação e espírito crítico não serão esquecidos.

Por último, mas não menos importante, quero agradecer à minha família pela compreensão e apoio constante em todos estes anos de curso.

Resumo

A forma padrão para responder aos pedidos de cotação, para a realização de empreitadas/obras de qualquer tipo (especialidade), implica a realização de uma proposta comercial (valor e condições), sendo que estas têm como base a elaboração de um orçamento, denominado orçamento de obra. A melhor forma de otimizar a resposta aos pedidos de cotação é otimizando o orçamento de obra.

Face aos novos desafios colocados pelos países em grande desenvolvimento estrutural no âmbito da construção civil, nomeadamente os países emergentes, tem sido possível manter viva a construção civil portuguesa durante a crise. Esta mesma oportunidade (“fora de portas”) criou um novo tipo de proposta comercial, à qual Portugal está a responder ainda deficientemente.

Constata-se que o elevado número de pedidos de propostas bem como a dimensão das obras, criaram uma grande pressão sobre as respostas (orçamentos) a dar. Para evitar essa situação e para melhorar todo este processo foi elaborado o simulador agora descrito.

O simulador tem como nome Business Assistance (tutorial disponível em https://www.youtube.com/watch?v=X4ZqR1C_yAI)

Esta dissertação incide, na implementação de expressões de cálculo de orçamentação, bem como na elaboração e conceção de um programa específico. Realizado para facilitar a elaboração de propostas, no qual foram utilizadas as linguagens de programação (Excel, Java, SQL e HTML).

Palavras-chave

Orçamentos
Otimização de cálculo
Propostas comerciais
Programa Business Assistance

Abstract

The standard way to answer a requests, for the execution of Construction (shops, schools and other), involves carrying out a commercial proposal (value and conditions), and these are based on a budget. The best way to optimize this response is optimizing the budget itself.

Facing new challenges created by countries in structural development (Construction), just like emerging markets, it was possible to keep alive the Portuguese construction during the main crisis. This same opportunity ("out of doors") created a new type of business proposal, to which Portugal is responding inadequately.

It appears that the large number of proposed and the size of them, created a lot of pressure on the answers (budgets) to respond. To avoid this situation and to improve the whole process was developed the simulator now described.

The simulator has the name Business Assistance (tutorial available https://www.youtube.com/watch?v=X4ZqR1C_yAI)

This dissertation focuses on the development of ways to calculation (mathematics) as well as the development of a specific program designed to facilitate the preparation of proposals, using a variety of programming languages (Excel, Java, SQL and HTML).

Keywords

Budgets
Optimization calculation
Business proposals
Business Assistance Program

ÍNDICE

Capítulo 1. Introdução	1
1.1 - Enquadramento.....	3
1.2 - Objetivo do Estudo.....	3
1.3 - Motivação.....	4
1.4 - Organização do Texto	4
1.5- Notação.....	6
Capítulo 2. Estado da arte e otimização das respostas.....	7
2.1 – Estado da arte.....	9
2.1.1 – <i>Software OranGest</i> Obras.....	9
2.1.1.1 – Aspectos positivos do simulador.....	10
2.1.1.2 – Aspectos negativos do simulador.....	10
2.1.2 – <i>Software Quote Roller</i>	11
2.1.2.1 – Aspectos positivos do simulador.....	12
2.1.2.2 – Aspectos negativos do simulador.....	12
2.1.3 – Comparação de <i>Softwares</i>	12
2.2 – Otimização das respostas.....	13
2.2.1 – Introdução as PCO.....	13
2.2.1.1 – Fator tempo.....	14
2.2.1.2 – Fator monetário.....	15
2.2.2 – Elementos otimizados no simulador desenvolvido	15
Capítulo 3. Ferramentas utilizadas na conceção do simulador	17
3.1 – Ferramentas utilizadas	19
3.1.1 – <i>Excel</i>	19
3.1.2 – <i>Java</i>	20
3.1.3 – <i>SQL</i>	20
3.2 – Conceção da simulação	20
3.2.1 – Input.....	22
3.2.1.1 – Bases para identificação da proposta.....	22
3.2.1.2 – Bases de cálculo para orçamentação	23
3.2.1.3 – Bases Auxiliares	26
3.2.1.4 – Bases Contentores.....	27
3.2.1.5 – Bases para identificação dos elementos referentes ao mapa de quantidades	28

3.2.2 – Cálculos e <i>Output</i>	37
3.2.2.1 – Bases Auxiliares	38
3.2.2.2 – Bases Contentores.....	42
3.2.2.3 – Bases para identificação dos elementos referentes ao mapa de quantidades	45
3.2.2.3 – Bases para resumo da orçamentação	67
3.2.2.4 – Restantes elementos de cálculo	74
Capítulo 4. Caso de estudo - Cálculos e Output.....	79
4.1– Variáveis introduzidas	80
4.2 – Cálculos comparativos das variáveis	82
Capítulo 5. Modo de utilização	89
5 – Introdução ao <i>UI (user interface)</i>	91
5.1 – <i>Client</i>	92
5.2 – <i>Materials</i>	94
5.3 – <i>Proposal</i>	96
Capítulo 6. Conclusões e desenvolvimentos futuros	99
6.1 – Principais conclusões.....	101
6.2 – Desenvolvimentos futuros	102
Referências Bibliográficas	105
Referências bibliográficas Programas	107
Referências bibliográficas Livros e apontamentos	107
Apêndices	
Apêndice 1 - Identificação das variáveis.....	109
1.1 - Input	111
1.2 - Cálculos e Output	115
Apêndice 2 - Programa Linguagem Java	123
2.1 - Bases para identificação da proposta (<i>Proposal</i>)	125
2.2 - Bases de cálculo para orçamentação (<i>BudgetBase</i>)	131
2.3 - Bases Auxiliares (<i>AuxiliarBase</i>)	139
2.4 - Bases Contentores (<i>Containers</i>)	143
2.5 - Bases para identificação dos elementos referentes ao mapa de quantidades (<i>Material</i>).....	147
2.6 - Cálculos (<i>FormulaUtils</i>)	153
Apêndice 3 - Programa de teste aos valores	181

Lista de Figuras

Figura 1.1 - Organização da dissertação.....	5
Figura 2.1 - Janela <i>software OranGest Obras (vista geral)</i>	9
Figura 2.2 - Janela <i>software OranGest Obras (ficha pessoal)</i>	10
Figura 2.3 - Janela <i>software Quote Roller (vista geral)</i>	11
Figura 2.4 - Janela <i>software Quote Roller (edit price list)</i>	12
Figura 2.5 - Relação cliente fornecedor.....	14
Figura 3.1 - Estrutura do simulador.....	19
Figura 3.2 - Contexto da utilização <i>Java</i>	20
Figura 3.3 - Diagrama <i>Input/Output</i>	21
Figura 3.4 - Organização estrutural das variáveis (<i>Input/Output</i>)	21
Figura 3.5 - Características referências para contentores 20”	27
Figura 3.6 - Características referências para contentores 40”	27
Figura 3.7 - Extensão da folha de cálculo	29
Figura 3.8 - Divisórias da folha de cálculo.....	30
Figura 5.1 - Janela inicial	91
Figura 5.2 - Composição vertente UI (sistema).....	91
Figura 5.3 - Composição vertente UI (aplicada)	92
Figura 5.4 - Janela <i>New Client</i>	93
Figura 5.5 - Janela <i>Client Details</i>	93
Figura 5.6 - Janela <i>Client Results</i>	94
Figura 5.7 - Janela <i>New Material</i>	95
Figura 5.8 - Janela <i>Material Details</i>	95
Figura 5.9 - Janela <i>Material Results</i>	96
Figura 5.10 - Janela <i>New Proposal</i>	97
Figura 5.11 - Janela <i>Proposal Details</i>	97
Figura 5.12 - Janela <i>Proposal Results</i>	98

Lista de Tabelas

Tabela 2.1 - Comparação de <i>Softwares</i> (disponíveis mercado).....	13
Tabela 3.1 - Bases para identificação (variáveis)	22
Tabela 3.2 - Bases para identificação (exemplo).....	23
Tabela 3.3 - Bases de cálculo para orçamentação (variáveis)	24
Tabela 3.4 - Bases de cálculo para orçamentação (exemplo).....	25
Tabela 3.5 - Bases auxiliares (variáveis)	26
Tabela 3.6 - Bases auxiliares (exemplo).....	26
Tabela 3.7 - Bases contentores (variáveis)	28
Tabela 3.8 - Bases contentores (exemplo).....	28
Tabela 3.9 - Valores possíveis para as variáveis i_{31} e i_{32}	29
Tabelas 3.10 - Bases para identificação dos elementos (variáveis).....	30
Tabela 3.11 - Valores possíveis para as variáveis i_{35}	33
Tabela 3.12 - Bases para identificação dos elementos (exemplo).....	34
Tabela 3.13 - Resumo da orçamentação (variáveis).....	68
Tabela 3.14 - Resumo da orçamentação (exemplo)	73
Tabela 4.1 - Valores das variáveis <i>Inputs</i>	80
Tabela 4.2 - Cálculos comparativos do valor das variáveis	82
Tabela 6.1 - Comparação de <i>Softwares</i> (conclusões)	102

Lista de Equações

Equação 1 – Cálculo A1	38
Equação 2 - Cálculo A2.....	38
Equação 3 - Cálculo A3.....	38
Equação 4 - Cálculo A4.....	39
Equação 5 - Cálculo A5.....	39
Equação 6 - Cálculo A6.....	39
Equação 7 - Cálculo A7.....	39
Equação 8 - Cálculo A8.....	40
Equação 9 - Cálculo A9.....	40
Equação 10 - Cálculo A10.....	40
Equação 11 - Cálculo A11	40
Equação 12 - Cálculo A12.....	41
Equação 13 - Cálculo A13.....	41
Equação 14 - Cálculo A14.....	41
Equação 15 - Cálculo A15.....	42
Equação 16 - Cálculo A16.....	42
Equação 17 - Cálculo A17.....	42
Equação 18 - Cálculo A18.....	43
Equação 19 - Cálculo A19.....	43
Equação 20 - Cálculo A20.....	43
Equação 21 - Cálculo A21	44
Equação 22 - Cálculo A22.....	44
Equação 23 - Cálculo A23.....	45
Equação 24 - Cálculo A24.....	45
Equação 27 - Cálculo A27.....	45
Equação 28 - Cálculo A28.....	46
Equação 29 - Cálculo A29.....	46
Equação 30 - Cálculo A30.....	46
Equação 31 - Cálculo A31	47
Equação 32 - Cálculo A32.....	47
Equação 33 - Cálculo A33.....	47
Equação 34 - Cálculo A34.....	48

Equação 35 - Cálculo A35.....	48
Equação 36 - Cálculo A36.....	48
Equação 37 - Cálculo A37.....	48
Equação 38 - Cálculo A38.....	49
Equação 39 - Cálculo A39.....	49
Equação 40 - Cálculo A40.....	50
Equação 41 - Cálculo A41.....	50
Equação 42 - Cálculo A42.....	50
Equação 43 - Cálculo A43.....	51
Equação 44 - Cálculo A44.....	51
Equação 45 - Cálculo A45.....	51
Equação 46 - Cálculo A46.....	52
Equação 47 - Cálculo A47.....	52
Equação 48 - Cálculo A48.....	52
Equação 49 - Cálculo A49.....	53
Equação 50 - Cálculo A50.....	53
Equação 51 - Cálculo A51.....	53
Equação 52 - Cálculo A52.....	54
Equação 53 - Cálculo A53.....	54
Equação 54 - Cálculo A54.....	54
Equação 55 - Cálculo A55.....	55
Equação 56 - Cálculo A56.....	55
Equação 57 - Cálculo A57.....	55
Equação 58 - Cálculo A58.....	56
Equação 59 - Cálculo A59.....	56
Equação 60 - Cálculo A60.....	56
Equação 61 - Cálculo A61.....	56
Equação 62 - Cálculo A62.....	57
Equação 63 - Cálculo A63.....	57
Equação 64 - Cálculo A64.....	57
Equação 65 - Cálculo A65.....	58
Equação 66 - Cálculo A66.....	58
Equação 67 - Cálculo A67.....	58
Equação 68 - Cálculo A68.....	58

Equação 69 - Cálculo A69.....	59
Equação 70 - Cálculo A70.....	59
Equação 71 - Cálculo A71.....	59
Equação 72 - Cálculo A72.....	59
Equação 73 - Cálculo A73.....	60
Equação 74 - Cálculo A74.....	60
Equação 75 - Cálculo A75.....	60
Equação 76 - Cálculo A76.....	60
Equação 77 - Cálculo A77.....	61
Equação 78 - Cálculo A78.....	61
Equação 79 - Cálculo A79.....	61
Equação 80 - Cálculo A80.....	62
Equação 81 - Cálculo A81.....	62
Equação 82 - Cálculo A82.....	62
Equação 83 - Cálculo A83.....	62
Equação 84 - Cálculo A84.....	63
Equação 85 - Cálculo A85.....	63
Equação 86 - Cálculo A86.....	63
Equação 87 - Cálculo A87.....	63
Equação 88 - Cálculo A88.....	64
Equação 89 - Cálculo A89.....	64
Equação 90 - Cálculo A90.....	64
Equação 91 - Cálculo A91.....	65
Equação 92 - Cálculo A92.....	65
Equação 93 - Cálculo A93.....	65
Equação 94 - Cálculo A94.....	65
Equação 95 - Cálculo A95.....	66
Equação 96 - Cálculo A96.....	66
Equação 97 - Cálculo A97.....	66
Equação 98 - Cálculo A98.....	67
Equação 99 - Cálculo A99.....	67
Equação 100 - Cálculo A100.....	68
Equação 101 - Cálculo A101.....	69
Equação 102 - Cálculo A102.....	69

Equação 103 - Cálculo A103.....	69
Equação 104 - Cálculo A104.....	69
Equação 105 - Cálculo A105.....	70
Equação 106 - Cálculo A106.....	70
Equação 107 - Cálculo A107.....	70
Equação 108 - Cálculo A108.....	70
Equação 109 - Cálculo A109.....	71
Equação 110 - Cálculo A110.....	71
Equação 111 - Cálculo A111.....	71
Equação 112 - Cálculo A112.....	72
Equação 113 - Cálculo A113.....	72
Equação 114 - Cálculo A114.....	72
Equação 115 - Cálculo A115.....	72
Equação 116 - Cálculo A116.....	73
Equação 117 - Cálculo A117.....	73
Equação 118 - Cálculo A118.....	74
Equação 119 - Cálculo A119.....	74
Equação 120 - Cálculo A120.....	75
Equação 121 - Cálculo A121.....	75
Equação 122 - Cálculo A122.....	75
Equação 123 - Cálculo A123.....	75
Equação 124 - Cálculo A124.....	76
Equação 125 - Cálculo A125.....	76
Equação 126 - Cálculo A126.....	76
Equação 127 - Cálculo A127.....	76

Lista de Abreviaturas

SIGLAS:

CIF	- <i>Cost, insurance and freight</i>
DEST	- País de destino
EMP	- Empresa utilizadora do simulador
FCL	- <i>Full container load</i>
FOB	- <i>Free on board</i>
HTML	- <i>Hyper text markup language</i>
ID	- <i>Identity document</i>
IRT	- Imposto sobre os rendimentos do trabalho
MDJ	- Memória descritiva e justificativa
ODO	- Orçamento de obra
OGO	- <i>OranGest OBRAS</i>
ORIG	- País de origem
PC	- <i>Personal computer</i>
PC´	- Preço de custo
PCO	- Propostas comerciais
PV	- Preço de venda
QR	- <i>Quote Roller</i>
SQL	- <i>Structured Query Language</i>
UI	- <i>User interface</i>

UNIDADES:

€	- Unidade monetária preferencial europeia (Euro)
\$	- Unidade monetária americana (Dólar)

CAPÍTULO

1

Introdução

Neste capítulo é feita uma introdução sucinta sobre o fundamento da presente dissertação, relativa à elaboração de propostas comerciais para a realização de obras e detecção das suas anomalias, bem como a necessidade de otimizar todo o processo do seu desenvolvimento. Apresenta-se, ainda, o enquadramento das mesmas e alguns exemplos práticos para a sua utilização. Neste capítulo terá também lugar uma explicação de como será elaborada a solução para colmatar as anomalias encontradas (desenvolvimento de simulador para orçamentação), assim como a notação utilizada ao longo do trabalho.

1.1 - Enquadramento

No mercado liberalizado as empresas de todas as nacionalidades podem concorrer a obras nos mais variados locais e continentes. Isto significa que países, como Portugal, podem concorrer, através de propostas comerciais (PCO), a obras originárias de outros países como, por exemplo, Angola. Estas PCO têm como objetivo a orçamentação de obras.

As obras em si surgem como motor de uma sociedade em desenvolvimento, pois todas as sociedades precisam de ampliar as suas instalações/infraestruturas ou mesmo criar novas de raiz, com os mais variados propósitos.

Desta forma, pode-se definir que as PCO servem para uma determinada organização ou empresa concorrer a uma construção ou reparação de uma instalação (escolas, fábricas, centros comerciais, entre outros). Importa, ainda, referir que nas PCO estão indicados os preços e as condições comerciais, observando as regras da concorrência, a lei geral e os regulamentos aplicáveis.

1.2 - Objetivo do Estudo

O presente trabalho tem por objetivo a elaboração de um simulador para a realização de estudos sobre a orçamentação de infraestruturas, nos mais variados contextos, como por exemplo do setor elétrico. O simulador será elaborado de forma ao utilizador dispor de uma plataforma de fácil interação bem como livre utilização.

Neste trabalho pretende-se refletir sobre a orçamentação e otimizar todo o processo conducente à mesma, em tudo o que se refere à determinação de valores de custo de materiais, mão-de-obra, fórmulas de revisão de preço. O simulador produzirá um documento com o resumo detalhado do estudo para utilização junto de empresas ou clientes, bem como para o respetivo utilizador.

O simulador será elaborado através de ferramentas computacionais e matemáticas. Em suma, esta dissertação visa evidenciar a importância da orçamentação de uma obra e a aplicação em contexto exemplificativo de um simulador que facilite e otimize o processo de orçamentação.

1.3 - Motivação

A motivação para a realização deste estudo deve-se ao facto de, na minha experiência profissional, ter tido passagem por uma empresa de orçamentação de obras, na qual havia um constante esforço para acompanhar e realizar todas as PCO. Pela experiência adquirida é possível afirmar que apenas um terço das propostas obtinha resposta. Tal se devia às dificuldades de realização de um orçamento preciso e elaborado em tempo útil.

Tendo em conta os montantes envolvidos em cada proposta (casa dos milhares ou milhões de euros), a percentagem de resposta é bastante desfavorável, podendo afirmar-se mesmo como sendo negativa.

Desta forma, pretende-se com o trabalho desta dissertação, fornecer uma nova base/pilar para a orçamentação de obras no seu carácter geral.

1.4 - Organização do Texto

O segundo capítulo é dividido em duas partes distintas, sendo que, na primeira, é apresentado um estudo sobre o estado da arte, onde será abordado/descrito eventuais *softwares* (programas informáticos) dedicados a este tipo de solução e na segunda é abordado o tema da otimização das respostas aos pedidos de cotação.

O terceiro capítulo está também repartido em duas partes, sendo que na primeira pretende-se aprofundar a temática e as ferramentas utilizadas na elaboração do simulador em si, enquanto na segunda parte é descrita a conceção da simulação e os seus *Inputs* e *Outputs*.

No quarto capítulo estão expressos os valores calculados referentes ao conjunto das diferentes variáveis através de ferramentas manuais e posterior comparação com os mesmos valores obtidos de acordo com o programa desenvolvido nesta dissertação.

No quinto capítulo é apresentado o modo de funcionamento do projeto desenvolvido, bem como uma pequena descrição para a utilização do simulador “manual de utilização”, onde estão indicadas as informações necessárias para um dado utilizador instruído possa fazer uso da aplicação.

As conclusões são expressas no sexto capítulo, assim como as possibilidades para desenvolvimentos futuros.

Por fim é apresentada a bibliografia utilizada, seguida dos respetivos apêndices.

Na figura 1.1 está esquematizada a organização da dissertação.

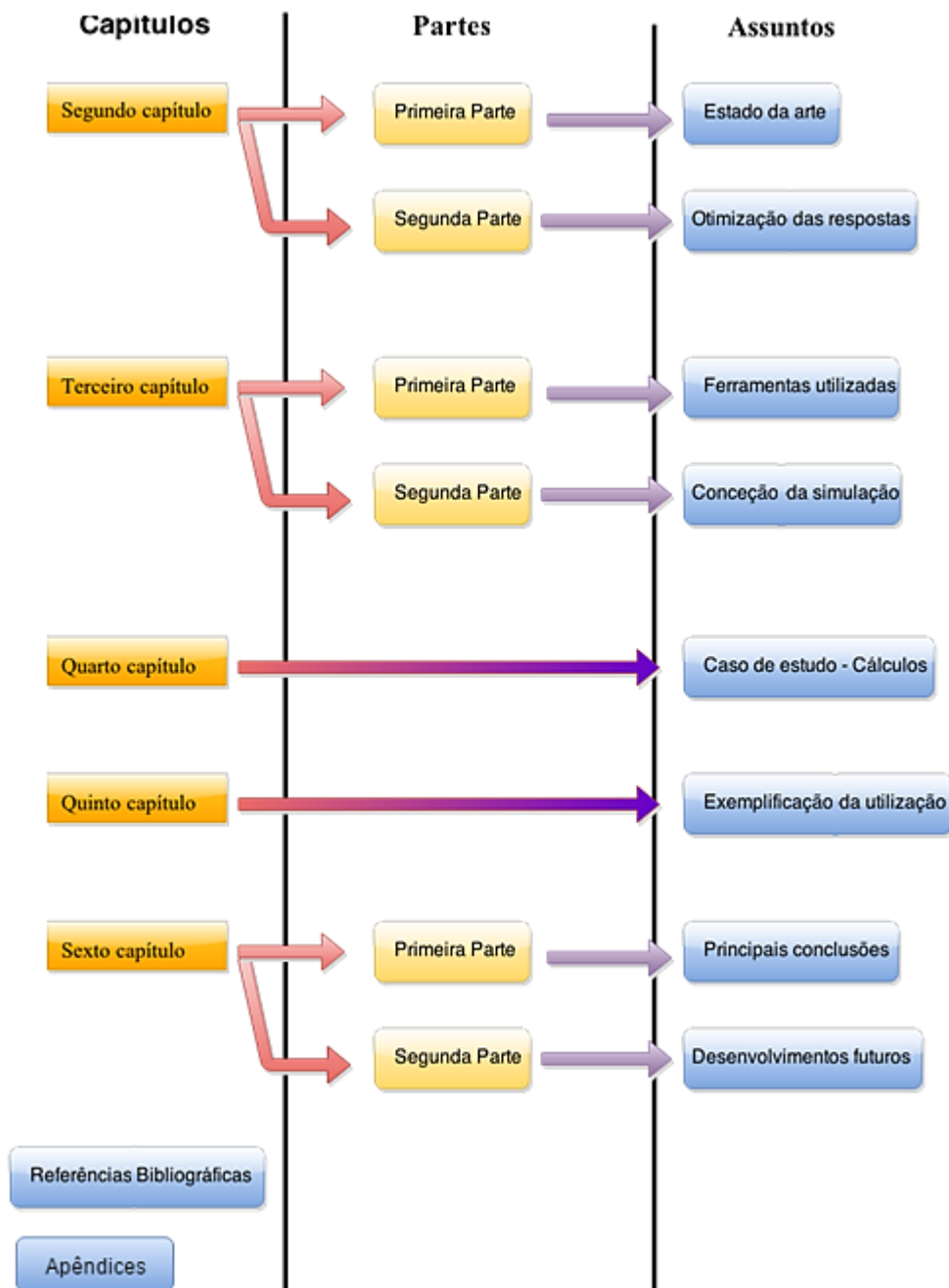


Figura 1.1 - Organização da dissertação

1.5- Notação

As figuras, tabelas e expressões são apresentadas com referência ao capítulo a que pertencem e são numeradas de ordem sequencial no respectivo capítulo. A identificação de expressões é apresentada entre parênteses curvos (), e a identificação de referências bibliográficas é apresentada entre parêntesis retos [], de acordo com a bibliografia consultada.

Expressões em língua estrangeira ou siglas são apresentadas em itálico.

As referências bibliográficas estão elaboradas de acordo com a normalização IEEE 2006 e respectiva atualização de 2010.

CAPÍTULO

2

Estado da arte e otimização das respostas

O 2º capítulo está repartido em duas partes, sendo a primeira referente ao estudo sobre o estado da arte, onde serão abordados/descritos softwares dedicados a este tipo de solução e a segunda referente ao tema da otimização das respostas aos pedidos de cotação.

2.1 – Estado da arte

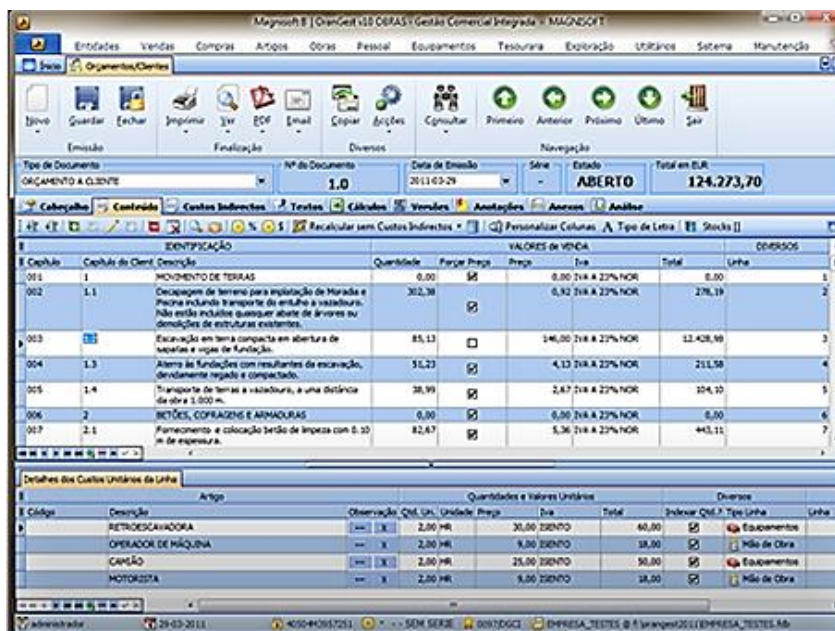
Apresenta-se neste ponto o estudo sobre o estado da arte, onde serão abordados/descritos *softwares* dedicados ao tipo de solução desejada, bem como o tema da otimização das respostas aos pedidos de cotação e analisadas as suas principais características.

Serão ainda abordados os aspetos positivos e negativos a ter em conta para a realização do simulador descrito no decorrer desta dissertação.

2.1.1 – *Software OranGest Obras*

O *software OranGest Obras* (OGO) apresentado nas figuras 2.1 e 2.2 é uma aplicação direcionada para as empresas relacionadas com a construção civil e áreas de negócio adjacentes [1]. Este mesmo *software* consiste numa ferramenta forte para o cálculo e gestão da orçamentação de obra.

A solução referida possibilita a realização e o acompanhamento rigoroso da orçamentação de obra.



IDENTIFICAÇÃO		VALORES DE VENDA				COSTOS		
Capítulo	Capítulo de Client	Descrição	Quantidade	Parâmetro Preço	Preço	Taxa	Total	Unidade
001	1	MOVIMENTO DE TERRAS	0,00	95	0,00	21% HOR	0,00	1
002	1.1	Desaparelhagem de terreno para implantação de Moradia e Piscina incluindo transporte de entulho e resíduos. Não estão incluídos quaisquer abates de árvores ou demolições de estruturas existentes.	302,38	95	0,92	21% HOR	276,19	2
003	1.2	Escavação em terra compacta em abertura de sapatas e sapas de fundações.	85,13	95	146,00	21% HOR	13.428,99	3
004	1.3	Atorno às fundações com resultantes da escavação, devidamente rejado e compactado.	51,23	95	4,13	21% HOR	211,58	4
005	1.4	Transporte de terras e rochas, a uma distância de obra > 2.000 m.	38,99	95	2,67	21% HOR	104,10	5
006	2	REDES, COFRAJES E ARMADURAS	0,00	95	0,00	21% HOR	0,00	6
007	2.1	Fornecimento e colocação betão de limpeza com 0,10 m de espessura.	82,67	95	5,36	21% HOR	443,11	7

Detalhes dos Custos Unidade de Linha		Quantidades e valores Unitários			Diversos						
Í	Código	Descrição	Observações	Qtd. Un.	Unidade	Preço	Taxa	Total	Incluir Qtd.?	Tip. Unid.	Unidade
1		RETROSCAVADORA		2,00	HR	30,00	200%	60,00	95	Equipamentos	
2		OPERADOR DE MÁQUINA		2,00	HR	9,00	200%	18,00	95	Hão de Obra	
3		CAMião		2,00	HR	25,00	200%	50,00	95	Equipamentos	
4		MOTOCICLETA		2,00	HR	9,00	200%	18,00	95	Hão de Obra	

Figura 2.1 - Janela *software OranGest Obras* (vista geral)

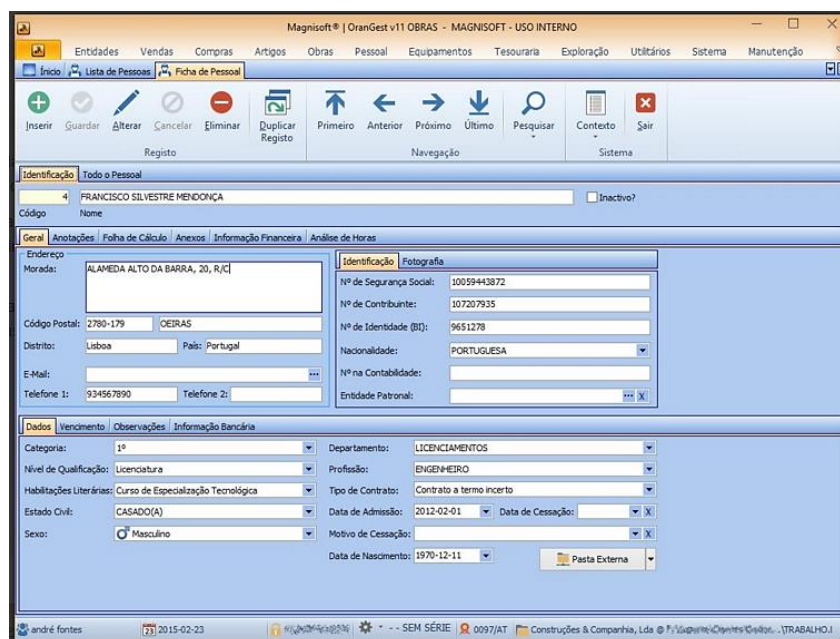


Figura 2.2 - Janela software OranGest Obras (ficha pessoal)

2.1.1.1 – Aspetos positivos do simulador

O OGO é constituído por diversos módulos: Obras, Materiais, Pessoal, Equipamentos, Orçamentos e Autos de Medição. Estes estão totalmente integrados, resultando numa solução útil para o setor, que abrange várias vertentes administrativas e operacionais.

À semelhança de toda a linha *OranGest* a fiabilidade e performance estão asseguradas pela utilização do motor de dados *FireBird SQL Server*; Esta aplicação encontra-se disponível para *Windows* e *Linux* o que permite a livre escolha da plataforma usada como servidor.

2.1.1.2 – Aspetos negativos do simulador

A associação de custos e recursos de cada capítulo apesar de funcional é ao mesmo tempo um pouco confusa e difícil de manejar. Acresce ainda, que a importação de ficheiros, falha com

alguma regularidade, obrigando desta forma a executar um preenchimento manual de grandes quantidades de informação.

É importante salientar o elevado preço desta aplicação em que uma licença anual para uma máquina/PC custa 400,00€ (com limite de solicitações, sem atualizações do *software*) e uma licença para cinco máquinas/PCs 1.250,00€ (com a subscrição de um contrato de assistência técnica, sem limite de solicitações e com atualizações do *software*).

2.1.2 – *Software Quote Roller*

O *software Quote Roller* (QR) encontra-se ilustrado nas figuras 2.3 e 2.4. Este define-se como uma aplicação *Web*, que em parceria com a anterior, foi ponto de focagem na elaboração do simulador descrito nesta dissertação, uma vez que o *software* QR possui aspetos positivos de interesse, tais como o tempo de processamento e a fácil utilização [2].

Foco ainda para o facto deste tipo de *software* permite um aumento de produtividade da equipa orçamentista, bem como da equipa comercial.

Name/Description	Price	Qty	Tax	Subtotal
Sample Product	\$1.99	3	10%	\$6.57
Sample Product description				
WordPress Twitter extension at \$15.00				\$0.00
WordPress Facebook Login at \$20.00				
WordPress Portfolio extension at \$35.00				
Subtotal:				\$6.57

Name/Description	Price	Time	Qty	Tax	Subtotal
Sample Service	\$10.99	Month	2	None	\$21.98

Figura 2.3 - Janela *software Quote Roller* (vista geral)

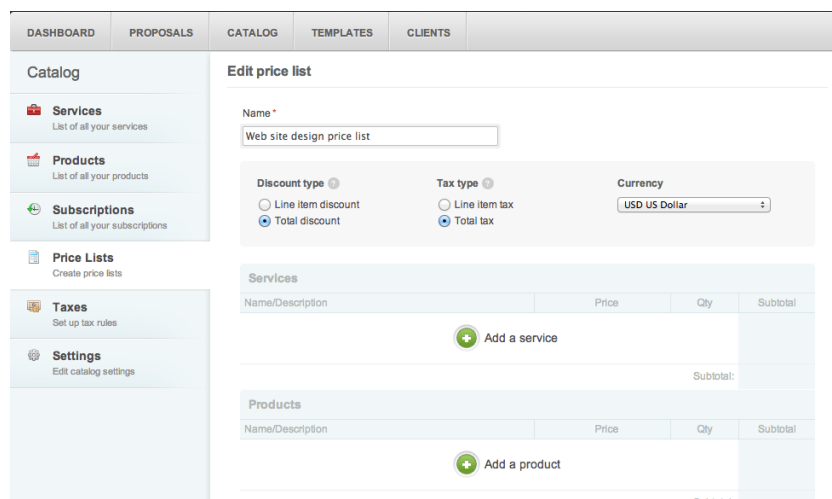


Figura 2.4 - Janela software Quote Roller (edit price list)

2.1.2.1 – Aspectos positivos do simulador

Este simulador possui uma plataforma gráfica e técnica para elaboração de propostas comerciais bastante “*User Friendly*”. Esta capacidade, possibilita que o *software* seja utilizado por um leque variado de pessoas sem as mesmas terem instrução para tal tarefa.

2.1.2.2 – Aspectos negativos do simulador

A aplicação referida apresenta cálculos deficitários, uma vez que só permite cálculos simples. Este tipo primitivo/simplista de cálculos não permite a elaboração de propostas internacionais. A juntar a este ponto negativo sublinha-se o valor monetário elevado para a aquisição das suas licenças, ou seja, a licença para utilização *free* (sem custo) apenas dura 15 dias e o valor das restantes variam entre 19€ e 49€ por mês.

2.1.3 – Comparação de *Softwares*

Para elaboração do simulador descrito nesta dissertação, foram tidos os vários aspectos positivos e negativos dos simuladores anteriormente referidos.

Tabela 2.1 - Comparação de Softwares (disponíveis mercado)

Classificações a comparar	Softwares	
	<i>OranGest Obras</i>	<i>Quote Roller</i>
User Friendly	Não	Sim (possibilita que o <i>software</i> seja utilizado por um leque variado de pessoas sem as mesmas terem instrução para tal tarefa)
Aspetto gráfico	Fraco	Bom
Funcionalidade	Bom	Médio
Conteúdos	Abrangente (constituído por diversos módulos: Obras, Materiais, Pessoal, Equipamentos, Orçamentos e Autos de Medição)	Exíguo
Tempo de processamento	Médio	Baixo
Variedade de plataformas (androide, Windows, ios, outros)	Alto	Baixo
Banco de dados	Alto	Alto
Licença de Software	Pagamento	Pagamento

2.2 – Otimização das respostas

2.2.1 – Introdução as PCO

As PCO são o modelo utilizado para concorrer a uma ou mais obras, sendo que estas se baseiam na elaboração de um orçamento, denominado orçamento de obra (ODO). A melhor forma de otimizar as respostas aos pedidos de cotação é otimizando o ODO, como tal pode-se evidenciar dois tipos de variáveis a otimizar: o fator tempo e o fator monetário [4], [5].

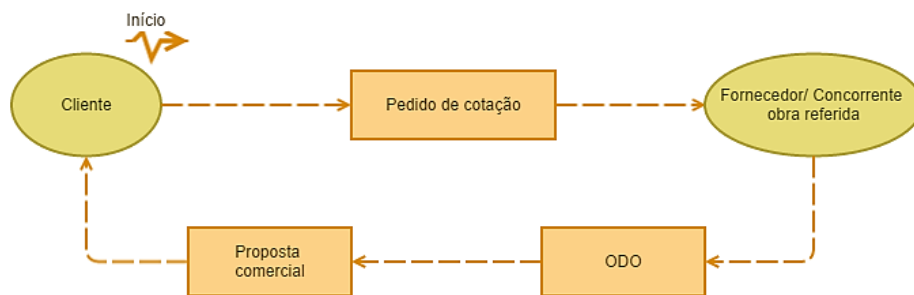


Figura 2.5 - Relação cliente fornecedor

2.2.1.1 – Fator tempo

Este fator possui três dimensões: oportunidade, capacidade de espera e sequência, [4].

Oportunidade: no sentido em que cada pedido de cotação é referente a uma obra única, sendo esta uma oportunidade de negócio. Pela minha experiência profissional posso afirmar que eram respondidos, em média, apenas um terço das propostas de cotação solicitadas. Neste sentido, pode-se assinalar a oportunidade como um dos elementos a otimizar.

Capacidade de espera: esta capacidade representa o tempo para elaboração da respetiva PCO, uma vez que cada pedido vem com um tempo máximo de resposta (quer positiva ou negativa). Em relação a este elemento, pode-se indicar como sendo a variável relacionada com o período compreendido entre a receção do pedido de cotação e a sua resposta.

Sequência: no sentido em que todas as propostas devem ser acompanhadas e executadas e em caso de necessidade elaboradas, posteriormente, as suas revisões/atualizações.

2.2.1.2 – Fator monetário

O fator monetário está diretamente relacionado com o valor monetário. Este valor é expresso numericamente e está associado a uma mercadoria/serviço. O conceito de preço é global, mas a sua unidade tem variantes em vários países/continentes.

Euro (€) é a moeda oficial de 18 dos países da União Europeia, da qual Portugal faz parte.

Dólar Americano (\$) é a moeda de referência para os negócios em diversos países, nomeadamente em Angola, devido a este facto o mesmo é utilizado para as PCO que têm como objetivo este mercado.

2.2.2 – Elementos otimizados no simulador desenvolvido

Com a realização do simulador, desenvolvido nesta dissertação destinado a orçamentação de obra, pretende-se otimizar com vantagem os seguintes aspetos:

Tempo:

- Elaboração dos cálculos através de ferramentas computacionais, evitando erros e diminuindo o tempo de resposta;
- Lista de equipamentos com tempos de referência para a sua montagem.

Monetário:

- Lista de equipamentos com preços de referência, oferecendo ao utilizador um preço médio de mercado;
- Cálculo da quantidade de pessoas necessárias para executar a obra no tempo pretendido automático;
- Facilidade de resposta aos pedidos de orçamento, criando oportunidade para elaborar mais PCO, aumentando o número de hipóteses para futuros negócios/parcerias.

CAPÍTULO

3

Ferramentas utilizadas na conceção do simulador

O 3º capítulo consiste na explicitação da base da própria simulação. Este capítulo aprofunda a temática e as ferramentas utilizadas na elaboração do simulador desenvolvido nesta dissertação.

3.1 – Ferramentas utilizadas

As ferramentas mais importantes utilizadas na elaboração deste simulador são o programa *Excel*, a linguagem *Java* e o *SQL* [6], [7] e [8] em que:

- 1 – O *Excel* é utilizado para a elaboração das tabelas de variáveis para teste aos valores.
- 2 – O programa é construído através de linguagem *Java*.
- 3 – O *SQL* surge na elaboração do simulador como base de dados para a produção dos orçamentos solicitados.

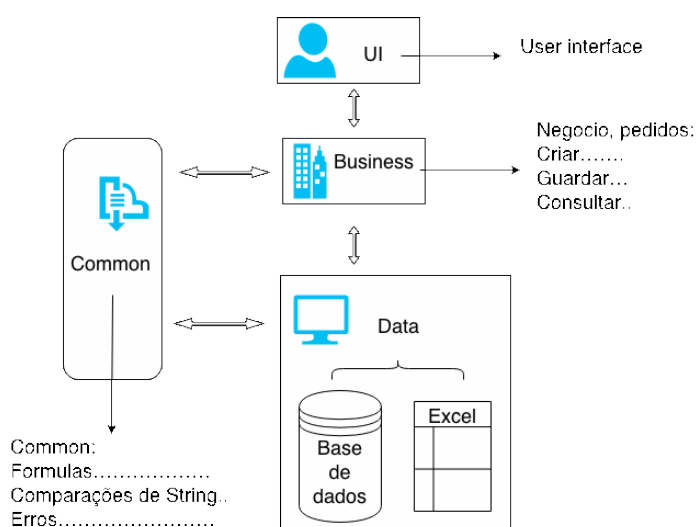


Figura 3.1 - Estrutura do simulador

3.1.1 – *Excel*

O elemento agora referido é o mais importante na realização do simulador desenvolvido, devidos as várias capacidades de cálculo e inúmeras interligações de variáveis.

Este elemento é fornecido pela equipa da *Microsoft* e consiste numa plataforma de cálculo avançada, que de uma forma geral pode ser definido como um *software* que permite criar tabelas, calcular e analisar dados.

Esta ferramenta é utilizada maioritariamente na elaboração das tabelas de variáveis para teste aos valores.

3.1.2 – Java

Java é uma linguagem de programação orientada a objetos/pacotes em que estes mesmos pacotes serão utilizados na elaboração do simulador proposto por esta dissertação.

O mesmo tem a seu cargo a tarefa de realização dos vários códigos de processamento conforme figura 3.2.

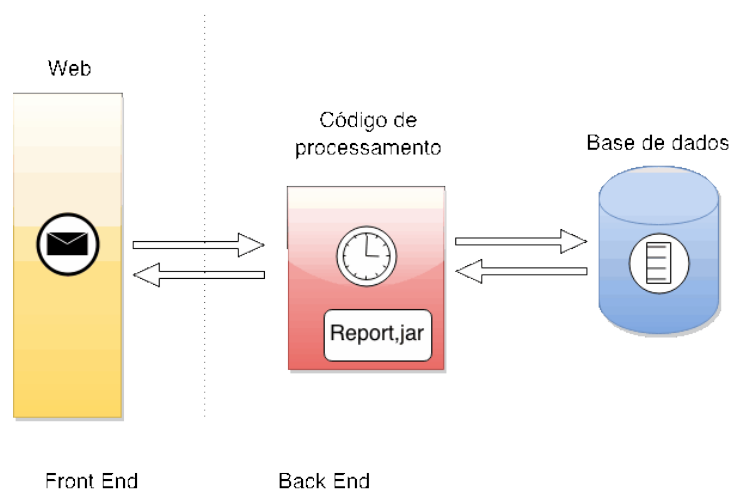


Figura 3.2 - Contexto da utilização Java

3.1.3 – SQL

SQL ou *Structured Query Language* é uma linguagem informática inspirada em ferramentas matemáticas que permite a criação de várias listas de tabelas de dados (linhas e colunas). Esta capacidade permite a criação da base de dados utilizada no simulador apresentado.

3.2 – Conceção da simulação

Na conceção da simulação são elaboradas/identificadas as diversas variáveis *Input* e *Ouput*, bem como a elaboração dos cálculos que as relacionam [9], [10] (figura 3.3).

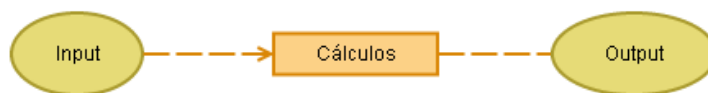


Figura 3.3 - Diagrama *Input/Output*

No conjunto de variáveis que se segue é importante referir que as variáveis relacionadas com o *Input* estão referenciados com a letra *i*, os cálculos e *Output* com a letra *A*.

Sublinha-se o facto da estrutura do simulador estar repartida em vários tópicos/bases: cada uma destas bases tem agregado um conjunto de *Input*, cálculos e *Output*.

Sendo que cada base aborda um tópico diferente (os diversos tópicos serão abordados posteriormente neste trabalho).

É essencial ressaltar que as bases comunicam entre si culminando, desta forma, para a elaboração do resultado final apresentado pelo simulador.

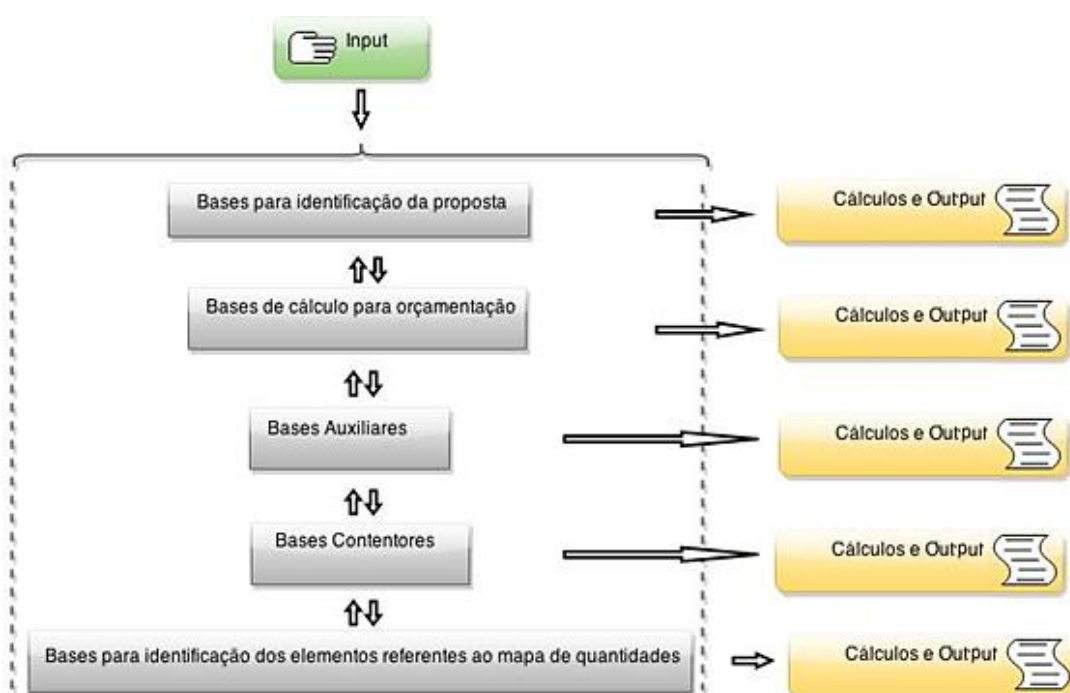


Figura 3.4 - Organização estrutural das variáveis (*Input/Output*)

Deverá ainda ser tido em consideração a ferramenta IntelliJ IDEA, pois foi nesta que as variadas linguagem de programação (*Java*, *SQL* e *HTML*) foram compiladas e conectadas [3], [11] e [12].

3.2.1 – Input

Input é o termo que denomina qualquer tipo de entrada ou mudanças que são inseridos num sistema como, por exemplo, os nomes, as datas ou outros e ainda ativa ou modifica um processo. É um conceito abstrato usado na exploração de modelagem/sistemas. Está habitualmente associado a outros termos como, por exemplo, o campo de entrada, variável de entrada e parâmetro de entrada.

Pode-se com isto afirmar que o *Input* define-se como as variáveis que são atribuídas pelo utilizador. Estas mesmas variáveis estão presentes na elaboração do relatório e respetivos cálculos [13], [14], [15] e [16].

3.2.1.1 – Bases para identificação da proposta

Neste ponto identificam-se os vários *Inputs* relacionados com as bases para identificação da proposta (tabela 3.1).

Tabela 3.1 - Bases para identificação (variáveis)

BASES PARA IDENTIFICAÇÃO DA PROPOSTA	
Nome Obra	i1
Localização	i2
	APLICAÇÃO DAS MARGENS
	MARGEM "A" ATÉ
	MARGEM "B" A PARTIR DE
	i3
Refª	i4
	Revisão
Ciente	i5
DATA	i6
A/CENÇª.	i7

Nas bases para identificação da proposta (tabela 3.1) tem-se:

- i1 – Nome da obra, conforme expresso na memória descritiva e justificativa (MDJ);
- i2 - Localização da obra, conforme o concelho ou província destinatária;
- i3 - Limite na qual a margem A se altera para a margem B (Nota: margem A e B serão referidas posteriormente nos elementos compreendidos entre i10 e i14);
- i4 - Referência interna da empresa concorrente à obra;
- i5 - Identificação do cliente destinatário da proposta;
- i6 - Data para a qual a proposta é entregue ao destinatário (solicitante);
- i7 - Nome da pessoa responsável pela receção da proposta (lado do cliente);
- i8 - Número de revisão da proposta.

A tabela 3.2 demonstra um exemplo do preenchimento da tabela bases para identificação da proposta.

Tabela 3.2 - Bases para identificação (exemplo)

BASES PARA IDENTIFICAÇÃO DA PROPOSTA			
Nome Obra	Maxi Viana – Remodelação da loja		
Localização	Angola		
	APLICAÇÃO DAS MARGENS		
	MARGEM "A" ATÉ		50 000,00 €
	MARGEM "B" A PARTIR DE		
Refª	Prop. 20150301	Revisão	R_00
Cliente	Teixeira Duarte		
DATA	7 de março de 2015		
A/C ENGº.	Eurico Clemente		

3.2.1.2 – Bases de cálculo para orçamentação

Conjuntamente com a tabela bases para identificação da proposta (tabela 3.1 e 3.2), a tabela 3.3 compreende outro conjunto de *Inputs* necessários para a realização da PCO.

Na tabela referida estão expressos os valores que vão influenciar o cálculo da orçamentação de forma dominante. É de referenciar que todos os valores apresentados deverão ser positivos, ou seja, superior a zero.

Tabela 3.3 - Bases de cálculo para orçamentação (variáveis)

BASES DE CÁLCULO PARA ORÇAMENTAÇÃO		
Câmbio USD/EURO		i9
Margem EMP. (Orig.)	A	i10
Margem EMP. (Orig.)	B	i11
Margem TA (Dist.)	A	i12
Margem TA (Dist.)	B	i13
Factor Produtividade		i14
Seguro		i15
Custo FCL 40*		i16
Custo FCL 20*		i17
Trasportes Internos (Dist.)		i18
Trasportes Internos (Orig.)		i19
Desalfandegamento		i20
Direitos		i21
Retenção IRT		i22
Imposto Selo		i23
Custo Hora/Homem - 1		i24
Custo Hora/Homem - 2		i25
Custo Hora/Homem - 3		i26
OUTROS CUSTOS		i44

Na tabela das bases de cálculo para orçamentação (tabela 3.3) tem-se:

- i9 - Identificação do câmbio/relação entre as moedas referência de dois tipos distintos, particularmente, neste exemplo, será utilizada a relação do câmbio entre Angola e Portugal \$/€;
- i10 - Margem de venda em Portugal (margem definida para elementos com valor inferior ao referido em i3);
- i11 - Margem de venda em Portugal (margem definida para elementos com valor superior ao referido em i3);
- i12 - Margem de venda no país de destino (margem definida para elementos com valor inferior ao referido em i3);
- i13 - Margem de venda no país de destino (margem definida para elementos com valor superior ao referido em i3);
- i14 - Definição por meio de percentagem da produtividade local;
- i15 - Valor do seguro (em percentagem);

- i16 - Valor de custo de um contentor de 40" (40 pés);
- i17 - Valor de custo de um contentor de 20" (20 pés);
- i18 - Custo com transporte de cada contentor no país de destino;
- i19 - Custo com transporte dos contentores em Portugal;
- i20 - Custos relacionados com o desalfandegamento (em percentagem);
- i21 - Custos relacionados com os direitos (em percentagem);
- i22 - Custos relacionados com a retenção IRT (em percentagem);
- i23 - Custos relacionados com o imposto selo (em percentagem);
- i24/ i25/ i26 - Definição de um dos três valores possíveis para o custo de mão-de-obra à hora (com referência Hora/Homem- 1, Hora/Homem- 2 ou Hora/Homem- 3);
- i44 - Valor gasto relacionado com outros custos, como elementos exteriores ao fornecimento/realização dos trabalhos (em percentagem) (somente nas propostas com países de destino e de origem diferentes).

A tabela 3.4 exemplifica o preenchimento das bases de cálculo para orçamentação:

Tabela 3.4 - Bases de cálculo para orçamentação (exemplo)

BASES DE CÁLCULO PARA ORÇAMENTAÇÃO		
Câmbio USD/EURO		1,40
Margem EMP.(Orig.)	A	15,00%
Margem EMP.(Orig.)	B	15,00%
Margem TA(Dist.)	A	25,00%
Margem TA(Dist.)	B	25,00%
Factor Produtividade		1,00
Seguro		1,50%
Custo FCL 40"		4 000,00 €
Custo FCL 20"		2 650,00
Trasportes Internos (Dist.)		\$ 1 725,00
Trasportes Internos (Orig.)		0,00
Desalfandegamento		10,50%
Direitos		0,00%
Retenção IRT		3,50%
Imposto Selo		1,00%
Custo Hora/Homem - 1		30,00 €
Custo Hora/Homem - 2		30,00 €
Custo Hora/Homem - 3		30,00 €
OUTROS CUSTOS		0,00%

3.2.1.3 – Bases Auxiliares

As bases auxiliares consistem nos elementos necessários para a elaboração do relatório e cálculos de orçamentação referentes ao tempo e equipas necessários para a conclusão da obra/trabalho mencionada (tabela 3.5).

Tabela 3.5 - Bases auxiliares (variáveis)

BASES AUXILIARES				
	Meses	Dias trabalháveis por mês	Duração/Dias	
Prazo de Execução Obra	i27	i28	A1	
	Horas	Dias	Homens	Equipa (1 Of+1 Aj)
AEROLICA	A2	A3	A4	A5
HIDRAULICA	A6	A7	A8	A9
ELECTRICIDADE	A10	A11	A12	A13
Total Obra	A14			
Dias	A15			
Homens	A16			
Equipa (1 Of+1 Aj)	A17			

Nesta tabela (tabela 3.5) tem-se:

i27 - Número de meses atribuído (pelo cliente) para a execução da obra;

i28 - Número de dias úteis/trabalháveis por mês.

A tabela 3.6 apresenta um exemplo do preenchimento das bases auxiliares:

Tabela 3.6 - Bases auxiliares (exemplo)

BASES AUXILIARES				
	Meses	Dias trabalháveis por mês	Duração/Dias	
Prazo de Execução Obra	3	20	60	
	Horas	Dias	Homens	Equipa (1 Of+1 Aj)
AEROLICA	833,33	104,1666667	2	1
HIDRAULICA	25,00	3,125	0	0
ELECTRICIDADE	2,50	0,3125	0	0
Total Obra	860,83			
Dias	108			
Homens	2			
Equipa (1 Of+1 Aj)	1			

3.2.1.4 – Bases Contentores

As bases contentores são os *Inputs* relacionados com os contentores para transportes marítimos (figuras 3.5 e figura 3.6).



Figura 3.5 - Características referências para contentores 20”

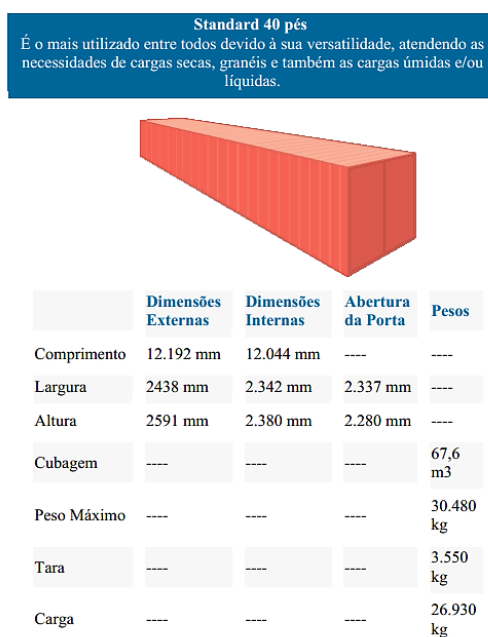


Figura 3.6 - Características referências para contentores 40”

Tabela 3.7 - Bases contentores (variáveis)

BASES CONTENTORES			
Quantidades contentores		VALOR FOB	A18
i29	Q. FCL 20"	SEGURO	A19
i30	Q. FCL 40"	FRETE	A20
		TRANS. INTERNO	A21
		VALOR CIF	A22
		% FRETE	A23
		% T. Interno Dist.	A24

Na tabela das bases contentores (tabela 3.7) tem-se:

i29 - Número de um contentor de 20" utilizados;

i30 - Número de um contentor de 40" utilizados.

A tabela 3.8 apresenta um exemplo do preenchimento das bases contentores:

Tabela 3.8 - Bases contentores (exemplo)

BASES CONTENTORES			
Quantidades contentores		VALOR FOB	\$ 47 518,15
2,00	Q. FCL 20"	SEGURO	\$ 3,49
1,00	Q. FCL 40"	FRETE	\$ 14 973,00
		TRANS. INTERNO	\$ 5 175,00
		VALOR CIF	\$ 67 669,64
		% FRETE	31,51%
		% T. Interno Dist.	10,89%

3.2.1.5 – Bases para identificação dos elementos referentes ao mapa de quantidades

Uma das variáveis mais importantes para a identificação dos elementos é o seu tipo e subtipo. A utilização destes dois parâmetros surgem como uma ajuda à criação do mapa de quantidades, pois desta forma, é possível agrupar os elementos pelo seu tipo de utilização ou características, elaborando e, deste modo, elaborar uma “árvore” de elementos [17].

Os valores para estas variáveis estão identificados na tabela 3.9.

Tabela 3.9 - Valores possíveis para as variáveis i31 e i32

Tipo (i31)	Subtipo (i32)
	Network
	Power
	Supply
	UPS
Electrical	Generator
Telecom	Productive
Security	Source
Waters	Specific
Sewage	Lighting
Building	Cables
Air	Sound
Mechanics	Panels
Other	Pipe
	Duct
	Fire
	CCTV

Após a identificação dos elementos é necessário listá-los. Para tal será criado um *Array* de elementos.

Devido ao tamanho da folha de cálculo, referente ao *Array* expresso nas figuras 3.7 e 3.8, a mesma está repartida em várias divisórias, sendo que cada uma das linhas horizontais, dentro das bases para identificação dos elementos representa um *Array* distinto.

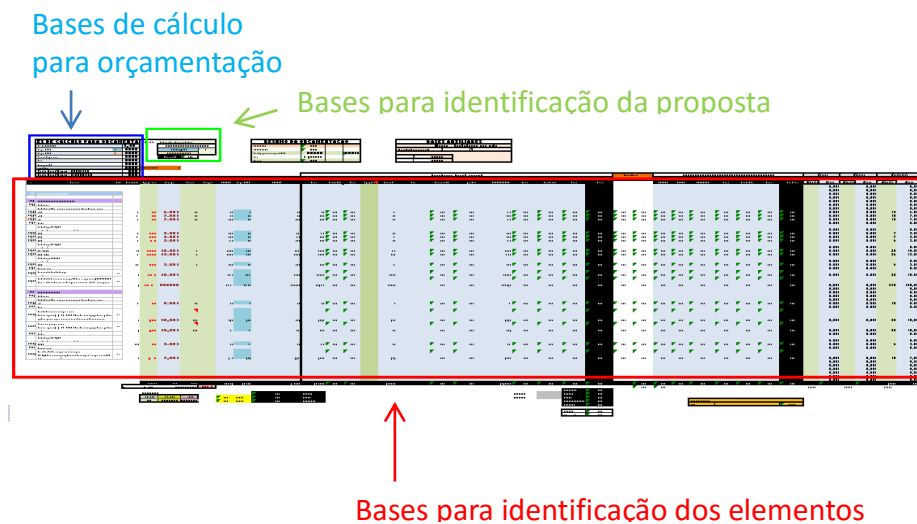


Figura 3.7 - Extensão da folha de cálculo

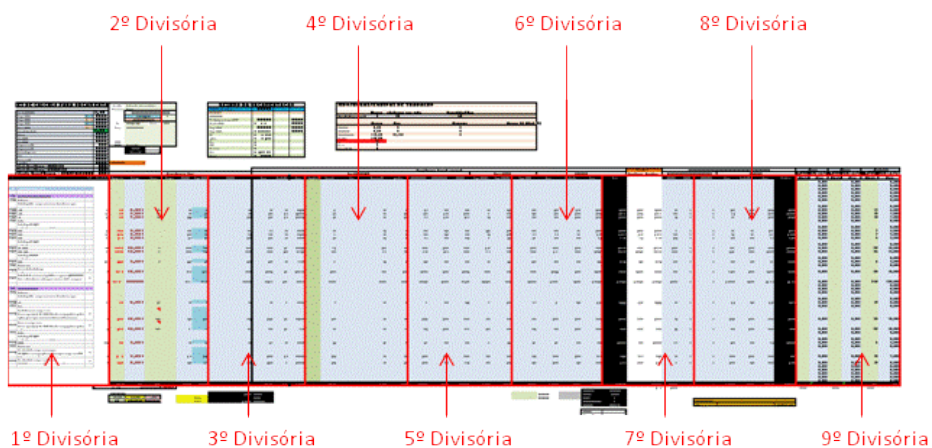


Figura 3.8 - Divisórias da folha de cálculo

Conforme já referido anteriormente as bases para identificação dos elementos encontram-se repartidas em várias divisórias distintas. Estas divisórias apresentam-se nas tabelas seguintes (tabela 3.10 a tabela 3.12) [18].

Tabelas 3.10 - Bases para identificação dos elementos (variáveis)

Tabela 3.10.1 – 1ª Divisória

Item	Descrição	Unid	Quantidades
i33	i34	i35	i36
i33	i34	i35	i36
i33	i34	i35	i36
i33	i34	i35	i36
i33	i34	i35	i36
i33	i34	i35	i36
i33	i34	i35	i36
i33	i34	i35	i36

Tabela 3.10.2 – 2ª Divisória

Valores Unitários em Euros (€)					
Equipamento	Montagem	Acessórios	Transportes	PCEMP.(Orig.)	Margem EMP.(Orig.)
i37	A27	i38	i39	A28	A29
i37	A27	i38	i39	A28	A29
i37	A27	i38	i39	A28	A29
i37	A27	i38	i39	A28	A29
i37	A27	i38	i39	A28	A29
i37	A27	i38	i39	A28	A29
i37	A27	i38	i39	A28	A29
A60	A61	A62	A63	A64	A65

Tabela 3.10.3 – 3ª Divisória

Valores Unitários em Euros (€)		Valores Unitários em Dolares Americanos (\$)		
PV EMP.(Orig.)		Custos EMP. Dist. (USD)		
		Materiais	Mão de Cora	Unitário
A30	A30	A31	A32	A33
A30	A30	A31	A32	A33
A30	A30	A31	A32	A33
A30	A30	A31	A32	A33
A30	A30	A31	A32	A33
A30	A30	A31	A32	A33
A30	A30	A31	A32	A33
A30	A30	A31	A32	A33
A66	A66	A67	A68	A69

Tabela 3.10.4 – 4ª Divisória

Valores Unitários em Dolares Americanos (\$)				
Custos EMP. Dist. (USD)				
Equip Locais	Desalfand.	Direitos	Retenção IRT	Imposto Selo
i40	A34	A35	A36	A37
i40	A34	A35	A36	A37
i40	A34	A35	A36	A37
i40	A34	A35	A36	A37
i40	A34	A35	A36	A37
i40	A34	A35	A36	A37
i40	A34	A35	A36	A37
i40	A34	A35	A36	A37
A70	A71	A72	A73	A74

Tabela 3.10.5 – 5ª Divisória

Valores Unitários em Dolares Americanos (\$)			
Custos EMP. Dist. (USD)			Lucro EMP. Dist.
PCMATERIAIS (Dist.)	PCMÃO OBRA (Dist.)	PC EMP.(Dist.)	Margem EMP.(Dist.) Mat.
A38	A39	A40	A41
A38	A39	A40	A41
A38	A39	A40	A41
A38	A39	A40	A41
A38	A39	A40	A41
A38	A39	A40	A41
A38	A39	A40	A41
A38	A39	A40	A41
A38	A39	A40	A41
A75	A76	A77	A78

Tabela 3.10.6 – 6ª Divisória

Valores Unitários em Dolares Americanos (\$)			
Lucro EMP. Dist.		PV EMP. Dist.	
Margem EMP.(Dist.) MO	Margem EMP.(Dist.)	Materiais	Mão de obra
A42	A43	A44	A45
A42	A43	A44	A45
A42	A43	A44	A45
A42	A43	A44	A45
A42	A43	A44	A45
A42	A43	A44	A45
A42	A43	A44	A45
A42	A43	A44	A45
A79	A80	A81	A82

Tabela 3.10.7 – 7ª Divisória

Valores Unitários em Dolares Americanos		Outros Custos		PREÇO VENDA EMP. Dist. CO	
PV EMP. Dist.		Valor Unitário	Valor Final	DESPESAS COM TRANSPORT	
Unitário	Parcial			SEGURO	FRETE
A46	A47	A48	A49	A50	A51
A46	A47	A48	A49	A50	A51
A46	A47	A48	A49	A50	A51
A46	A47	A48	A49	A50	A51
A46	A47	A48	A49	A50	A51
A46	A47	A48	A49	A50	A51
A46	A47	A48	A49	A50	A51
A46	A47	A48	A49	A50	A51
A83	A84	A85	A86	A87	A88

Tabela 3.10.8 – 8ª Divisória

M DESPESAS DE TRANSPORTE (\$)				
E	PV EMP. Dist.			
T. INTERNOS	Materiais	Mão de Obra	Unitário	Total Parcial
A52	A53	A54	A55	A56
A52	A53	A54	A55	A56
A52	A53	A54	A55	A56
A52	A53	A54	A55	A56
A52	A53	A54	A55	A56
A52	A53	A54	A55	A56
A52	A53	A54	A55	A56
A52	A53	A54	A55	A56
A89	A90	A91	A92	A93

Tabela 3.10.9 – 9ª Divisória

Custo Hora/Homem -1		Custo Hora/Homem -2		Custo Hora/Homem -3	
Tempos	Custo MO Unitário	Tempos	Custo MO Unitário	Tempos	Custo MO Unitário
Minutos	Euros	Minutos	Euros	Minutos	Euros
i41	A57	i42	A58	i43	A59
i41	A57	i42	A58	i43	A59
i41	A57	i42	A58	i43	A59
i41	A57	i42	A58	i43	A59
i41	A57	i42	A58	i43	A59
i41	A57	i42	A58	i43	A59
i41	A57	i42	A58	i43	A59
i41	A57	i42	A58	i43	A59
A94	A95	A96	A97	A98	A99

Nas tabelas anteriores (tabelas 3.10.1 a 3.10.9) tem-se:

i33 - Código numérico endereço (ID) a um elemento específico;

i34 - Descrição ortográfica de um elemento específico;

i35 - Identificação do tipo de unidade, em que os valores estão expressos (tabela 3.11);

Tabela 3.11 - Valores possíveis para as variáveis i35

i35	Identificação
Uh.	Unidade
Vg.	Conjunto
m	Metro
O	Outros

i36 - Identificação da quantidade de elementos referidos;

i37 - Valor unitário do equipamento;

- i38 - Valor unitário dos acessórios (referentes ao equipamento identificado na variável anterior);
- i39 - Valor referente ao transporte do equipamento;
- i40 - Valor monetário referente aos equipamento locais;
- i41 - Tempo de montagem do equipamento em minutos (tendo como referência o custo de Hora/Homem -1);
- i42 - Tempo de montagem do equipamento em minutos (tendo como referência o custo de Hora/Homem -2);
- i43 - Tempo de montagem do equipamento em minutos (tendo como referência o custo de Hora/Homem -3).
- i45 - Definição das condições de venda.

Nota: O facto destas variáveis pertencerem a um *Array* significa que as variáveis referidas serão multiplicadas em quantidade pelo número de elementos. Querendo com isto dizer que para cada elemento haverá um i31, i32, i33,...até ao i43. Esta multiplicação encontra-se especialmente visível na programação, uma vez que nesta estão identificadas as variáveis como lista (conjunto de elementos).

Nas tabelas 3.12.1 a 3.12.9 são apresentados os exemplos de preenchimento das bases para identificação dos elementos referentes às divisórias em causa.

Tabela 3.12 - Bases para identificação dos elementos (exemplo)

Tabela 3.12.1 – 1ª Divisória

Item	Descrição	Unid	Quantidades
	Building		
	Lighting		
1	d 25	m	50
2	d 32	m	100
3	d 40	m	10

Tabela 3.12.2 – 2ª Divisória

Valores Unitários em Euros (€)					
Equipamento	Montagem	Acessórios	Transportes	PC EMP.(Orig.)	Margem EMP.(Orig.)
0,35	500,00 €	0,22		0,57	0,09
0,53	7,50 €	0,33		0,86	0,13
0,67	7,50 €	0,44		1,11	0,17
77,20	25 825,00	48,40	0,00	125,60	18,84

Tabela 3.12.3 – 3ª Divisória

Valores Unitários em Euros (€)		Valores Unitários em Dolares Americanos (\$)		
PV EMP.(Orig.)		Custos EMP. Dist. (USD)		
		Materiais	Mão de Obra	Unitário
	0,66	0,92	700,00	700,9177
	0,99	1,38	10,50	11,8846
	1,28	1,79	10,50	12,2871
	144,44	202,22	36 155,00	36 357,22

Tabela 3.12.4 – 4ª Divisória

Valores Unitários em Dolares Americanos (\$)				
Custos EMP. Dist. (USD)				
Equip Locais	Desalfand.	Direitos	Retenção IRT	Imposto Selo
	0,10	0,00	24,53	7,01
	0,15	0,00	0,42	0,12
	0,19	0,00	0,43	0,12
0,00	21,23	0,00	1 272,50	363,57

Tabela 3.12.5 – 5ª Divisória

Valores Unitários em Dolares Americanos (\$)			
Custos EMP. Dist. (USD)			Lucro EMP. Dist.
PC MATERIAIS (Dist.)	PC MÃO OBRA (Dist.)	PC EMP.(Dist.)	Margem EMP.(Dist.) Mat.
1,06	731,50	732,56	0,26
1,59	10,97	12,56	0,40
2,06	10,97	13,03	0,51
232,55	37 781,98	38 014,52	58,14

Tabela 3.12.6 – 6ª Divisória

Valores Unitários em Dolares Americanos (\$)			
Lucro EMP. Dist.		PV EMP. Dist.	
Margem EMP.(Dist.) MO	Margem EMP.(Dist.)	Materiais	Mão de obra
182,88	183,14	1,32	914,38
2,74	3,14	1,99	13,72
2,74	3,26	2,57	13,72
9 445,49	9 503,63	290,69	47 227,47

Tabela 3.12.7 – 7ª Divisória

Valores Unitários em Dolares Americanos		Outros Custos		PREÇO VENDA EMP. Dist. C	
PV EMP. Dist.		Valor Unitário	Valor Final	DESPESAS COM TRANSPORTE	
Unitário	Parcial			SEGURO	FRETE
915,69	45 784,71	915,69	45 784,71	0,79	\$14 426,79
15,71	1 570,60	15,71	1 570,60	2,39	\$494,90
16,28	162,85	16,28	162,85	0,31	\$51,31
47 518,15	47 518,15		47 518,15	\$3,49	\$14 973,00

Tabela 3.12.8 – 8ª Divisória

PREÇO VENDA EMP. Dist. COM DESPESAS DE TRANSPORTE (\$)				
	PV EMP. Dist.			
T. INTERNOS	Materiais	Mão de Obra	Unitário	Total Parcial
\$4 986,22	389,60	914,38	1 303,97	65 198,51
\$171,05	8,67	13,72	22,39	2 238,93
\$17,73	9,50	13,72	23,22	232,20
\$5 175,00	20 442,17	47 227,47	67 669,64	\$67 669,64

Tabela 3.12.9 – 9ª Divisória

Custo Hora/Homem -1		Custo Hora/Homem -2		Custo Hora/Homem -3	
Tempos	Custo MO Unitário	Tempos	Custo MO Unitário	Tempos	Custo MO Unitário
Minutos	Euros	Minutos	Euros	Minutos	Euros
1000	500,00 €	15	7,50 €	15	7,50 €
833,33	25 000,00	25,00	750,00	2,50	75,00

3.2.2 – Cálculos e *Output*

Os cálculos têm um papel crucial, uma vez que são estes que permitem relacionar as entradas *Inputs*, com as saídas, *Outputs*.

Output é o termo que denomina qualquer tipo de saída referente a um sistema como, por exemplo, os nomes, as datas ou outros. É um conceito abstrato usado na exploração de modelagem/sistemas. Está habitualmente associado a outros termos, como, por exemplo, o campo de saída, variável de saída e parâmetro de saída.

Neste sentido, o *output* define-se como o conjunto de informações alusivas "a saída" de um sistema (organismo, mecanismo ou digital). Estas mesmas variáveis estão presentes na elaboração do relatório e respetivos cálculos.

3.2.2.1 – Bases Auxiliares

As bases auxiliares, como já referido anteriormente consistem na tabela onde estão expressos os valores relacionados com a mão-de-obra (tabela 3.5).

Os vários cálculos elaborados ao longo desta tabela encontram-se abaixo discriminados:

A1 - Número total de dias atribuído (pelo cliente) para a execução da obra;

$$A1 = i27 \times i28 \text{ [dias]}$$

Equação 1 – Cálculo A1

i27 - Definição do número de meses atribuído (pelo cliente) para a execução da obra;

i28 - Definição do número de dias úteis/trabalháveis por mês.

A2 - Número de horas total relacionada com o custo hora/homem -1;

$$A2 = A94 \times i14 \text{ [horas]}$$

Equação 2 - Cálculo A2

i14 - Definição por meio de percentagem da produtividade local;

A94 - Somatório temporal (em horas) relativo à mão-de-obra, do tipo 1;

A3 - Número de dias totais relacionada com o custo hora/homem -1;

$$A3 = A2 \div 8 \text{ [dias]}$$

Equação 3 - Cálculo A3

A2 - Número de horas total relacionada com o custo hora/homem -1;

A4 - Número de homens necessário, para realizar os trabalhos relacionados com o custo hora/homem -1;

$$A4 = A3 \div A1 \text{ [número de homens]}$$

Equação 4 - Cálculo A4

A1 - Número total de dias atribuído (pelo cliente) para a execução da obra;

A3 - Número de dias totais relacionada com o custo hora/homem -1;

A5 - Número de equipas compostas por um oficial mais um ajudante (custo hora/homem -1);

$$A5 = A4 \div 2 \text{ [número de equipas]}$$

Equação 5 - Cálculo A5

A4 - Número de homens necessário, para realizar os trabalhos relacionados com o custo hora/homem -1;

A6 - Número de horas total relacionada com o custo hora/homem -2;

$$A6 = A96 \times i14 \text{ [horas]}$$

Equação 6 - Cálculo A6

i14 - Definição por meio de percentagem da produtividade local;

A96 - Somatório temporal (em horas) relativo à mão-de-obra, do tipo 2;

A7 - Número de dias total relacionada com o custo hora/homem -2;

$$A7 = A6 \div 8 \text{ [dias]}$$

Equação 7 - Cálculo A7

A6 - Número de horas total relacionada com o custo hora/homem -2;

A8 - Número de homens necessários, para realizar os trabalhos relacionados com o custo hora/homem -2;

$$A8 = A7 \div A1 \text{ [número de homens]}$$

Equação 8 - Cálculo A8

A1 - Número total de dias atribuído (pelo cliente) para a execução da obra;

A7 - Número de dias total relacionada com o custo hora/homem -2;

A9 - Número de equipas compostas por um oficial mais um ajudante (custo hora/homem -2);

$$A9 = A8 \div 2 \text{ [número de equipas]}$$

Equação 9 - Cálculo A9

A8 - Número de homens necessários, para realizar os trabalhos relacionados com o custo hora/homem -2;

A10 - Número de horas total relacionada com o custo hora/homem -3;

$$A10 = A98 \times i14 \text{ [horas]}$$

Equação 10 - Cálculo A10

i14 - Definição por meio de percentagem da produtividade local;

A98 - Somatório temporal (em horas) relativo à mão-de-obra, do tipo 3;

A11 - Número de dias total relacionada com o custo hora/homem -3;

$$A11 = A10 \div 8 \text{ [dias]}$$

Equação 11 - Cálculo A11

A10 - Número de horas total relacionada com o custo hora/homem -3;

A12 - Número de homens necessários, para realizar os trabalhos relacionados com o custo hora/homem -3;

$$A12 = A11 \div A1 \text{ [número de homens]}$$

Equação 12 - Cálculo A12

A1 - Número total de dias atribuído (pelo cliente) para a execução da obra;

A11 - Número de dias total relacionada com o custo hora/homem -3;

A13 - Número de equipas compostas por um oficial mais um ajudante (custo hora/homem -3);

$$A13 = A12 \div 2 \text{ [número de equipas]}$$

Equação 13 - Cálculo A13

A12 - Número de homens necessários, para realizar os trabalhos relacionados com o custo hora/homem -3;

A14 - Número total de horas gasto na execução da obra;

$$A14 = A2 + A6 + A10 \text{ [horas]}$$

Equação 14 - Cálculo A14

A2 - Número de horas total relacionada com o custo hora/homem -1;

A6 - Número de horas total relacionada com o custo hora/homem -2;

A10 - Número de horas total relacionada com o custo hora/homem -3;

A15 - Número total de dias gasto na realização da empreitada;

$$A15 = A14 \div 8 \text{ [dias]}$$

Equação 15 - Cálculo A15

A14 - Número total de horas gasto na execução da obra;

A16 - Número total de homens necessários, para realizar os trabalhos;

$$A16 = A15 \div A1 \text{ [número de homens]}$$

Equação 16 - Cálculo A16

A1 - Número total de dias atribuído (pelo cliente) para a execução da obra;

A15 - Número total de dias gasto na realização da empreitada;

A17 - Número de equipas totais compostas por um oficial mais um ajudante;

$$A17 = A16 \div 2 \text{ [número de equipas]}$$

Equação 17 - Cálculo A17

A16 - Número total de homens necessários, para realizar os trabalhos;

3.2.2.2 – Bases Contentores

As bases contentores, conforme já referido anteriormente é a tabela onde estão expressos os valores relacionados com transporte e aluguer dos contentores marítimos (tabela 3.7).

Nesta tabela tem-se:

A18 - Somatório dos custos finais dos equipamentos no país de destino;

$$A18 = A84 \text{ [€]}$$

Equação 18 - Cálculo A18

A84 - Somatório dos preços de venda totais dos vários equipamentos, no país de destino;

A19 - Custo derivado ao seguro dos contentores;

$$A19 = [A66 \times i15 \times (1 + (i10 \text{ ou } i11))] \times i9; \text{ Dependendo da relação entre o A66 e o } i3 \text{ [€]}$$

Equação 19 - Cálculo A19

Se $A66 < i3$, valor a assumir na fórmula $i10$ (margem menor)

Se $A66 > i3$, valor a assumir na fórmula $i11$ (margem maior)

$i3$ - Descrição do limite na qual a margem A se altera para a margem B;

$i9$ - Identificação do câmbio/relação entre as moedas referência de dois tipos distintos;

$i10$ - Identificação da margem de venda em Portugal (margem definida para elementos com valor inferior ao referido em $i3$);

$i11$ - Identificação da margem de venda em Portugal (margem definida para elementos com valor superior ao referido em $i3$);

$i15$ - Definição do valor do seguro (em percentagem);

A66 - Somatório dos preços de venda, referente aos equipamentos no país de envio;

A20 - custo derivado ao frete marítimo;

$$A20 = [i29 \times (i17 \times i9) + i30(i16 \times i9)] + [i29 \times (i17 \times i9) + i30(i16 \times i9)] \times (i10 \text{ ou } i11); \text{ Dependendo da relação entre o } i3 \text{ e } (i29 \times i17 + i30 \times i16) \text{ [€]}$$

Equação 20 - Cálculo A20

Se $i29 \times i17 + i30 \times i16 < i3$, valor a assumir na fórmula $i10$ (margem menor)

Se $i29 \times i17 + i30 \times i16 > i3$, valor a assumir na fórmula $i11$ (margem maior)

- i3 - Descrição do limite na qual a margem A se altera para a margem B;
- i9 - Identificação do câmbio/relação entre as moedas referência de dois tipos distintos;
- i10 - Identificação da margem de venda em Portugal (margem definida para elementos com valor inferior ao referido em i3);
- i11 - Identificação da margem de venda em Portugal (margem definida para elementos com valor superior ao referido em i3);
- i16 - Representação do valor de custo de um contentor de 40" (40 pés);
- i17 - Representação do valor de custo de um contentor de 20" (20 pés);
- i29 - Representação do número de um contentor de 20" utilizados;
- i30 - Representação do número de um contentor de 40" utilizados.

A21 - Custo relacionado com o transporte interno no país de destino;

$$A21 = (i29 + i30) \times i18 \text{ [€]}$$

Equação 21 - Cálculo A21

- i18 - Identificação do custo com transporte de cada contentor no país de destino;
- i29 - Representação do número de um contentor de 20" utilizados;
- i30 - Representação do número de um contentor de 40" utilizados.

A22 - Somatório dos elementos referenciados como bases contentores;

$$A22 = A18 + A19 + A20 + A21 \text{ [€]}$$

Equação 22 - Cálculo A22

- A18 - Somatório dos custos finais dos equipamentos no país de destino;
- A19 - Custo derivado ao seguro dos contentores;
- A20 - custo derivado ao frete marítimo;
- A21 - Custo relacionado com o transporte interno no país de destino;

A23 - Valor do custo relacionado com o frete marítimo (em percentagem);

$$A23 = A20 \div A84 \text{ [€]}$$

Equação 23 - Cálculo A23

A20 - custo derivado ao frete marítimo;

A84 - Somatório dos preços de venda totais dos vários equipamentos, no país de destino;

A24 - Valor referente ao custo relacionado com o transporte interno no país de destino (em percentagem);

$$A24 = A21 \div A84 \text{ [€]}$$

Equação 24 - Cálculo A24

A21- Custo relacionado com o transporte interno no país de destino;

A84 - Somatório dos preços de venda totais dos vários equipamentos, no país de destino;

3.2.2.3 – Bases para identificação dos elementos referentes ao mapa de quantidades

Conforme o próprio nome sugere nas bases para identificação dos elementos, é apresentada uma descrição dos vários elementos, bem como os cálculos por detrás dos mesmos (tabela 3.10).

A27 - Preço da montagem, referente a um equipamento específico (unitário);

$$A27 = A57 + A58 + A59 \text{ [€]}$$

Equação 27 - Cálculo A27

A57 - Preço de custo relativo à mão-de-obra, do tipo 1;

A58 - Preço de custo relativo à mão-de-obra, do tipo 2;

A59 - Preço de custo relativo à mão-de-obra, do tipo 3;

A28 - Preço de custo, referente a um equipamento específico (unitário);

$$A28 = i37 + i38 + i39 \text{ [€]}$$

Equação 28 - Cálculo A28

i37 - Definição do valor unitário do equipamento;

i38 - Definição do valor unitário dos acessórios (referentes ao equipamento identificado na variável anterior);

i39 - Definição do valor referente ao transporte do equipamento;

A29 - Margem no equipamento referente ao país de envio (unitária);

$$A29 = A28 \times (i10 \text{ ou } i11) \quad ; \text{ Dependendo da relação entre o } i37 \text{ e o } i3 \text{ [€]}$$

Equação 29 - Cálculo A29

i3 - Descrição do limite na qual a margem A se altera para a margem B;

i10 - Identificação da margem de venda em Portugal (margem definida para elementos com valor inferior ao referido em i3);

i11 - Identificação da margem de venda em Portugal (margem definida para elementos com valor superior ao referido em i3);

A28 - Preço de custo, referente a um equipamento específico (unitário);

i37 - Definição do valor unitário do equipamento;

A30 - Preço de venda, referente a um equipamento específico no país de envio (unitário);

$$A30 = A28 + A29 \text{ [€]}$$

Equação 30 - Cálculo A30

A28 - Preço de custo, referente a um equipamento específico (unitário);

A29 - Margem no equipamento referente ao país de envio (unitária);

A31 - Custo do material físico, no país de destino (unitário);

$$A31 = A30 \times i9 \text{ [€]}$$

Equação 31 - Cálculo A31

i9 - Identificação do câmbio/relação entre as moedas referência de dois tipos distintos,

A30 - Preço de venda, referente a um equipamento específico no país de envio (unitário);

A32 - Custo relacionado com a mão-de-obra, no país de destino (unitário);

$$A32 = A27 \times i14 \times i9 \text{ [€]}$$

Equação 32 - Cálculo A32

i9 - Identificação do câmbio/relação entre as moedas referência de dois tipos distintos,

i14 - Definição por meio de percentagem da produtividade local;

A27 - Preço da montagem, referente a um equipamento específico (unitário);

A33 - Custo referente ao equipamento, no país de destino (unitário);

$$A33 = A31 + A32 \text{ [€]}$$

Equação 33 - Cálculo A33

A31 - Custo do material físico, no país de destino (unitário);

A32 - Custo relacionado com a mão-de-obra, no país de destino (unitário);

A34 - Valor de custo destinado ao desalfandegamento (unitário);

$$A34 = A31 \times i20 \text{ [€]}$$

Equação 34 - Cálculo A34

i20 - Definição dos custo relacionados com o desalfandegamento (em percentagem);

A31 - Custo do material físico, no país de destino (unitário);

A35 - Valor destinado aos direitos (unitário);

$$A35 = A31 \times i21 \text{ [€]}$$

Equação 35 - Cálculo A35

i21 - Identificação dos custo relacionados com os direitos (em percentagem);

A31 - Custo do material físico, no país de destino (unitário);

A36 - Valor destinado a retenção IRT (unitário);

$$A36 = A33 \times i22 \text{ [€]}$$

Equação 36 - Cálculo A36

i22 - Identificação dos custo relacionados com a retenção IRT (em percentagem);

A33 - Custo referente ao equipamento, no país de destino (unitário);

A37 - Valor destinado ao imposto selo (unitário);

$$A37 = A33 \times i23 \text{ [€]}$$

Equação 37 - Cálculo A37

i23 - Identificação dos custo relacionados com o imposto selo (em percentagem);

A33 - Custo referente ao equipamento, no país de destino (unitário);

A38 - Custo físico total, referente a um equipamento específico no país de destino (unitário);

$$A38 = (A31 + i40 + A34 + A35) + (A31 \times i22) + (A31 \times i23) \text{ [€]}$$

Equação 38 - Cálculo A38

i22 - Identificação dos custo relacionados com a retenção IRT (em percentagem);

i23 - Identificação dos custo relacionados com o imposto selo (em percentagem);

A31 - Custo do material físico, no país de destino (unitário);

A34 - Valor de custo destinado ao desalfandegamento (unitário);

A35 - Valor destinado aos direitos (unitário);

i40 - Identificação do valor monetário referente aos equipamento locais;

A39 - Custo alusivo a mão-de-obra total, referente a um equipamento específico no país de destino (unitário);

$$A39 = A32 + (A32 \times i22) + (A32 \times i23) \text{ [€]}$$

Equação 39 - Cálculo A39

i22 - Identificação dos custo relacionados com a retenção IRT (em percentagem);

i23 - Identificação dos custo relacionados com o imposto selo (em percentagem);

A32 - Custo relacionado com a mão-de-obra, no país de destino (unitário);

A40 - Custo total, referente a um equipamento específico no país de destino (unitário);

$$A40 = A38 + A39 \text{ [€]}$$

Equação 40 - Cálculo A40

A38 - Custo físico total, referente a um equipamento específico no país de destino (unitário);

39 - Custo alusivo a mão-de-obra total, referente a um equipamento específico no país de destino (unitário);

A41 - Margem no equipamento físico referente ao país de destino (unitária);

$$A41 = A38 \times (i12 \text{ ou } i13) \text{ ; Dependendo da relação entre o } i37 \text{ e o } i3 \text{ [€]}$$

Equação 41 - Cálculo A41

i3 - Descrição do limite na qual a margem A se altera para a margem B;

i12 - Representação da margem de venda no país de destino (margem definida para elementos com valor inferior ao referido em i3);

i13 - Identificação da margem de venda no país de destino (margem definida para elementos com valor superior ao referido em i3);

i37 - Definição do valor unitário do equipamento;

A38 - Custo físico total, referente a um equipamento específico no país de destino (unitário);

A42 - Margem relacionada com a mão-de-obra, referente ao país de destino (unitário);

$$A42 = A39 \times (i12 \text{ ou } i13) \text{ ; Dependendo da relação entre o } i37 \text{ e o } i3 \text{ [€]}$$

Equação 42 - Cálculo A42

i3 - Descrição do limite na qual a margem A se altera para a margem B;

i12 - Representação da margem de venda no país de destino (margem definida para elementos com valor inferior ao referido em i3);

i13 - Identificação da margem de venda no país de destino (margem definida para elementos com valor superior ao referido em i3);

i37 - Definição do valor unitário do equipamento;

A39 - Custo alusivo a mão-de-obra total, referente a um equipamento específico no país de destino (unitário);

A43 - Margem total no equipamento referente ao país de destino (unitária);

$$A43 = A40 \times (i12 \text{ ou } i13); \text{ Dependendo da relação entre o } i37 \text{ e o } i3 \text{ [€]}$$

Equação 43 - Cálculo A43

i3 - Descrição do limite na qual a margem A se altera para a margem B;

i12 - Representação da margem de venda no país de destino (margem definida para elementos com valor inferior ao referido em i3);

i13 - Identificação da margem de venda no país de destino (margem definida para elementos com valor superior ao referido em i3);

i37 - Definição do valor unitário do equipamento;

A40 - Custo total, referente a um equipamento específico no país de destino (unitário);

A44 - Preço de venda dos materiais, no país de destino (unitário);

$$A44 = A38 + A41 \text{ [€]}$$

Equação 44 - Cálculo A44

A38 - Custo físico total, referente a um equipamento específico no país de destino (unitário);

A41 - Margem no equipamento físico referente ao país de destino (unitária);

A45 - Preço de venda referente a mão-de-obra, no país de destino (unitário);

$$A45 = A39 + A42 \text{ [€]}$$

Equação 45 - Cálculo A45

A39 - Custo alusivo a mão-de-obra total, referente a um equipamento específico no país de destino (unitário);

A42 - Margem relacionada com a mão-de-obra, referente ao país de destino (unitário);

A46 - Preço de venda total do equipamento, no país de destino (unitário);

$$A46 = A40 + A43 \text{ [€]}$$

Equação 46 - Cálculo A46

A40 - Custo total, referente a um equipamento específico no país de destino (unitário);

A43 - Margem total no equipamento referente ao país de destino (unitária);

A47 - Preço de venda total dos vários equipamentos, referenciados como iguais e quantificados na mesma célula, no país de destino (parcial);

$$A47 = A46 \times i36 \text{ [€]}$$

Equação 47 - Cálculo A47

i36 - Identificação da quantidade de elementos referidos;

A46 - Preço de venda total do equipamento, no país de destino (unitário);

A48 - Valor gasto relacionado com outros custos, como elementos exteriores ao fornecimento/realização dos trabalhos (unitário);

$$A48 = A46 \times i44 \text{ [€]}$$

Equação 48 - Cálculo A48

A46 - Preço de venda total do equipamento, no país de destino (unitário);

i44 - Identificação do valor gasto relacionado com outros custos, como elementos exteriores ao fornecimento/realização dos trabalhos (em percentagem);

A49 - Valor gasto relacionado com outros custos, como elementos exteriores ao fornecimento/realização dos trabalhos (parcial);

$$A49 = A48 \times i36 \text{ [€]}$$

Equação 49 - Cálculo A49

i36 - Identificação da quantidade de elementos referidos;

A48 - Valor gasto relacionado com outros custos, como elementos exteriores ao fornecimento/realização dos trabalhos (unitário);

A50 - Valor destinado as despesas com transporte (seguro);

$A50 = i36 \times A30 \times i15 \times [1 + (i10 \text{ ou } i11)] \times i9$; Dependendo da relação entre o $A66 \times i15$ e o $i3$ [€]

Equação 50 - Cálculo A50

i3 - Descrição do limite na qual a margem A se altera para a margem B;

i9 - Identificação do câmbio/relação entre as moedas referência de dois tipos distintos;

i10 - Identificação da margem de venda em Portugal (margem definida para elementos com valor inferior ao referido em i3);

i11 - Identificação da margem de venda em Portugal (margem definida para elementos com valor superior ao referido em i3);

i15 - Definição do valor do seguro (em percentagem);

A30 - Preço de venda, referente a um equipamento específico no país de envio (unitário);

i36 - Identificação da quantidade de elementos referidos;

A66 - Somatório dos preços de venda, referente aos equipamentos no país de envio;

A51 - Valor destinado as despesas com transporte (frete);

$$A51 = A47 \times (A20 \div A84) \text{ [€]}$$

Equação 51 - Cálculo A51

A20 - custo derivado ao frete marítimo;

A47 - Preço de venda total dos vários equipamentos, referenciados como iguais e quantificados na mesma célula, no país de destino (parcial);

A84 - Somatório dos preços de venda totais dos vários equipamentos, no país de destino;

A52 - Valor destinado as despesas com transporte interno, no país de destino;

$$A52 = A47 \times (A21 \div A84) [\text{€}]$$

Equação 52 - Cálculo A52

A21 - Custo relacionado com o transporte interno no país de destino;

A47 - Preço de venda total dos vários equipamentos, referenciados como iguais e quantificados na mesma célula, no país de destino (parcial);

A84 - Somatório dos preços de venda totais dos vários equipamentos, no país de destino;

A53 - Preço de venda dos materiais e do transporte, no país de destino (unitário);

$$A53 = A44 + (A50 + A51 + A52) \div i36 [\text{€}]$$

Equação 53 - Cálculo A53

i36 - Identificação da quantidade de elementos referidos;

A44 - Preço de venda dos materiais, no país de destino (unitário);

A50 - Valor destinado as despesas com transporte (seguro);

A51 - Valor destinado as despesas com transporte (frete);

A52 - Valor destinado as despesas com transporte interno, no país de destino;

A54 - Preço referente a mão-de-obra, no país de destino (unitário);

$$A54 = A45 [\text{€}]$$

Equação 54 - Cálculo A54

A45 - Preço de venda referente a mão-de-obra, no país de destino (unitário);

A55 - Preço de venda total e do transporte, no país de destino (unitário);

$$A55 = A53 + A54 \text{ [€]}$$

Equação 55 - Cálculo A55

A53 - Preço de venda dos materiais e do transporte, no país de destino (unitário);

A54 - Preço referente a mão-de-obra, no país de destino (unitário);

A56 - Preço de venda total e do transporte, no país de destino (parcial);

$$A56 = A47 + A50 + A51 + A52 \text{ [€]}$$

Equação 56 - Cálculo A56

A47 - Preço de venda total dos vários equipamentos, referenciados como iguais e quantificados na mesma célula, no país de destino (parcial);

A50 - Valor destinado as despesas com transporte (seguro);

A51 - Valor destinado as despesas com transporte (frete);

A52 - Valor destinado as despesas com transporte interno, no país de destino;

A57 - Preço de custo relativo à mão-de-obra, do tipo 1;

$$A57 = i41 \times (i24 \div 60) \text{ [€]}$$

Equação 57 - Cálculo A57

i24 - Definição de um dos três valores possíveis para o custo de mão-de-obra à hora;

i41 - Identificação do tempo de montagem do equipamento em minutos (tendo como referência o custo de Hora/Homem -1);

A58 - Preço de custo relativo à mão-de-obra, do tipo 2;

$$A58 = i42 \times (i25 \div 60) [\text{€}]$$

Equação 58 - Cálculo A58

i25 - Definição de um dos três valores possíveis para o custo de mão-de-obra à hora;

i42 - Identificação do tempo de montagem do equipamento em minutos (tendo como referência o custo de Hora/Homem -2);

A59 - Preço de custo relativo à mão-de-obra, do tipo 3;

$$A59 = i43 \times (i26 \div 60) [\text{€}]$$

Equação 59 - Cálculo A59

i26 - Definição de um dos três valores possíveis para o custo de mão-de-obra à hora;

i43 - Identificação do tempo de montagem do equipamento em minutos (tendo como referência o custo de Hora/Homem -3).

A60 - Somatório dos preços dos equipamentos descritos no mapa de quantidade;

$$A60 = \sum(i36 \times i37) [\text{€}]$$

Equação 60 - Cálculo A60

i36 - Identificação da quantidade de elementos referidos;

i37 - Definição do valor unitário do equipamento;

A61 - Somatório dos preços das montagens dos equipamentos descritos no mapa de quantidade;

$$A61 = \sum(i36 \times A27) [\text{€}]$$

Equação 61 - Cálculo A61

i36 - Identificação da quantidade de elementos referidos;

A27 - Preço da montagem, referente a um equipamento específico (unitário);

A62 - Somatório dos preços dos acessórios, necessários para o bom funcionamento dos equipamentos referidos;

$$A62 = \sum(i36 \times i38) [\text{€}]$$

Equação 62 - Cálculo A62

i36 - Identificação da quantidade de elementos referidos;

i38 - Definição do valor unitário dos acessórios (referentes ao equipamento identificado na variável anterior);

A63 - Valor destinado aos transportes;

$$A63 = \sum(i36 \times i39) [\text{€}]$$

Equação 63 - Cálculo A63

i36 - Identificação da quantidade de elementos referidos;

i39 - Definição do valor referente ao transporte do equipamento;

A64 - Somatório dos vários preços de custo, referente aos equipamentos;

$$A64 = \sum(i36 \times A28) [\text{€}]$$

Equação 64 - Cálculo A64

i36 - Identificação da quantidade de elementos referidos;

A28 - Preço de custo, referente a um equipamento específico (unitário);

A65 - Somatório das margens, referente ao país de envio;

$$A65 = \sum(i36 \times A29) [\text{€}]$$

Equação 65 - Cálculo A65

i36 - Identificação da quantidade de elementos referidos;

A29 - Margem no equipamento referente ao país de envio (unitária);

A66 - Somatório dos preços de venda, referente aos equipamentos no país de envio;

$$A66 = \sum(i36 \times A30) [\text{€}]$$

Equação 66 - Cálculo A66

i36 - Identificação da quantidade de elementos referidos;

A30 - Preço de venda, referente a um equipamento específico no país de envio (unitário);

A67 - Somatório alusivo ao custo dos materiais no país de destino;

$$A67 = \sum(i36 \times A31) [\text{€}]$$

Equação 67 - Cálculo A67

i36 - Identificação da quantidade de elementos referidos;

A31 - Custo do material físico, no país de destino (unitário);

A68 - Somatório do custo da mão-de-obra no país de destino;

$$A68 = \sum(i36 \times A32) [\text{€}]$$

Equação 68 - Cálculo A68

i36 - Identificação da quantidade de elementos referidos;

A32 - Custo relacionado com a mão-de-obra, no país de destino (unitário);

A69 - Somatório do custo dos equipamentos no país de destino;

$$A69 = \sum(i36 \times A33) [\text{€}]$$

Equação 69 - Cálculo A69

i36 - Identificação da quantidade de elementos referidos;

A33 - Custo referente ao equipamento, no país de destino (unitário);

A70 - Somatório do custo relacionado com os equipamentos locais, no país de destino;

$$A70 = \sum(i36 \times i40) [\text{€}]$$

Equação 70 - Cálculo A70

i36 - Identificação da quantidade de elementos referidos;

i40 - Identificação do valor monetário referente aos equipamento locais;

A71 - Somatório dos custos derivados ao desalfandegamento;

$$A71 = \sum(i36 \times A34) [\text{€}]$$

Equação 71 - Cálculo A71

i36 - Identificação da quantidade de elementos referidos;

A34 - Valor de custo destinado ao desalfandegamento (unitário);

A72 - Somatório dos custos derivados aos direitos;

$$A72 = \sum(i36 \times A35) [\text{€}]$$

Equação 72 - Cálculo A72

i36 - Identificação da quantidade de elementos referidos;

A35 - Valor destinado aos direitos (unitário);

A73 - Somatório dos custos derivados à retenção IRT;

$$A73 = \sum(i36 \times A36) [\text{€}]$$

Equação 73 - Cálculo A73

i36 - Identificação da quantidade de elementos referidos;

A36 - Valor destinado a retenção IRT (unitário);

A74 - Somatório dos custos oriundos do imposto de selo;

$$A74 = \sum(i36 \times A37) [\text{€}]$$

Equação 74 - Cálculo A74

i36 - Identificação da quantidade de elementos referidos;

A37 - Valor destinado ao imposto selo (unitário);

A75 - Somatório dos custos dos vários equipamentos referidos, referente ao país de destino;

$$A75 = \sum(i36 \times A38) [\text{€}]$$

Equação 75 - Cálculo A75

i36 - Identificação da quantidade de elementos referidos;

A38 - Custo físico total, referente a um equipamento específico no país de destino (unitário);

A76 - Somatório dos custos, referentes à mão-de-obra dos vários equipamentos referidos, no país de destino;

$$A76 = \sum(i36 \times A39) [\text{€}]$$

Equação 76 - Cálculo A76

i36 - Identificação da quantidade de elementos referidos;

A39 - Custo alusivo a mão-de-obra total, referente a um equipamento específico no país de destino (unitário);

A77 - Somatório dos preços de custo, no país de destino,

$$A77 = \sum(i36 \times A40) [\text{€}]$$

Equação 77 - Cálculo A77

i36 - Identificação da quantidade de elementos referidos;

A40 - Custo total, referente a um equipamento específico no país de destino (unitário);

A78 - Somatório das margens alusivas aos materiais no país de destino;

$$A78 = \sum(i36 \times A41) [\text{€}]$$

Equação 78 - Cálculo A78

i36 - Identificação da quantidade de elementos referidos;

A41 - Margem no equipamento físico referente ao país de destino (unitária);

A79 - Somatório das margens alusivas à mão-de-obra no país de destino;

$$A79 = \sum(i36 \times A42) [\text{€}]$$

Equação 79 - Cálculo A79

i36 - Identificação da quantidade de elementos referidos;

A42 - Margem relacionada com a mão-de-obra, referente ao país de destino (unitário);

A80 - Somatório das margens, no país de destino;

$$A80 = \sum(i36 \times A43) [\text{€}]$$

Equação 80 - Cálculo A80

i36 - Identificação da quantidade de elementos referidos;

A43 - Margem total no equipamento referente ao país de destino (unitária);

A81 - Somatório dos preços de venda dos materiais;

$$A81 = \sum(i36 \times A44) [\text{€}]$$

Equação 81 - Cálculo A81

i36 - Identificação da quantidade de elementos referidos;

A44 - Preço de venda dos materiais, no país de destino (unitário);

A82 - Somatório dos preços de venda referente à mão-de-obra;

$$A82 = \sum(i36 \times A45) [\text{€}]$$

Equação 82 - Cálculo A82

i36 - Identificação da quantidade de elementos referidos;

A45 - Preço de venda referente a mão-de-obra, no país de destino (unitário);

A83 - Somatório dos preços de venda dos trabalhos a realizar, referente ao país de destino;

$$A83 = \sum(i36 \times A46) [\text{€}]$$

Equação 83 - Cálculo A83

i36 - Identificação da quantidade de elementos referidos;

A46 - Preço de venda total do equipamento, no país de destino (unitário);

A84 - Somatório dos preços de venda totais dos vários equipamentos, no país de destino;

$$A84 = \sum A47 \text{ [€]}$$

Equação 84 - Cálculo A84

A47 - Preço de venda total dos vários equipamentos, referenciados como iguais e quantificados na mesma célula, no país de destino (parcial);

A85 - Somatório dos valores gastos relacionados com outros custos, como por exemplo, elementos exteriores ao fornecimento/realização dos trabalhos (unitário);

$$A85 = \sum A48 \text{ [€]}$$

Equação 85 - Cálculo A85

A48 - Valor gasto relacionado com outros custos, como elementos exteriores ao fornecimento/realização dos trabalhos (unitário);

A86 - Somatório dos valores gastos relacionados com outros custos, como por exemplo, elementos exteriores ao fornecimento/realização dos trabalhos (parcial);

$$A86 = \sum A49 \text{ [€]}$$

Equação 86 - Cálculo A86

A49 - Valor gasto relacionado com outros custos, como elementos exteriores ao fornecimento/realização dos trabalhos (parcial);

A87 - Somatório dos valores destinado às despesas com transporte (seguro);

$$A87 = \sum A50 \text{ [€]}$$

Equação 87 - Cálculo A87

A50 - Valor destinado as despesas com transporte (seguro);

A88 - Somatório dos valores destinado às despesas com transporte (frete);

$$A88 = \sum A51 \text{ [€]}$$

Equação 88 - Cálculo A88

A51 - Valor destinado as despesas com transporte (frete);

A89 - Somatório dos valores destinado às despesas com transporte interno, no país de destino;

$$A89 = \sum A52 \text{ [€]}$$

Equação 89 - Cálculo A89

A52 - Valor destinado as despesas com transporte interno, no país de destino;

A90 - Somatório dos preços de venda dos materiais e do transporte, no país de destino (unitários);

$$A90 = \sum (i36 \times A53) \text{ [€]}$$

Equação 90 - Cálculo A90

i36 - Identificação da quantidade de elementos referidos;

A53 - Preço de venda dos materiais e do transporte, no país de destino (unitário);

A91 - Somatório dos preços referentes à mão-de-obra e transporte, no país de destino (unitários);

$$A91 = \sum(i36 \times A54) [\text{€}]$$

Equação 91 - Cálculo A91

i36 - Identificação da quantidade de elementos referidos;

A54 - Preço referente a mão-de-obra e transporte, no país de destino (unitário);

A92 - Somatório dos preços de venda totais e do transporte, no país de destino (unitários);

$$A92 = \sum(i36 \times A55) [\text{€}]$$

Equação 92 - Cálculo A92

i36 - Identificação da quantidade de elementos referidos;

A55 - Preço de venda total e do transporte, no país de destino (unitário);

A93 - Somatório dos preços de venda total e do transporte, no país de destino (parciais);

$$A93 = \sum A56 [\text{€}]$$

Equação 93 - Cálculo A93

A56 - Preço de venda total e do transporte, no país de destino (parcial);

A94 - Somatório temporal (em horas) relativo à mão-de-obra, do tipo 1;

$$A94 = [\sum(i36 \times i41)] \div 60 [\text{horas}]$$

Equação 94 - Cálculo A94

i36 - Identificação da quantidade de elementos referidos;

i41 - Identificação do tempo de montagem do equipamento em minutos (tendo como referência o custo de Hora/Homem -1);

A95 - Somatório dos preços de custo relativo à mão-de-obra, do tipo 1;

$$A95 = \sum(i36 \times A57) [\text{€}]$$

Equação 95 - Cálculo A95

i36 - Identificação da quantidade de elementos referidos;

A57 - Preço de custo relativo à mão-de-obra, do tipo 1;

A96 - Somatório temporal (em horas) relativo à mão-de-obra, do tipo 2;

$$A96 = [\sum(i36 \times i42)] \div 60 [\text{horas}]$$

Equação 96 - Cálculo A96

i36 - Identificação da quantidade de elementos referidos;

i42 - Identificação do tempo de montagem do equipamento em minutos (tendo como referência o custo de Hora/Homem -2);

A97 - Somatório dos preços de custo relativo à mão-de-obra, do tipo 2;

$$A97 = \sum(i36 \times A58) [\text{€}]$$

Equação 97 - Cálculo A97

i36 - Identificação da quantidade de elementos referidos;

A58 - Preço de custo relativo à mão-de-obra, do tipo 2;

A98 - Somatório temporal (em horas) relativo à mão-de-obra, do tipo 3;

$$A98 = [\sum(i36 \times i43)] \div 60 \text{ [horas]}$$

Equação 98 - Cálculo A98

i36 - Identificação da quantidade de elementos referidos;

i43 - Identificação do tempo de montagem do equipamento em minutos (tendo como referência o custo de Hora/Homem -3).

A99 - Somatório dos preços de custo relativo à mão-de-obra, do tipo 3;

$$A99 = \sum(i36 \times A59) \text{ [€]}$$

Equação 99 - Cálculo A99

i36 - Identificação da quantidade de elementos referidos;

A59 - Preço de custo relativo à mão-de-obra, do tipo 3;

3.2.2.3 – Bases para resumo da orçamentação

A tabela 3.13 permite ao utilizador do simulador retirar conclusões da sua própria orçamentação. Para isso, a tabela apresenta as variáveis mais importantes de forma a possibilitar ao utilizador uma fácil leitura de todo o processo.

Tabela 3.13 - Resumo da orçamentação (variáveis)

RESUMO DA ORÇAMENTAÇÃO			
PV EMP.(Orig.) (Inc. Frete e Seguro)	A100		
PV CIF EMP.(Dist.)	A101		
PV FOB EMP.(Dist.)	A102		
Custo(Equip., acc, e transportes) EMP.(Orig.)	A103		A114
Custo Mont. EMP.(Dist.)	A104		A115
Margem EMP.(Orig.)	A105		A116
Margem EMP.(Dist.)	A106		A117
IRT	A107		
Imp. Selo	A108		
Direitos	A109		
FCL	A110		
Frete Interno	A111		
Desalf.	A112		
Seguros	A113		

As variáveis da tabela 3.13 têm o seguinte significado:

A100 - Preço de venda total da empresa no país de origem;

$A100 = A66 + [A66 \times i15 \times (1 + (i10 \text{ ou } i11))] + (A20 \div i9)$; Dependendo da relação entre o $A66 \times i15$ e o $i3$ [€]

Equação 100 - Cálculo A100

i3 - Descrição do limite na qual a margem A se altera para a margem B;

i9 - Identificação do câmbio/relação entre as moedas referência de dois tipos distintos;

i10 - Identificação da margem de venda em Portugal (margem definida para elementos com valor inferior ao referido em i3);

i11 - Identificação da margem de venda em Portugal (margem definida para elementos com valor superior ao referido em i3);

i15 - Definição do valor do seguro (em percentagem);

A20 - custo derivado ao frete marítimo;

A66 - Somatório dos preços de venda, referente aos equipamentos no país de envio;

A101 - Preço de venda total CIF, da empresa no país de destino;

$$A101 = A22 \text{ [€]}$$

Equação 101 - Cálculo A101

A22 - Somatório dos elementos referenciados como bases contentores;

A102 - Preço de venda total FOB, da empresa no país de destino;

$$A102 = A18 \text{ [€]}$$

Equação 102 - Cálculo A102

A18 - Somatório dos custos finais dos equipamentos no país de destino;

A103 - Custo dos vários equipamentos no país de origem;

$$A103 = A64 \text{ [€]}$$

Equação 103 - Cálculo A103

A64 - Somatório dos vários preços de custo, referente aos equipamentos;

A104 - Custo de montagem (mão-de-obra) dos vários equipamentos no país de destino;

$$A104 = A61 \times i9 \times i14 \text{ [€]}$$

Equação 104 - Cálculo A104

i9 - Identificação do câmbio/relação entre as moedas referência de dois tipos distintos;

i14 - Definição por meio de percentagem da produtividade local;

A61 - Somatório dos preços das montagens dos equipamentos descritos no mapa de quantidade;

A105 - Margem da empresa no país de origem;

$$A105 = A65 \text{ [€]}$$

Equação 105 - Cálculo A105

A65 - Somatório das margens, referente ao país de envio;

A106 - Margem da empresa no país de destino;

$$A106 = A80 \text{ [€]}$$

Equação 106 - Cálculo A106

A80 - Somatório das margens, no país de destino;

A107 - Valor destinado à retenção IRT;

$$A107 = A73 \text{ [€]}$$

Equação 107 - Cálculo A107

A73 - Somatório dos custos derivados à retenção IRT;

A108 - Valor destinado a retenção do imposto de selo;

$$A108 = A74 \text{ [€]}$$

Equação 108 - Cálculo A108

A74 - Somatório dos custos oriundos do imposto de selo;

A109 - Valor destinado ao diretos;

$$A109 = A72 \text{ [€]}$$

Equação 109 - Cálculo A109

A72 - Somatório dos custos derivados aos diretos;

A110 - Preço dos contentores, referente ao país de destino;

$$A110 = (i29 \times i17 + i30 \times i16) \times i9 \text{ [€]}$$

Equação 110 - Cálculo A110

i9 - Identificação do câmbio/relação entre as moedas referência de dois tipos distintos;

i16 - Representação do valor de custo de um contentor de 40" (40 pés);

i17 - Representação do valor de custo de um contentor de 20" (20 pés);

i29 - Representação do número de um contentor de 20" utilizados;

i30 - Representação do número de um contentor de 40" utilizados.

A111 - Valor destinado ao transporte interno (contentores) no país de destino;

$$A111 = (i29 + i30) \times i18 \text{ [€]}$$

Equação 111 - Cálculo A111

i18 - Identificação do custo com transporte de cada contentor no país de destino;

i29 - Representação do número de um contentor de 20" utilizados;

i30 - Representação do número de um contentor de 40" utilizados.

A112 - Valor destinado ao desalfandegamento;

$$A112 = A71 \text{ [€]}$$

Equação 112 - Cálculo A112

A71 - Somatório dos custos derivados ao desalfandegamento;

A113 - Valor destinado ao seguro transposto no país de destino;

$$A113 = A19 \text{ [€]}$$

Equação 113 - Cálculo A113

A19 - Custo derivado ao seguro dos contentores;

A114 - Custo dos vários equipamentos no país de origem (em percentagem);

$$A114 = A103 \div (A103 + A104 \div i9) \text{ [%]}$$

Equação 114 - Cálculo A114

i9 - Identificação do câmbio/relação entre as moedas referência de dois tipos distintos;

A103 - Custo dos vários equipamentos no país de origem;

A104 - Custo de montagem (mão-de-obra) dos vários equipamentos no país de destino;

A115 - Custo de montagem (mão-de-obra) dos vários equipamentos no país de destino (em percentagem);

$$A115 = A104 \div (A104 + A103 \times i9) \text{ [%]}$$

Equação 115 - Cálculo A115

i9 - Identificação do câmbio/relação entre as moedas referência de dois tipos distintos;

A103 - Custo dos vários equipamentos no país de origem;

A104 - Custo de montagem (mão-de-obra) dos vários equipamentos no país de destino;

A116 - Margem da empresa no país de origem (em percentagem);

$$A116 = A105 \div A103 [\%]$$

Equação 116 - Cálculo A116

A103 - Custo dos vários equipamentos no país de origem;

A105 - Margem da empresa no país de origem;

A117 - Margem da empresa no país de destino (em percentagem);

$$A117 = A106 \div A77 [\%]$$

Equação 117 - Cálculo A117

A106 - Margem da empresa no país de destino;

A77 - Somatório dos preços de custo, no país de destino,

Na tabela 3.14 apresenta-se um exemplo de preenchimento da tabela resumo da orçamentação:

Tabela 3.14 - Resumo da orçamentação (exemplo)

RESUMO DA ORÇAMENTAÇÃO			
PV EMP.(Orig.) (Inc. Frete e Seguro)	10 841,93 €		
PV CIF EMP.(Dist.)	\$ 67 669,64		
PV FOB EMP.(Dist.)	\$ 47 518,15		
Custo(Equip., acc, e transportes) EMP.(Orig.)	125,60 €		0,48%
Custo Mont. EMP.(Dist.)	\$ 36 155,00		99,52%
Margem EMP.(Orig.)	18,84 €		15,00%
Margem EMP.(Dist.)	\$ 9 503,63		25,00%
IRT	\$ 1 272,50		
Imp. Selo	\$ 363,57		
Direitos	\$ -		
FCL	\$ 13 020,00		
Frete Interno	\$ 5 175,00		
Desalf.	\$ 21,23		
Seguros	\$ 3,49		

Nota: A sigla CIF significa que o frete e o seguro são pagos pelo fornecedor que é o responsável pela entrega até o local de destino. No caso da FOB o cliente é que paga o frete e o seguro da mercadoria em questão.

3.2.2.4 – Restantes elementos de cálculo

Devido aos vários tipos de exigências/normas aplicadas pelos clientes e de forma a responder a todos estes, este simulador terá vários tipos de documentos (pró-formas) disponibilizados para entrega ao cliente.

Nos pró-formas disponíveis existem as seguintes variáveis de referência:

A118 - Valor total referente ao frete marítimo;

$$A118 = A20 + A21 \text{ [€]}$$

Equação 118 - Cálculo A118

A20 - custo derivado ao frete marítimo;

A21 - Custo relacionado com o transporte interno no país de destino;

A119 - Valor final dos trabalhos referidos no mapa de quantidades, no país de destino;

$$A119 = A118 + A19 + A86 \text{ [€]}$$

Equação 119 - Cálculo A119

A19 - Custo derivado ao seguro dos contentores;

A86 - Somatório dos valores gastos relacionados com outros custos, como por exemplo, elementos exteriores ao fornecimento/realização dos trabalhos (parcial);

A118 - Valor total referente ao frete marítimo;

A120 - Preço parcial dos equipamentos (sem mão-de-obra), projetados no país de origem;

$$A120 = i36 \times A30 \text{ [€]}$$

Equação 120 - Cálculo A120

A30 - Preço de venda, referente a um equipamento específico no país de envio (unitário);

i36 - Identificação da quantidade de elementos referidos;

A121 - Valor do seguro, projetado no país de origem;

$$A121 = A19 \div i9 \text{ [€]}$$

Equação 121 - Cálculo A121

i9 - Identificação do câmbio/relação entre as moedas referência de dois tipos distintos;

A19 - Custo derivado ao seguro dos contentores;

A122 - Valor do frete, projetado no país de origem;

$$A122 = A20 \div i9 \text{ [€]}$$

Equação 122 - Cálculo A122

i9 - Identificação do câmbio/relação entre as moedas referência de dois tipos distintos;

A20 - custo derivado ao frete marítimo;

A123 - Valor total, referente ao pró-forma número 5;

$$A123 = A122 + A121 + A120 \text{ [€]}$$

Equação 123 - Cálculo A123

A120 - Preço parcial dos equipamentos (sem mão-de-obra), projetados no país de origem;

A121 - Valor do seguro, projetado no país de origem;

A122 - Valor do frete, projetado no país de origem;

A124 - Preço unitário da montagem, referente ao país de origem;

$$A124 = A27 \times i14 \text{ [€]}$$

Equação 124 - Cálculo A124

i14 - Definição por meio de percentagem da produtividade local;

A27 - Preço da montagem, referente a um equipamento específico (unitário);

A125 - Preço parcial montagem, referente ao país de origem;

$$A125 = A124 \times i36 \text{ [€]}$$

Equação 125 - Cálculo A125

i36 - Identificação da quantidade de elementos referidos;

A124 - Preço unitário da montagem, referente ao país de origem;

A126 - Somatório dos preços parcial da montagem, referente ao país de origem;

$$A126 = \sum A125 \text{ [€]}$$

Equação 126 - Cálculo A126

A125 - Preço parcial montagem, referente ao país de origem;

A127 - Valor total, referente ao pró-forma número 7;

$$A127 = A66 + A126 \text{ [€]}$$

Equação 127 - Cálculo A127

A66 - Somatório dos preços de venda, referente aos equipamentos no país de envio;

A126 - Somatório dos preços parcial da montagem, referente ao país de origem;

CAPÍTULO

4

Caso de estudo - Cálculos e *Output*

O 4º capítulo consiste na verificação/comprovação das várias fórmulas (cálculo manual), bem como o resultado verificado no programa (vertente informática) na aplicação do caso estudo.

4.1– Variáveis introduzidas

Para a verificação do bom funcionamento do programa e respetivos resultados foram executados vários testes, dos quais, um deles será a seguir apresentado e analisado.

Para a execução do respetivo teste teve-se se em contas os seguintes *Inputs*:

Tabela 4.1 - Valores das variáveis *Inputs*

	Valores
Variável introduzida	Valor introduzido
i1	Teste_1
i2	Angola
i3	50000
i4	EC_Int_001
i5	I.S.E.L
i6	07/03/2015
i7	Eurico Clemente
i8	Rev_01
i9	1,4
i10	15
i11	15
i12	25
i13	25
i14	1
i15	1,5
i16	4000
i17	2650
i18	1750
i19	0
i20	10,5
i21	0

i22	3,5
i23	1
i24	30
i25	30
i26	30
i27	3
i28	20
i29	2
i30	1
i31	<i>Building</i>
i32	<i>Lighting</i>
i33.1	1.1
i34.1	d 25
i35.1	m
i36.1	50
i37.1	0,35
i38.1	0,22
i39.1	0
i40.1	0
i41.1	1000
i42.1	0
i43.1	0
i33.2	1.2
i34.2	d 32
i35.2	m
i36.2	100
i37.2	0,53
i38.2	0,33
i39.2	0
i40.2	0
i41.2	0
i42.2	15

i43.2	0
i33.3	1.3
i34.3	d 40
i35.3	m
i36.3	10
i37.3	0,67
i38.3	0,44
i39.3	0
i40.3	0
i41.3	0
i42.3	0
i43.3	15
i44	0

Essencial ainda frisar o facto de estarem presentes três elementos/materiais distintos (valores compreendidos entre i33 e i43) e identificados como .1, .2 e .3.

4.2 – Cálculos comparativos das variáveis

Na tabela 4.2 esta expresso os valores referentes aos cálculos provenientes dos *Inputs* acima mencionados. Importa referir que o resultado e respetiva programação do teste encontra-se em anexo, como apêndice 3.

Tabela 4.2 - Cálculos comparativos do valor das variáveis

Fórmula de cálculo	Valores	
	Valor cálculo manual/teórico	Valor cálculo Programa
$A1 = i27 \times i28$	60	60
$A2 = A94 \times i14$	833,333	833,333
$A3 = A2 \div 8$	104,167	104,166

$A4 = A3 \div A1$	2	2
$A5 = A4 \div 2$	1	1
$A6 = A96 \times i14$	25	25
$A7 = A6 \div 8$	3,125	3,125
$A8 = A7 \div A1$	1	1
$A9 = A8 \div 2$	0	0
$A10 = A98 \times i14$	2,5	2,5
$A11 = A10 \div 8$	0,313	0,312
$A12 = A11 \div A1$	1	1
$A13 = A12 \div 2$	0	0
$A14 = A2 + A6 + A10$	860,83	860,83
$A15 = A14 \div 8$	108	108
$A16 = A15 \div A1$	2	2
$A17 = A16 \div 2$	1	1
$A18 = A84$	47518,15	47518,15
$A19 = [A66 \times i15 \times (1 + (i10 \text{ ou } i11))] \times i9$	3,49	3,49
$A20 = [i29 \times (i17 \times i9) + i30(i16 \times i9)]$ $+ [i29 \times (i17 \times i9) + i30(i16$ $\times i9)] \times (i10 \text{ ou } i11)$	14973	14973
$A21 = (i29 + i30) \times i18$	5175	5175
$A22 = A18 + A19 + A20 + A21$	67669,64	67669,64
$A23 = A20 \div A84$	31,51	31,51
$A24 = A21 \div A84$	10,89	10,89
$A27 = A57 + A58 + A59$	7,5	7,5
$A28 = i37 + i38 + i39$	1,11	1,11
$A29 = A28 \times (i10 \text{ ou } i11)$	0,17	0,17
$A30 = A28 + A29$	1,28	1,28
$A31 = A30 \times i9$	1,79	1,79
$A32 = A27 \times i14 \times i9$	10,50	10,50
$A33 = A31 + A32$	12,287	12,286

$A34 = A31 \times i20$	0,19	0,19
$A35 = A31 \times i21$	0	0
$A36 = A33 \times i22$	0,43	0,43
$A37 = A33 \times i23$	0,12	0,12
$A38 = (A31 + i40 + A34 + A35) + (A31 \times i22) + (A31 \times i23)$	2,06	2,06
$A39 = A32 + (A32 \times i22) + (A32 \times i23)$	10,97	10,97
$A40 = A38 + A39$	13,03	13,03
$A41 = A38 \times (i12 \text{ ou } i13)$	0,51	0,51
$A42 = A39 \times (i12 \text{ ou } i13)$	2,74	2,74
$A43 = A40 \times (i12 \text{ ou } i13)$	3,26	3,26
$A44 = A38 + A41$	2,57	2,57
$A45 = A39 + A42$	13,72	13,72
$A46 = A40 + A43$	16,28	16,28
$A47 = A46 \times i36$	162,85	162,85
$A48 = A46 \times i44$	16,28	16,28
$A49 = A48 \times i36$	162,85	162,85
$A50 = i36 \times A30 \times i15 \times [1 + (i10 \text{ ou } i11)] \times i9$	0,31	0,31
$A51 = A47 \times (A20 \div A84)$	51,31	51,31
$A52 = A47 \times (A21 \div A84)$	17,73	17,73
$A53 = A44 + (A50 + A51 + A52) \div i36$	9,5	9,5
$A54 = A45$	13,72	13,72
$A55 = A53 + A54$	23,22	23,22
$A56 = A47 + A50 + A51 + A52$	232,2	232,2
$A57 = i41 \times (i24 \div 60)$	0	0
$A58 = i42 \times (i25 \div 60)$	0	0
$A59 = i43 \times (i26 \div 60)$	7,5	7,5
$A60 = \sum (i36 \times i37)$	77,2	77,2

$A61 = \sum (i36 \times A27)$	25825	25825
$A62 = \sum (i36 \times i38)$	48,4	48,4
$A63 = \sum (i36 \times i39)$	0	0
$A64 = \sum (i36 \times A28)$	125,6	125,6
$A65 = \sum (i36 \times A29)$	18,84	18,84
$A66 = \sum (i36 \times A30)$	144,44	144,44
$A67 = \sum (i36 \times A31)$	202,22	202,22
$A68 = \sum (i36 \times A32)$	36155	36155
$A69 = \sum (i36 \times A33)$	36357,22	36357,22
$A70 = \sum (i36 \times i40)$	0	0
$A71 = \sum (i36 \times A34)$	21,23	21,23
$A72 = \sum (i36 \times A35)$	0	0
$A73 = \sum (i36 \times A36)$	1272,5	1272,5
$A74 = \sum (i36 \times A37)$	363,57	363,57
$A75 = \sum (i36 \times A38)$	232,55	232,55
$A76 = \sum (i36 \times A39)$	37781,98	37781,98
$A77 = \sum (i36 \times A40)$	38014,52	38014,52
$A78 = \sum (i36 \times A41)$	58,14	58,14
$A79 = \sum (i36 \times A42)$	9445,49	9445,49
$A80 = \sum (i36 \times A43)$	9503,63	9503,63
$A81 = \sum (i36 \times A44)$	290,69	290,69

$A82 = \sum (i36 \times A45)$	47227,47	47227,47
$A83 = \sum (i36 \times A46)$	47518,15	47518,15
$A84 = \sum A47$	47518,15	47518,15
$A85 = \sum A48$	947,68	947,68
$A86 = \sum A49$	47518,15	47518,15
$A87 = \sum A50$	3,49	3,49
$A88 = \sum A51$	14973	14973
$A89 = \sum A52$	5175	5175
$A90 = \sum (i36 \times A53)$	20442,17	20442,17
$A91 = \sum (i36 \times A54)$	47227,47	47227,47
$A92 = \sum (i36 \times A55)$	67669,64	67669,64
$A94 = \left[\sum (i36 \times i41) \right] \div 60$	833,33	833,33
$A95 = \sum (i36 \times A57)$	25000	25000
$A96 = \left[\sum (i36 \times i42) \right] \div 60$	25	25
$A97 = \sum (i36 \times A58)$	750	750
$A98 = \left[\sum (i36 \times i43) \right] \div 60$	2,5	2,5
$A99 = \sum (i36 \times A59)$	75	75
$A100 = A66 + [A66 \times i15 \times (1 + (i10 \text{ ou } i11))] + (A20 \div i9)$	10841,93	10841,93
$A101 = A22$	67669,64	67669,64
$A102 = A18$	47518,15	47518,15
$A103 = A64$	125,6	125,6

$A104 = A61 \times i9 \times i14$	36155	36155
$A105 = A65$	18,84	18,84
$A106 = A80$	9503,63	9503,63
$A107 = A73$	1272,5	1272,5
$A108 = A74$	363,57	363,57
$A109 = A72$	0	0
$A110 = (i29 \times i17 + i30 \times i16) \times i9$	13020	13020
$A111 = (i29 + i30) \times i18$	5175	5175
$A112 = A71$	21,23	21,23
$A113 = A19$	3,49	3,49
$A114 = A103 \div [A103 + A104 \div i9]$	0,48	0,48
$A115 = A104 \div [A104 + A103 \times i9]$	99,52	99,52
$A116 = A105 \div A103$	15	15
$A117 = A106 \div A77$	25	25
$A118 = A20 + A21$	20148	20148
$A119 = A118 + A19 + A86$	67669,64	67669,64
$A120 = i36 \times A30$	12,77	12,77
$A121 = A19 \div i9$	2,49	2,49
$A122 = A20 \div i9$	10695	10695
$A123 = A122 + A121 + A120$	10841,93	10841,93
$A124 = A27 \times i14$	7,5	7,5
$A125 = A124 \times i36$	75	75
$A126 = \sum A125$	25825	25825
$A127 = A66 + A126$	25969,44	25969,44

Através da análise da tabela 4.2, confirma-se o correto funcionamento do programa de cálculo desenvolvido em linguagem *Java* e alvo desta dissertação.

As diferenças relacionadas com a casa decimal das milésimas proveem do cálculo manual não ser cem por cento correto, uma vez que no mesmo houve arredondamentos.

É importante salientar o facto das fórmulas anteriormente referidas encontrarem-se numeradas por ordem crescente e seguindo o padrão definido no programa/simulador desenvolvido. Devido a isto as variáveis definidas como A25 e A26 não se encontram indicadas, pois estas foram usadas como variáveis auxiliares (permitindo guardar valores de transição, uma vez que estes são necessários para o bom funcionamento do programa/simulador supracitado).

CAPÍTULO

5

Modo de utilização

O 5º capítulo aborda a temática do modo de utilização relacionado com programa desenvolvido.

Desta forma, pode-se definir este capítulo como um “manual de instruções”.

5 – Introdução ao *UI* (*user interface*)

O programa na sua vertente *UI* está repartido em três partes. Esta divisão surge de forma a criar um fácil manuseamento por parte do utilizador do programa.

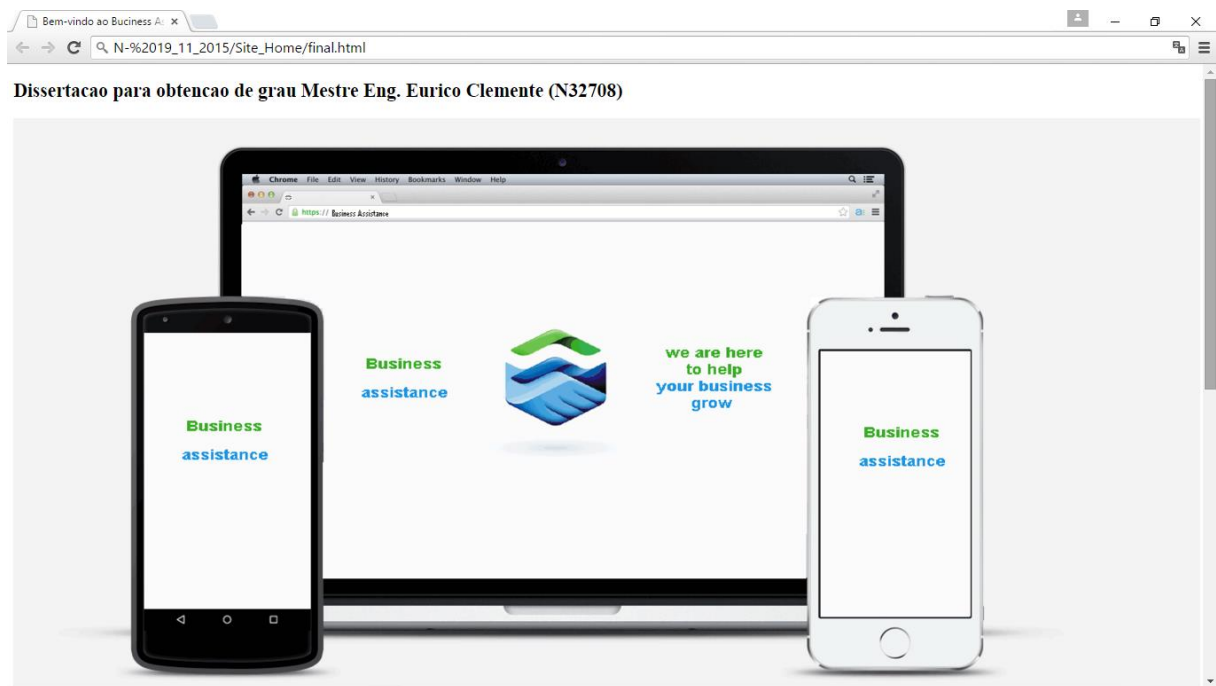


Figura 5.1 - Janela inicial

A primeira parte é referente aos clientes e está mencionada como *Client* (dentro do campo *Management*, juntamente com o *units* e *currencies*). A segunda é alusiva aos materiais e é denominada como *Materials*. A terceira parte apresenta as propostas e tem como nome *Proposal*.

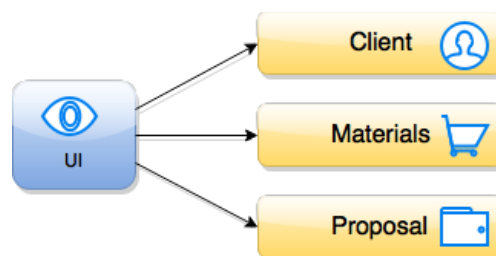


Figura 5.2 - Composição vertente UI (sistema)



Figura 5.3 - Composição vertente UI (aplicada)

5.1 – *Client*

Um cliente representa um indivíduo ou entidade determinado, para o qual um orçamento se destina. Este campo está dentro do *management* (juntamente com *units* e *currencies*).

Há ainda a referir que este campo está por sua vez repartido em três divisórias, criação, detalhes e consulta.

Criação: Esta divisória encontra-se referenciada como *new client*. Nela é preciso atribuir um nome, email, localização e contacto telefónico do mesmo, conforme exposto na figura 5.4.

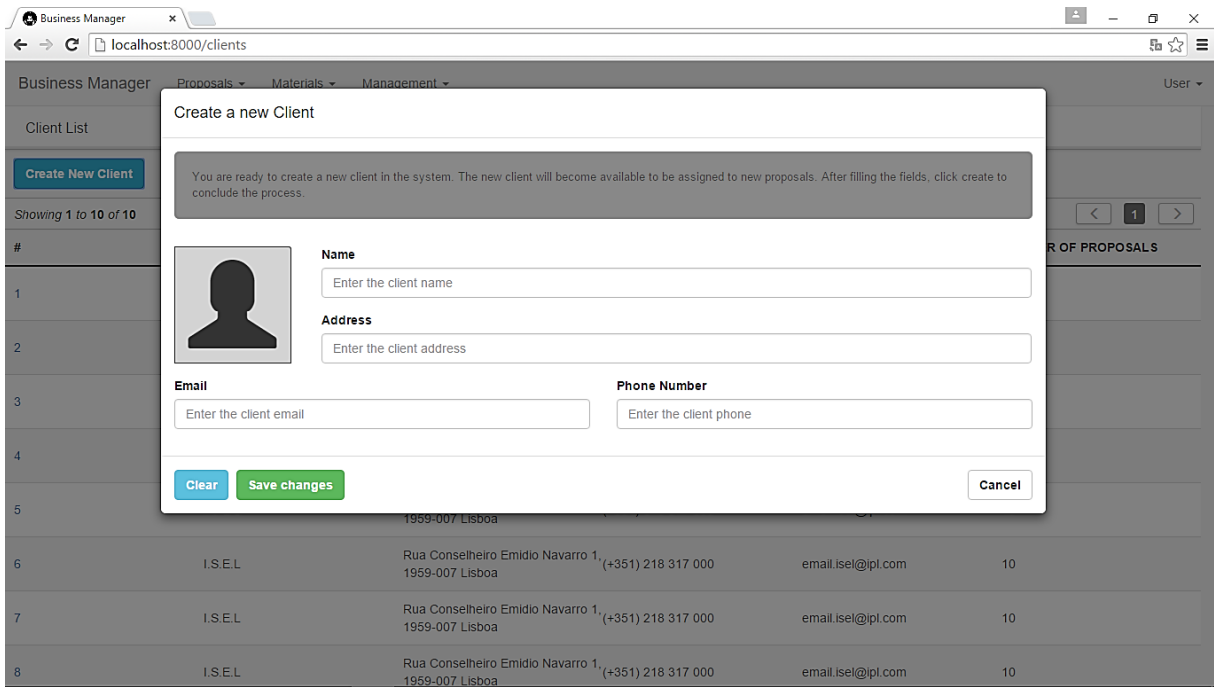


Figura 5.4 - Janela New Client

Detalhes: Nesta estão expressos os elementos mais importantes referentes ao cliente (figura 5.5).

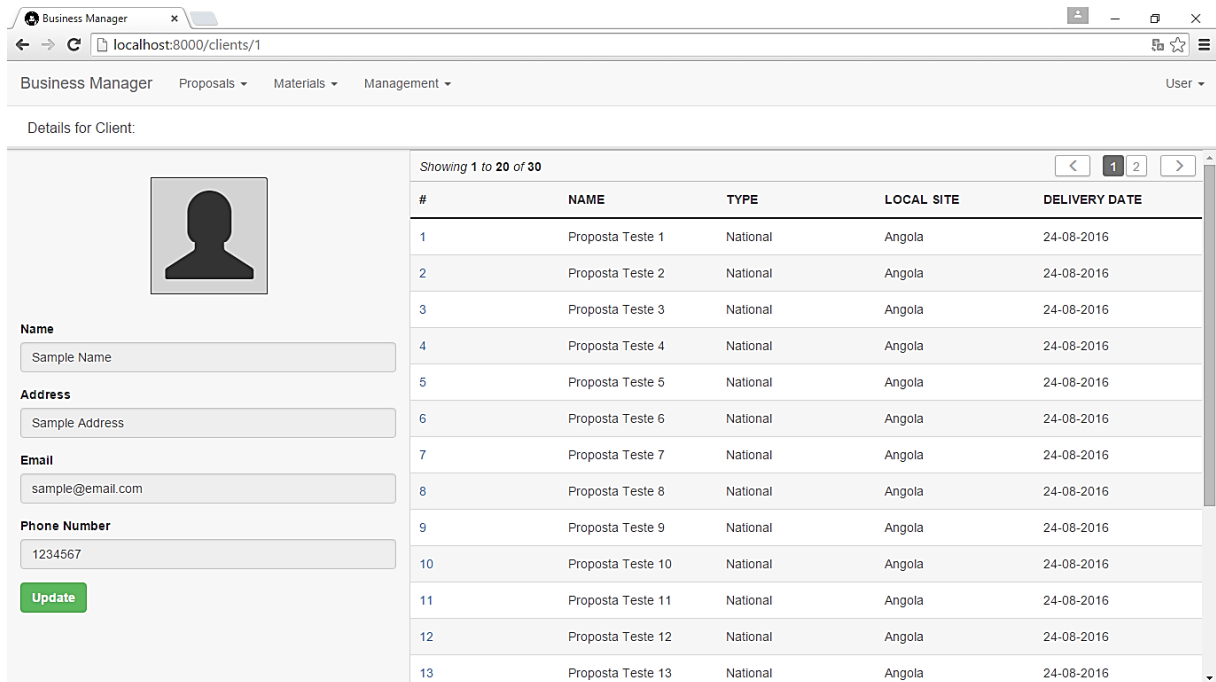
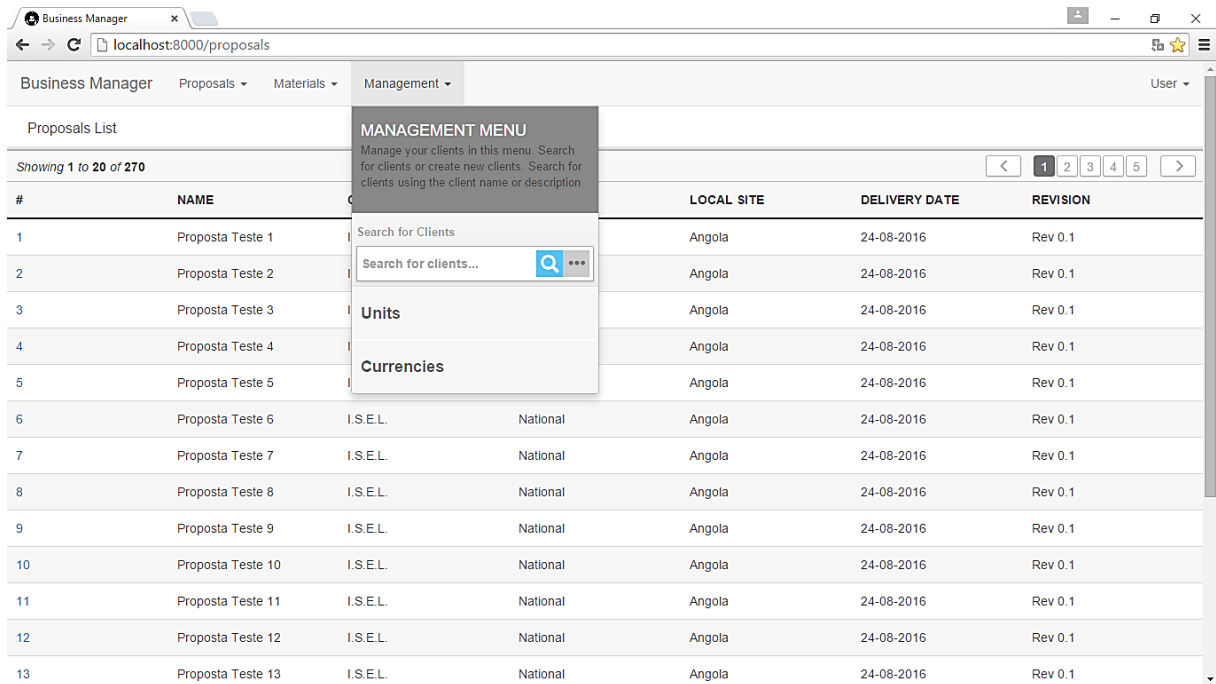


Figura 5.5 - Janela Client Details

Consulta: Esta divisória, conforme o próprio nome indica a consulta de clientes dentro da respetiva base de dados (figura 5.6).

Nota para o facto de a Consulta comunicar com os Detalhes, criando desta forma uma otimização de recursos bem como de tempo (fácil utilização).



The screenshot shows a web browser window with the URL localhost:8000/proposals. The application interface includes a navigation bar with 'Business Manager', 'Proposals', 'Materials', and 'Management' menus. A 'Proposals List' table is displayed, showing 13 test proposals. A 'MANAGEMENT MENU' is overlaid on the table, providing search and management options for clients, units, and currencies.

#	NAME	LOCAL SITE	DELIVERY DATE	REVISION		
1	Proposta Teste 1	Angola	24-08-2016	Rev 0.1		
2	Proposta Teste 2	Angola	24-08-2016	Rev 0.1		
3	Proposta Teste 3	Angola	24-08-2016	Rev 0.1		
4	Proposta Teste 4	Angola	24-08-2016	Rev 0.1		
5	Proposta Teste 5	Angola	24-08-2016	Rev 0.1		
6	Proposta Teste 6	I.S.E.L.	National	Angola	24-08-2016	Rev 0.1
7	Proposta Teste 7	I.S.E.L.	National	Angola	24-08-2016	Rev 0.1
8	Proposta Teste 8	I.S.E.L.	National	Angola	24-08-2016	Rev 0.1
9	Proposta Teste 9	I.S.E.L.	National	Angola	24-08-2016	Rev 0.1
10	Proposta Teste 10	I.S.E.L.	National	Angola	24-08-2016	Rev 0.1
11	Proposta Teste 11	I.S.E.L.	National	Angola	24-08-2016	Rev 0.1
12	Proposta Teste 12	I.S.E.L.	National	Angola	24-08-2016	Rev 0.1
13	Proposta Teste 13	I.S.E.L.	National	Angola	24-08-2016	Rev 0.1

Figura 5.6 - Janela Client Results

5.2 – Materials

Os materiais são o pilar base da orçamentação, pois toda esta baseia-se na necessidade de os vender/fornecer ou mesmo de os comprar. Deste modo, os materiais são um dos três elementos a ser referidos no *UI*.

Seguindo a mesma cadeia de pensamento e a elaboração do simulador, explicada anteriormente (*Client*), este tópico (*Materials*) será repartido em três divisórias (criação, detalhes e consulta).

Criação: Nesta divisória pode-se criar materiais específicos para uma proposta, bem como arquivá-los na base de dados para posterior utilização em propostas futuras.

Business Manager

localhost:8000/materials/new

Business Manager Proposals Materials Management User

Create a new Material

You are ready to create a new material in the system. This new material will become available to be added into new or current proposals. After filling the fields, click create to conclude the process.

Numeric Code
Enter a numeric code

Description
Enter a material description

Type -- Select a Material Category --

SubType -- Select a Material Sub Category --

Unit -- Select a unit --

Ref. Price Set the Reference cost for Price

Ref. Price Accessorise Set the Reference cost for Accessorise

Ref. Price Transport Set the Reference cost for Transportation

Ref. Time Assembly 1 Set the Reference cost for Time Assembly 1

Ref. Time Assembly 2 Set the Reference cost for Time Assembly 2

Ref. Time Assembly 3 Set the Reference cost for Time Assembly 3

Clear Create

Figura 5.7 - Janela *New Material*

Detalhes: Os *Details* dos *Materials* permite para além dos restantes (*Clients* e *Proposal*) uma análise entre preços nas várias propostas como, por exemplo, o preço máximo ou mínimo. Nesta divisória estão expressos os elementos mais importantes referentes aos materiais.

Business Manager

localhost:8000/materials/1

Business Manager Proposals Materials Management User

Materials Details: 1

You are viewing the material details. In this section you can also update the material information by clicking in update button.

Numeric Code
12345

Description
f 30

Type Building & Construction

SubType Lighting

Unit Galons [g]

Ref. Price 5

Ref. Price Accessorise 10

Ref. Price Transport 20

Ref. Time Assembly 1 30

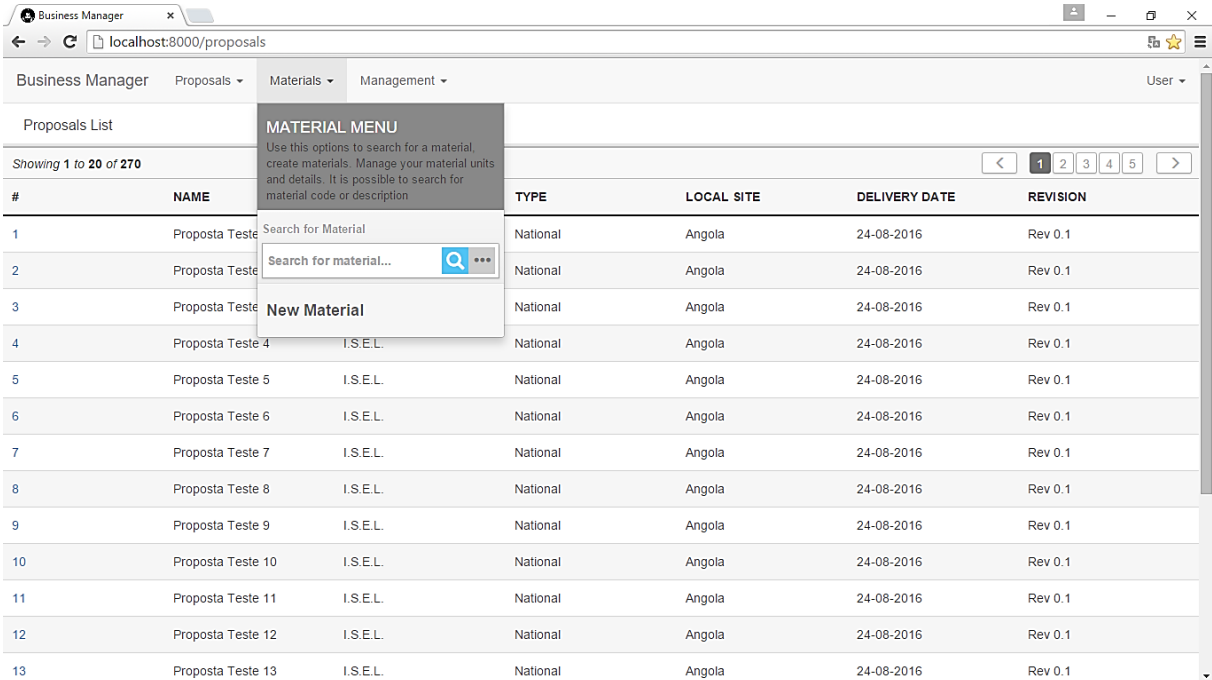
Ref. Time Assembly 2 40

Ref. Time Assembly 3 50

Update

Figura 5.8 - Janela *Material Details*

Consulta: Esta divisória exibe resultados idênticos aos referidos no *Client Results*, mas referente aos *Materials*.



Business Manager Proposals Materials Management User

Proposals List

Showing 1 to 20 of 270

MATERIAL MENU
Use this options to search for a material, create materials. Manage your material units and details. It is possible to search for material code or description

#	NAME		TYPE	LOCAL SITE	DELIVERY DATE	REVISION
1	Proposta Teste	Search for Material	National	Angola	24-08-2016	Rev 0.1
2	Proposta Teste	Search for material...	National	Angola	24-08-2016	Rev 0.1
3	Proposta Teste	New Material	National	Angola	24-08-2016	Rev 0.1
4	Proposta Teste 4	I.S.E.L.	National	Angola	24-08-2016	Rev 0.1
5	Proposta Teste 5	I.S.E.L.	National	Angola	24-08-2016	Rev 0.1
6	Proposta Teste 6	I.S.E.L.	National	Angola	24-08-2016	Rev 0.1
7	Proposta Teste 7	I.S.E.L.	National	Angola	24-08-2016	Rev 0.1
8	Proposta Teste 8	I.S.E.L.	National	Angola	24-08-2016	Rev 0.1
9	Proposta Teste 9	I.S.E.L.	National	Angola	24-08-2016	Rev 0.1
10	Proposta Teste 10	I.S.E.L.	National	Angola	24-08-2016	Rev 0.1
11	Proposta Teste 11	I.S.E.L.	National	Angola	24-08-2016	Rev 0.1
12	Proposta Teste 12	I.S.E.L.	National	Angola	24-08-2016	Rev 0.1
13	Proposta Teste 13	I.S.E.L.	National	Angola	24-08-2016	Rev 0.1

Figura 5.9 - Janela *Material Results*

5.3 – Proposal

O último dos elementos referido no *UI* é a *Proposal*. Este campo representa a proposta na sua vertente comercial.

Uma vez mais, este elemento está repartido em três divisórias:

Criação: Nesta divisória pode-se criar propostas (nacionais ou internacionais), bem como introduzir informação adicional.

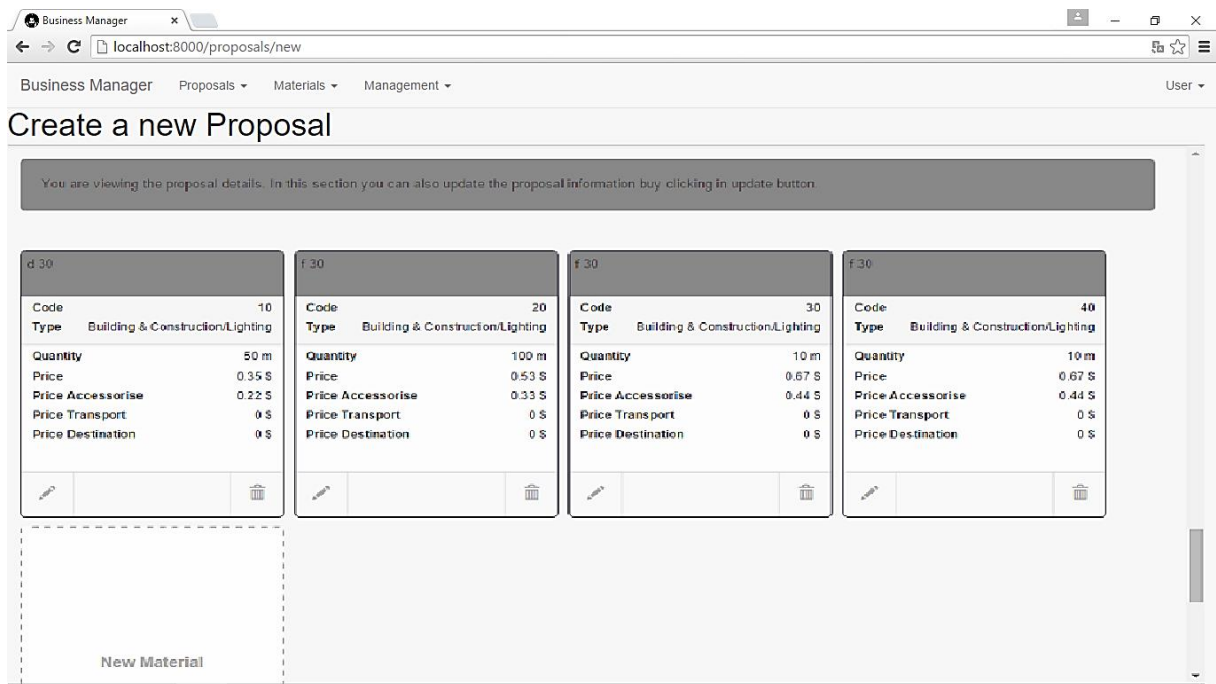


Figura 5.10 - Janela *New Proposal*

Detalhes: Nos *Details* da *Proposal* é possível fazer uma revisão/inspeção de toda a orçamentação referente à proposta identificada.

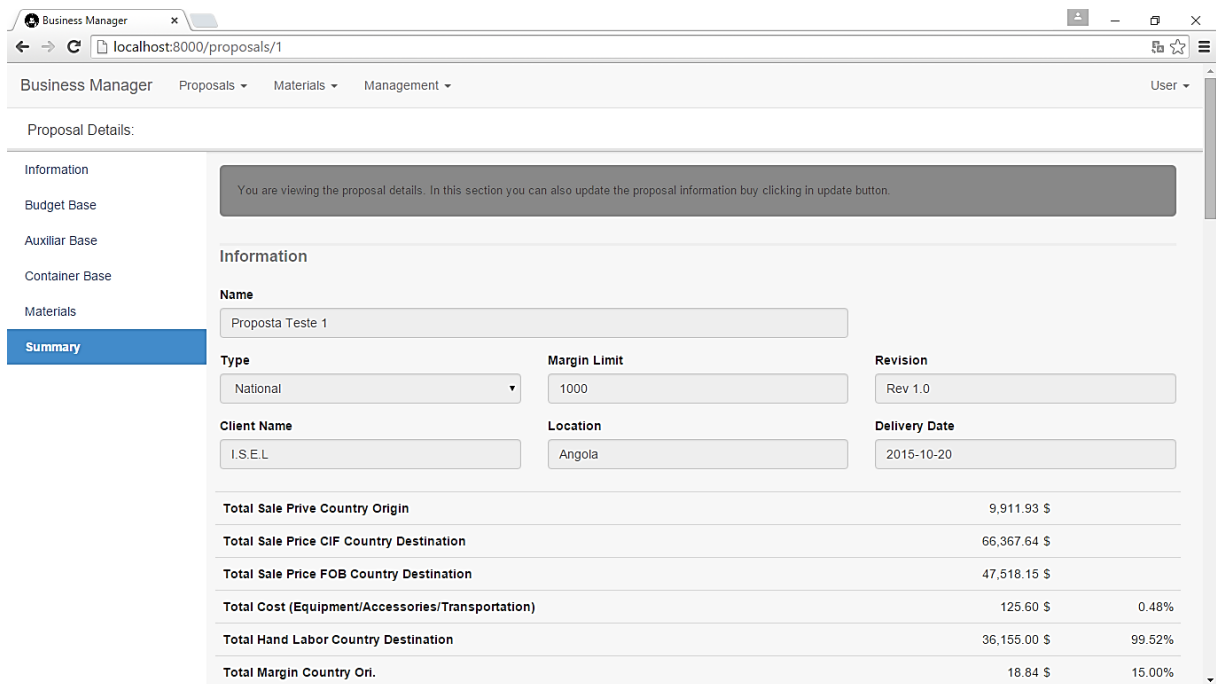


Figura 5.11 - Janela *Proposal Details*

Consulta: Conforme exposto anteriormente a consulta representa a capacidade para pesquisar um determinado elemento na base de dados. No caso das *Proposal* o mesmo executa uma ação idêntica aos referidos no *Clients e Materials*.

The screenshot shows a web browser window with the URL `localhost:8000/proposals`. The application interface includes a navigation bar with 'Business Manager', 'Proposals', 'Materials', and 'Management' menus. A 'Proposals List' section is visible, showing a table of proposals. A 'PROPOSAL MENU' is overlaid on the table, containing a search box and a 'New Proposal' button. The table data is as follows:

#			TYPE	LOCAL SITE	DELIVERY DATE	REVISION
1			National	Angola	24-08-2016	Rev 0.1
2			National	Angola	24-08-2016	Rev 0.1
3			National	Angola	24-08-2016	Rev 0.1
4	Proposta Teste 4	I.S.E.L.	National	Angola	24-08-2016	Rev 0.1
5	Proposta Teste 5	I.S.E.L.	National	Angola	24-08-2016	Rev 0.1
6	Proposta Teste 6	I.S.E.L.	National	Angola	24-08-2016	Rev 0.1
7	Proposta Teste 7	I.S.E.L.	National	Angola	24-08-2016	Rev 0.1
8	Proposta Teste 8	I.S.E.L.	National	Angola	24-08-2016	Rev 0.1
9	Proposta Teste 9	I.S.E.L.	National	Angola	24-08-2016	Rev 0.1
10	Proposta Teste 10	I.S.E.L.	National	Angola	24-08-2016	Rev 0.1
11	Proposta Teste 11	I.S.E.L.	National	Angola	24-08-2016	Rev 0.1
12	Proposta Teste 12	I.S.E.L.	National	Angola	24-08-2016	Rev 0.1
13	Proposta Teste 13	I.S.E.L.	National	Angola	24-08-2016	Rev 0.1

Figura 5.12 - Janela *Proposal Results*

Importante salientar que as propostas devem ser criadas pela mesma ordem de execução acima referida (*Client, Material e Proposal*).

CAPÍTULO

6

Conclusões e desenvolvimentos futuros

Apresenta-se neste capítulo um resumo das ilações mais significativas resultantes da elaboração desta dissertação.

Neste capítulo estão, ainda, indicados possíveis desenvolvimentos futuros do trabalho efetuado.

6.1 – Principais conclusões

Devido ao grande número de orçamentos “exigidos” pelos requerentes não é fisicamente possível, às pequenas empresas (menos trabalhadores), concorrer com as empresas de maior dimensão (maior número de trabalhadores).

Mesmo nas empresas de maior dimensão, algumas vezes os orçamentos/propostas comerciais são realizados de forma ineficiente.

No decorrer desta dissertação foram abordadas as principais razões e as causas para tal comportamento. Para evitar estas situações e para melhorar todo este processo foi elaborado um simulador que visa colmatar estes problemas.

Este simulador pretende substituir e padronizar a forma como as empresas constroem as suas propostas comerciais.

As atuais formas para execução dos orçamentos utilizam ferramentas rudimentares como *Word*, *Excel* ou programas proprietários.

Estas plataformas de cálculo não são exatas e compõem um vasto leque de componentes que prolongam o tempo necessário para a execução das propostas.

A dissertação foi construída com o objetivo de salientar as fórmulas de cálculo uma vez, que estas foram realizadas/concebidas de forma a poderem responder aos vários tipos de utilizadores, bem como as suas exigências. De salientar ainda o facto destas fórmulas matemáticas serem a base/pilar para a realização do ODO.

Do ponto de vista técnico nesta dissertação, foram utilizadas linguagens de programação como, por exemplo, linguagem *Java*. Foi realizado um cálculo exaustivo e minucioso de forma a construir uma plataforma (programa) de cálculo estável e notoriamente *User Friendly*.

Realizou-se, ainda, uma pesquisa intensiva em vários campos, como a orçamentação de obra, importação/exportação de bens, decretos de leis e normas em vigor.

Tabela 6.1 - Comparação de *Softwares* (conclusões)

Classificações a comparar	<i>Softwares</i>		
	<i>OranGest Obras</i>	<i>Quote Roller</i>	<i>Business Assistance (dissertação)</i>
User Friendly	Não	Sim (possibilita que o <i>software</i> seja utilizado por um leque variado de pessoas sem as mesmas terem instrução para tal tarefa)	Sim
Aspetto gráfico	Fraco	Bom	Bom
Funcionalidade	Bom	Médio	Bom
Conteúdos	Abrangente (constituído por diversos módulos: Obras, Materiais, Pessoal, Equipamentos, Orçamentos e Autos de Medição)	Exíguo	Abrangente (constituído por diversas bases, como por exemplo, Bases para identificação)
Tempo de processamento	Médio	Baixo	Baixo
Variedade de plataformas (androide, Windows, ios, outros)	Alto	Baixo	Alto
Banco de dados	Alto	Alto	Alto
Licença de Software	Pagamento	Pagamento	Free

6.2 – Desenvolvimentos futuros

Este tipo de tema (programa/simulador), sendo uma ferramenta informática é necessária uma constante atualização/manutenção, pois estas evoluem em paralelo com as necessidades dos utilizadores. Estas atualizações/manutenções podem ser minimizadas com um melhor aproveitamento de espaço (memória informática) e acompanhamento temporal do decorrer da utilização deste (programa/simulador).

É de interesse no futuro a possibilidade de ampliação do mesmo (programa/simulador) a outras áreas ou campos de estudo, como a contabilidade das empresas.

Desenvolvimentos futuros e importantes são ainda a melhoria de alguns componentes do programa desenvolvidos, como a componente linguista, na adição de novas línguas (como, por exemplo o Francês ou Mandarim).

REFERÊNCIAS BIBLIOGRÁFICAS

No capítulo agora descrito são apresentadas as várias referências bibliográficas, consultadas no decorrer desta dissertação.

Referências bibliográficas Programas

- [1] OranGest (2014, Novembro), *Programa Orçamentação [Online]*
Disponível: <http://www.magnisoft.pt/OranGest-OBRAS.aspx>.
- [2] Quote Roller (2014, Novembro), *Programa Orçamentação [Online]*
Disponível: <http://www.quoteroller.com>.
- [3] IntelliJ IDEA 13.1.5 (2015, Janeiro), *Framework coding assistance*
Disponível: <http://www.jetbrains.com>.

Referências bibliográficas Livros e apontamentos

- [4] Elizabete Nunes, *Fatores que influenciam as negociações*
Apontamentos Universidade Gestão (Aula nº5), 2008.
- [5] Rovigati Alyrio, *Negociação e Processo Decisório*
Fundação Cecierj, 2011.
- [6] Gonçalo João, *Excel Avançado*
Instituto Superior de Economia e Gestão, 2015.
- [7] Bruce Eckel, *Thinking in Java, 4th Edition*
Prentice Hall, 2006.
- [8] Martin Holzke e Tom Stachowitz, *SQL Database for Beginners*
Learntoprogram, 2014.
- [9] Coutinho Puccini, *Matemática Financeira e Análise de Investimentos*
Ministério da Educação, 2011.
- [10] Abelardo Lima, *Matemática Financeira Objetiva e Aplicada (8ª edição)*
Saraiva, 2009.
- [11] Nicole Bartelt , *Global CSS settings*
Bootstrap, 2015.

- [12] Jack Mckalling, *Matemática Financeira Objetiva e Aplicada (8ª edição)* W3Schools, 2015.
- [13] Armando José, *Matemática Financeira Com Utilização do Excel 2010* Editora Atlas, 2012.
- [14] José A Faria, *Noções elementares sobre orçamentos de obras de construção civil (10 versão)* Faculdade Engenharia Universidade Porto, 2014.
- [15] Estudos da Central de Balanços, *Análise do setor da construção* Banco De Portugal, 2014.
- [16] Ana Botelho, *Modelo de controlo de custos de uma obra pública, do ponto de vista do dono da obra* Instituto Superior Técnico, 2009.
- [17] Manual do Formando, *Segurança, Higiene e Saúde do Trabalho da Construção Civil* CICCOPN, 2005.
- [18] Maçahico Tisaka, *Orçamento na Construção Civil* Pini, 2006.

APÊNDICE

1

Identificação das variáveis

APÊNDICE

1.1

Input

Apêndice nº1 - Identificação das variáveis

1.1 - Input

Input

- i1** – Definição do nome da obra, conforme expresso na memória descritiva e justificativa (MDJ);
- i2** - Identificação da localização da obra, conforme o concelho ou província destinatária;
- i3** - Descrição do limite na qual a margem A se altera para a margem B (Nota: margem A e B serão referidas posteriormente nos elementos compreendidos entre i10 e i14);
- i4** - Identificação da referência interna da empresa concorrente à obra;
- i5** - Identificação do cliente destinatário da proposta;
- i6** - Definição da data para a qual a proposta é entregue ao destinatário (solicitante);
- i7** - Indicação do nome da pessoa responsável pela receção da proposta (lado do cliente);
- i8** - Indicação do número de revisão da proposta;
- i9** - Identificação do câmbio/relação entre as moedas referência de dois tipos distintos, particularmente, neste exemplo, será utilizada a relação do câmbio entre Angola e Portugal \$/€;
- i10** - Identificação da margem de venda em Portugal (margem definida para elementos com valor inferior ao referido em i3);
- i11** - Identificação da margem de venda em Portugal (margem definida para elementos com valor superior ao referido em i3);
- i12** - Representação da margem de venda no país de destino (margem definida para elementos com valor inferior ao referido em i3);
- i13** - Identificação da margem de venda no país de destino (margem definida para elementos com valor superior ao referido em i3);
- i14** - Definição por meio de percentagem da produtividade local;
- i15** - Definição do valor do seguro (em percentagem);
- i16** - Representação do valor de custo de um contentor de 40” (40 pés);
- i17** - Representação do valor de custo de um contentor de 20” (20 pés);
- i18** - Identificação do custo com transporte de cada contentor no país de destino;
- i19** - Identificação do custo com transporte dos contentores em Portugal;
- i20** - Definição dos custos relacionados com o desalfandegamento (em percentagem);
- i21** - Identificação dos custos relacionados com os direitos (em percentagem);
- i22** - Identificação dos custos relacionados com a retenção IRT (em percentagem);
- i23** - Identificação dos custos relacionados com o imposto selo (em percentagem);

Apêndice nº1 - Identificação das variáveis

1.1 - Input

- i24** - Definição de um dos três valores possíveis para o custo de mão-de-obra à hora;
- i25** - Definição de um dos três valores possíveis para o custo de mão-de-obra à hora;
- i26** - Definição de um dos três valores possíveis para o custo de mão-de-obra à hora;
- i27** - Definição do número de meses atribuído (pelo cliente) para a execução da obra;
- i28** - Definição do número de dias úteis/trabalháveis por mês;
- i29** - Representação do número de um contentor de 20” utilizados;
- i30** - Representação do número de um contentor de 40” utilizados;
- i31** - Representação o tipo do material;
- i32** - Representação o subtipo do material;
- i33** - Identificação do código numérico endereço (ID) a um elemento específico;
- i34** - Descrição ortográfica de um elemento específico;
- i35** - Identificação do tipo de unidade, em que os valores estão expressos (tabela 3.11);
- i36** - Identificação da quantidade de elementos referidos;
- i37** - Definição do valor unitário do equipamento;
- i38** - Definição do valor unitário dos acessórios (referentes ao equipamento identificado na variável anterior);
- i39** - Definição do valor referente ao transporte do equipamento;
- i40** - Identificação do valor monetário referente aos equipamento locais;
- i41** - Identificação do tempo de montagem do equipamento em minutos (tendo como referência o custo de Hora/Homem -1);
- i42** - Identificação do tempo de montagem do equipamento em minutos (tendo como referência o custo de Hora/Homem -2);
- i43** - Identificação do tempo de montagem do equipamento em minutos (tendo como referência o custo de Hora/Homem -3);
- i44** - Identificação do valor gasto relacionado com outros custos, como elementos exteriores ao fornecimento/realização dos trabalhos (em percentagem) (somente nas propostas com países de destino e de origem diferentes);
- i45** - Definição das condições de venda.

APÊNDICE

1.2

Cálculos e Output

Apêndice nº1 - Identificação das variáveis

1.2 - Cálculos e Output

Cálculos e Output

- A1** - Número total de dias atribuído (pelo cliente) para a execução da obra;
- A2** - Número de horas total relacionada com o custo hora/homem -1;
- A3** - Número de dias totais relacionada com o custo hora/homem -1;
- A4** - Número de homens necessário, para realizar os trabalhos relacionados com o custo hora/homem -1;
- A5** - Número de equipas compostas por um oficial mais um ajudante (custo hora/homem -1);
- A6** - Número de horas total relacionada com o custo hora/homem -2;
- A7** - Número de dias total relacionada com o custo hora/homem -2;
- A8** - Número de homens necessários, para realizar os trabalhos relacionados com o custo hora/homem -2;
- A9** - Número de equipas compostas por um oficial mais um ajudante (custo hora/homem -2);
- A10** - Número de horas total relacionada com o custo hora/homem -3;
- A11** - Número de dias total relacionada com o custo hora/homem -3;
- A12** - Número de homens necessários, para realizar os trabalhos relacionados com o custo hora/homem -3;
- A13** - Número de equipas compostas por um oficial mais um ajudante (custo hora/homem -3);
- A14** - Número total de horas gasto na execução da obra;
- A15** - Número total de dias gasto na realização da empreitada;
- A16** - Número total de homens necessários, para realizar os trabalhos;
- A17** - Número de equipas totais compostas por um oficial mais um ajudante;
- A18** - Somatório dos custos finais dos equipamentos no país de destino;
- A19** - Custo derivado ao seguro dos contentores;
- A20** - custo derivado ao frete marítimo;
- A21** - Custo relacionado com o transporte interno no país de destino;
- A22** - Somatório dos elementos referenciados como bases contentores;
- A23** - Valor do custo relacionado com o frete marítimo (em percentagem);

Apêndice nº1 - Identificação das variáveis

1.2 - Cálculos e Output

- A24** - Valor referente ao custo relacionado com o transporte interno no país de destino (em percentagem);
- A25** – Valor auxiliar 1, necessário para o bom funcionamento do programa (guarda temporariamente valores de outras variáveis);
- A26** – Valor auxiliar 2, necessário para o bom funcionamento do programa (guarda temporariamente valores de outras variáveis);
- A27** - Preço da montagem, referente a um equipamento específico (unitário);
- A28** - Preço de custo, referente a um equipamento específico (unitário);
- A29** - Margem no equipamento referente ao país de envio (unitária);
- A30** - Preço de venda, referente a um equipamento específico no país de envio (unitário);
- A31** - Custo do material físico, no país de destino (unitário);
- A32** - Custo relacionado com a mão-de-obra, no país de destino (unitário);
- A33** - Custo referente ao equipamento, no país de destino (unitário);
- A34** - Valor de custo destinado ao desalfandegamento (unitário);
- A35** - Valor destinado aos direitos (unitário);
- A36** - Valor destinado a retenção IRT (unitário);
- A37** - Valor destinado ao imposto selo (unitário);
- A38** - Custo físico total, referente a um equipamento específico no país de destino (unitário);
- A39** - Custo alusivo a mão-de-obra total, referente a um equipamento específico no país de destino (unitário);
- A40** - Custo total, referente a um equipamento específico no país de destino (unitário);
- A41** - Margem no equipamento físico referente ao país de destino (unitária);
- A42** - Margem relacionada com a mão-de-obra, referente ao país de destino (unitário);
- A43** - Margem total no equipamento referente ao país de destino (unitária);
- A44** - Preço de venda dos materiais, no país de destino (unitário);
- A45** - Preço de venda referente a mão-de-obra, no país de destino (unitário);
- A46** - Preço de venda total do equipamento, no país de destino (unitário);
- A47** - Preço de venda total dos vários equipamentos, referenciados como iguais e quantificados na mesma célula, no país de destino (parcial);
- A48** - Valor gasto relacionado com outros custos, como elementos exteriores ao fornecimento/realização dos trabalhos (unitário);

Apêndice nº1 - Identificação das variáveis

1.2 - Cálculos e Output

- A49** - Valor gasto relacionado com outros custos, como elementos exteriores ao fornecimento/realização dos trabalhos (parcial);
- A50** - Valor destinado as despesas com transporte (seguro);
- A51** - Valor destinado as despesas com transporte (frete);
- A52** - Valor destinado as despesas com transporte interno, no país de destino;
- A53** - Preço de venda dos materiais e do transporte, no país de destino (unitário);
- A54** - Preço referente a mão-de-obra e transporte, no país de destino (unitário);
- A55** - Preço de venda total e do transporte, no país de destino (unitário);
- A56** - Preço de venda total e do transporte, no país de destino (parcial);
- A57** - Preço de custo relativo à mão-de-obra, do tipo 1;
- A58** - Preço de custo relativo à mão-de-obra, do tipo 2;
- A59** - Preço de custo relativo à mão-de-obra, do tipo 3;
- A60** - Somatório dos preços dos equipamentos descritos no mapa de quantidade;
- A61** - Somatório dos preços das montagens dos equipamentos descritos no mapa de quantidade;
- A62** - Somatório dos preços dos acessórios, necessários para o bom funcionamento dos equipamentos referidos;
- A63** - Valor destinado aos transportes;
- A64** - Somatório dos vários preços de custo, referente aos equipamentos;
- A65** - Somatório das margens, referente ao país de envio;
- A66** - Somatório dos preços de venda, referente aos equipamentos no país de envio;
- A67** - Somatório alusivo ao custo dos materiais no país de destino;
- A68** - Somatório do custo da mão-de-obra no país de destino;
- A69** - Somatório do custo dos equipamentos no país de destino;
- A70** - Somatório do custo relacionado com os equipamentos locais, no país de destino;
- A71** - Somatório dos custos derivados ao desalfandegamento;
- A72** - Somatório dos custos derivados aos direitos;
- A73** - Somatório dos custos derivados à retenção IRT;
- A74** - Somatório dos custos oriundos do imposto de selo;
- A75** - Somatório dos custos dos vários equipamentos referidos, referente ao país de destino;
- A76** - Somatório dos custos, referentes à mão-de-obra dos vários equipamentos referidos, no país de destino;

Apêndice nº1 - Identificação das variáveis

1.2 - Cálculos e Output

- A77 - Somatório dos preços de custo, no país de destino,
- A78 - Somatório das margens alusivas aos materiais no país de destino;
- A79 - Somatório das margens alusivas à mão-de-obra no país de destino;
- A80 - Somatório das margens, no país de destino;
- A81 - Somatório dos preços de venda dos materiais;
- A82 - Somatório dos preços de venda referente à mão-de-obra;
- A83 - Somatório dos preços de venda dos trabalhos a realizar, referente ao país de destino;
- A84 - Somatório dos preços de venda totais dos vários equipamentos, no país de destino;
- A85 - Somatório dos valores gastos relacionados com outros custos, como por exemplo, elementos exteriores ao fornecimento/realização dos trabalhos (unitário);
- A86 - Somatório dos valores gastos relacionados com outros custos, como por exemplo, elementos exteriores ao fornecimento/realização dos trabalhos (parcial);
- A87 - Somatório dos valores destinado às despesas com transporte (seguro);
- A88 - Somatório dos valores destinado às despesas com transporte (frete);
- A89 - Somatório dos valores destinado às despesas com transporte interno, no país de destino;
- A90 - Somatório dos preços de venda dos materiais e do transporte, no país de destino (unitários);
- A91 - Somatório dos preços referentes à mão-de-obra e transporte, no país de destino (unitários);
- A92 - Somatório dos preços de venda totais e do transporte, no país de destino (unitários);
- A93 - Somatório dos preços de venda total e do transporte, no país de destino (parciais);
- A94 - Somatório temporal (em horas) relativo à mão-de-obra, do tipo 1;
- A95 - Somatório dos preços de custo relativo à mão-de-obra, do tipo 1;
- A96 - Somatório temporal (em horas) relativo à mão-de-obra, do tipo 2;
- A97 - Somatório dos preços de custo relativo à mão-de-obra, do tipo 2;
- A98 - Somatório temporal (em horas) relativo à mão-de-obra, do tipo 3;
- A99 - Somatório dos preços de custo relativo à mão-de-obra, do tipo 3;
- A100 - Preço de venda total da empresa no país de origem;
- A101 - Preço de venda total CIF, da empresa no país de destino;
- A102 - Preço de venda total FOB, da empresa no país de destino;
- A103 - Custo dos vários equipamentos no país de origem;
- A104 - Custo de montagem (mão-de-obra) dos vários equipamentos no país de destino;

Apêndice nº1 - Identificação das variáveis

1.2 - Cálculos e Output

- A105** - Margem da empresa no país de origem;
- A106** - Margem da empresa no país de destino;
- A107** - Valor destinado à retenção IRT;
- A108** - Valor destinado a retenção do imposto de selo;
- A109** - Valor destinado ao diretos;
- A110** - Preço dos contentores, referente ao país de destino;
- A111** - Valor destinado ao transporte interno (contentores) no país de destino;
- A112** - Valor destinado ao desalfandegamento;
- A113** - Valor destinado ao seguro transposto no país de destino;
- A114** - Custo dos vários equipamentos no país de origem (em percentagem);
- A115** - Custo de montagem (mão-de-obra) dos vários equipamentos no país de destino (em percentagem);
- A116** - Margem da empresa no país de origem (em percentagem);
- A117** - Margem da empresa no país de destino (em percentagem);
- A118** - Valor total referente ao frete marítimo;
- A119** - Valor final dos trabalhos referidos no mapa de quantidades, no país de destino;
- A120** - Preço parcial dos equipamentos (sem mão-de-obra), projetados no país de origem;
- A121** - Valor do seguro, projetado no país de origem;
- A122** - Valor do frete, projetado no país de origem;
- A123** - Valor total, referente ao pró-forma número 5;
- A124** - Preço unitário da montagem, referente ao país de origem;
- A125** - Peça parcial montagem, referente ao país de origem;
- A126** - Somatório dos preços parcial da montagem, referente ao país de origem;
- A127** - Valor total, referente ao pró-forma número 7.

APÊNDICE

2

Programa Linguagem *Java*

APÊNDICE

2.1

Bases para identificação da proposta (*Proposal*)

Apêndice nº2 - Programa Linguagem Java

2.1 - Bases para identificação da proposta (*Proposal*)

```
package com.isel.mestrado.dto;

/**
 * Created by Eurico on 13/12/2014.
 *
 * ----- 3.2.1.1 - Bases para identificação da proposta -----
 */
public class Proposal {

    private long    proposalId;
    private ProposalType proposalType; /* i0 - national or
international budget */
    private String  name;              /* i1 - Work name */
    private String  localSite;         /* i2 - Work location */
    private float   marginLimit;       /* i3 - Limit in which the
margin A changes to the margin B */
    private Company company;           /* i4 - Company Reference
*/
    private Client  client;            /* i5 - Customer name */
    private Date    deliveryDate;      /* i6 - Current date */
    private String  clientResponsible; /* i7 - It might be
deleted */

    private String  revisionVersion;   /* i8 - Number of proposal
revision */
    private Revision revision;

    public long getProposalId() {
        return proposalId;
    }

    public void setProposalId(long proposalId) {
        this.proposalId = proposalId;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getLocalSite() {
        return localSite;
    }

    public void setLocalSite(String localSite) {
        this.localSite = localSite;
    }

    public float getMarginLimit() {
        return marginLimit;
    }
}
```

Apêndice nº2 - Programa Linguagem Java

2.1 - Bases para identificação da proposta (*Proposal*)

```
public void setMarginLimit(float marginLimit) {
    this.marginLimit = marginLimit;
}

public Company getCompany() {
    return company;
}

public void setCompany(Company company) {
    this.company = company;
}

public Client getClient() {
    return client;
}

public void setClient(Client client) {
    this.client = client;
}

public Date getDeliveryDate() {
    return deliveryDate;
}

public void setDeliveryDate(Date deliveryDate) {
    this.deliveryDate = deliveryDate;
}

public String getClientResponsible() {
    return clientResponsible;
}

public void setClientResponsible(String clientResponsible) {
    this.clientResponsible = clientResponsible;
}

public String getRevisionVersion() {
    return revisionVersion;
}

public void setRevisionVersion(String revisionVersion) {
    this.revisionVersion = revisionVersion;
}

public Revision getRevision() {
    return revision;
}

public void setRevision(Revision revision) {
    this.revision = revision;
}

public ProposalType getProposalType() {
    return proposalType;
}
```

Apêndice nº2 - Programa Linguagem *Java*

2.1 - Bases para identificação da proposta (*Proposal*)

```
public void setProposalType(ProposalType proposalType) {  
    this.proposalType = proposalType;  
}  
}
```

APÊNDICE

2.2

Bases de cálculo para orçamentação (*BudgetBase*)

Apêndice nº2 - Programa Linguagem Java

2.2 - Bases de cálculo para orçamentação (*BudgetBase*)

```
package com.isel.mestrado.dto;

/**
 * Created by Eurico on 13/12/2014.
 *
 * ----- 3.2.1.2 - Bases de cálculo para orçamentação -----
 *
 */
public class BudgetBase {

    private long        budgetBaseId;

    private double      currencyExchange;          /* i9 -
exchange/Relationship between coins */
    private Currency    origin;
    private Currency    destination;

    private double      minimumMarginOrigin;      /* i10 - Company
margin in source A (Org.) */
    private double      maximumMarginOrigin;     /* i11 - Company
margin in source B (Org.)*/
    private double      minimumMarginDestiny;    /* i12 - Company
margin in the target A (Dist.) */
    private double      maximumMarginDestiny;    /* i13 - Company
margin in the target B (Dist.) */
    private double      productivityFactor;       /* i14 - Local
productivity (percentage) */
    private double      insurancePercent;        /* i15 - Insurance
value (percentage) */
    private double      containerCost40;         /* i16 - Cost value of
a container 40" */
    private double      containerCost20;        /* i17 - Cost value of
a container 20" */
    private double      containerTransportDestiny; /* i18 - Transport
cost of 1 container in the destination country */
    private double      containerTransportOrigin; /* i19 - Transport
cost of containers in the source country */

    private double      clearancePercent;        /* i20 - Cost related
to customs clearance (percentage) */
    private double      rightCostPercent;       /* i21 - Right Cost
(percentage) */
    private double      rightCostIRTPercent;    /* i22 - Costs
relating to importation taxes */
    private double      sealPercent;           /* i23 - Cost related
to the stamp tax (percentage) */
    private double      assemblyPerHour1;      /* i24 - Cost of hand
labor time type 1 */
    private double      assemblyPerHour2;      /* i25 - Cost of hand
labor time type 2 */
    private double      assemblyPerHour3;      /* i26 - Cost of hand
labor time type 3 */
    private double      otherCostsPercent;     /* i44 - Amount spent
related to other cost (percentage) (only on proposals with the
country of destination and origin is different */
}
```

Apêndice nº2 - Programa Linguagem Java

2.2 - Bases de cálculo para orçamentação (*BudgetBase*)

```
private String salesterms;          /* i45 - sales terms */

public String getSalesterms() {
    return salesterms;
}

public void setSalesterms(String salesterms) {
    this.salesterms = salesterms;
}

public long getBudgetBaseId() {
    return budgetBaseId;
}

public void setBudgetBaseId(long budgetBaseId) {
    this.budgetBaseId = budgetBaseId;
}

public double getCurrencyExchange() {
    return currencyExchange;
}

public void setCurrencyExchange(double currencyExchange) {
    this.currencyExchange = currencyExchange;
}

public Currency getOrigin() {
    return origin;
}

public void setOrigin(Currency origin) {
    this.origin = origin;
}

public Currency getDestination() {
    return destination;
}

public void setDestination(Currency destination) {
    this.destination = destination;
}

public double getMinimumMarginOrigin() {
    return minimumMarginOrigin;
}

public void setMinimumMarginOrigin(double minimumMarginOrigin)
{
    this.minimumMarginOrigin = minimumMarginOrigin;
}

public double getMaximumMarginOrigin() {
    return maximumMarginOrigin;
}

public void setMaximumMarginOrigin(double maximumMarginOrigin)
{
```

Apêndice nº2 - Programa Linguagem Java

2.2 - Bases de cálculo para orçamentação (*BudgetBase*)

```
        this.maximumMarginOrigin = maximumMarginOrigin;
    }

    public double getMinimumMarginDestiny() {
        return minimumMarginDestiny;
    }

    public void setMinimumMarginDestiny(double
minimumMarginDestiny) {
        this.minimumMarginDestiny = minimumMarginDestiny;
    }

    public double getMaximumMarginDestiny() {
        return maximumMarginDestiny;
    }

    public void setMaximumMarginDestiny(double
maximumMarginDestiny) {
        this.maximumMarginDestiny = maximumMarginDestiny;
    }

    public double getProductivityFactor() {
        return productivityFactor;
    }

    public void setProductivityFactor(double productivityFactor) {
        this.productivityFactor = productivityFactor;
    }

    public double getInsurancePercent() {
        return insurancePercent;
    }

    public void setInsurancePercent(double insurancePercent) {
        this.insurancePercent = insurancePercent;
    }

    public double getContainerCost40() {
        return containerCost40;
    }

    public void setContainerCost40(double containerCost40) {
        this.containerCost40 = containerCost40;
    }

    public double getContainerCost20() {
        return containerCost20;
    }

    public void setContainerCost20(double containerCost20) {
        this.containerCost20 = containerCost20;
    }

    public double getContainerTransportDestiny() {
        return containerTransportDestiny;
    }
}
```

Apêndice nº2 - Programa Linguagem Java

2.2 - Bases de cálculo para orçamentação (*BudgetBase*)

```
public void setContainerTransportDestiny(double
containerTransportDestiny) {
    this.containerTransportDestiny = containerTransportDestiny;
}

public double getContainerTransportOrigin() {
    return containerTransportOrigin;
}

public void setContainerTransportOrigin(double
containerTransportOrigin) {
    this.containerTransportOrigin = containerTransportOrigin;
}

public double getClearancePercent() {
    return clearancePercent;
}

public void setClearancePercent(double clearancePercent) {
    this.clearancePercent = clearancePercent;
}

public double getRightCostPercent() {
    return rightCostPercent;
}

public void setRightCostPercent(double rightCostPercent) {
    this.rightCostPercent = rightCostPercent;
}

public double getRightCostIRTPercent() {
    return rightCostIRTPercent;
}

public void setRightCostIRTPercent(double rightCostIRTPercent)
{
    this.rightCostIRTPercent = rightCostIRTPercent;
}

public double getSealPercent() {
    return sealPercent;
}

public void setSealPercent(double sealPercent) {
    this.sealPercent = sealPercent;
}

public double getAssemblyPerHour1() {
    return assemblyPerHour1;
}

public void setAssemblyPerHour1(double assemblyPerHour1) {
    this.assemblyPerHour1 = assemblyPerHour1;
}

public double getAssemblyPerHour2() {
    return assemblyPerHour2;
}
```

Apêndice nº2 - Programa Linguagem Java

2.2 - Bases de cálculo para orçamentação (*BudgetBase*)

```
}

public void setAssemblyPerHour2(double assemblyPerHour2) {
    this.assemblyPerHour2 = assemblyPerHour2;
}

public double getAssemblyPerHour3() {
    return assemblyPerHour3;
}

public void setAssemblyPerHour3(double assemblyPerHour3) {
    this.assemblyPerHour3 = assemblyPerHour3;
}

public double getOtherCostsPercent() {
    return otherCostsPercent;
}

public void setOtherCostsPercent(double otherCostsPercent) {
    this.otherCostsPercent = otherCostsPercent;
}
}
```

APÊNDICE

2.3

Bases Auxiliares (*AuxiliarBase*)

Apêndice nº2 - Programa Linguagem Java

2.3 - Bases Auxiliares (*AuxiliarBase*)

```
package com.isel.mestrado.dto;

/**
 * Created by Eurico on 11/01/2015.
 */
public class AuxiliarBase {

    private long auxiliarBaseId;
    private int numMonths;      /*i27 - Identifica o número de
meses atribuído (pelo cliente) para execução da obra;*/
    private int businessDays;   /*i28 - Define o número de dias
úteis/trabalháveis por mês;*/

    public long getAuxiliarBaseId() {
        return auxiliarBaseId;
    }

    public void setAuxiliarBaseId(long auxiliarBaseId) {
        this.auxiliarBaseId = auxiliarBaseId;
    }

    public int getNumMonths() {
        return numMonths;
    }

    public void setNumMonths(int numMonths) {
        this.numMonths = numMonths;
    }

    public int getBusinessDays() {
        return businessDays;
    }

    public void setBusinessDays(int businessDays) {
        this.businessDays = businessDays;
    }
}
```

APÊNDICE

2.4

Bases Contentores (*Containers*)

Apêndice nº2 - Programa Linguagem Java

2.4 - Bases Contentores (*Containers*)

```
package com.isel.mestrado.dto;

/**
 * Created by Eurico on 11/01/2015.
 */
public class Containers {

    private long    containersId;

    private int numcontainers20; /*i29 - Representa o número de
um contentor de 20" utilizados;*/
    private int numcontainers40; /*i30 - Representa o número de
um contentor de 40" utilizados;*/

    public long getContainersId() {
        return containersId;
    }

    public void setContainersId(long containersId) {
        this.containersId = containersId;
    }

    public int getNumcontainers20() {
        return numcontainers20;
    }

    public void setNumcontainers20(int numcontainers20) {
        this.numcontainers20 = numcontainers20;
    }

    public int getNumcontainers40() {
        return numcontainers40;
    }

    public void setNumcontainers40(int numcontainers40) {
        this.numcontainers40 = numcontainers40;
    }
}
```

APÊNDICE

2.5

**Bases para identificação dos elementos referentes ao mapa
de quantidades (*Material*)**

Apêndice nº2 - Programa Linguagem Java

2.5 - Bases para identificação dos elementos referentes ao mapa de quantidades

(Material)

```
package com.isel.mestrado.dto;

/**
 * Created by Eurico on 13/12/2014.
 *
 * ----- 3.2.1.4 - Bases para identificação dos elementos
referentes ao mapa de quantidades -----
 *
 */
public class Material {

    private long        materialId;

    private ElemType    elemType;          /* i31 - Tipo de
elemento*/
    private ElemSubtype elemSubtype;      /* i32 - Subtipo de
elemento*/
    private long        numericCode;      /* i33 - Contains the
material numeric code */
    private String      description;      /* i34 - Contains the
material description */
    private UnitType    unitType;        /* i35 - Contains the
material unit type */
    private double      quantity;        /* i36 - Contains the
material quantity */
    private double      price;           /* i37 - Unit value
(material) */
    private double      priceAccessorise; /* i38 - Unit value
of accessories */
    private double      priceTransport;   /* i39 - Value for
the transportation (equipment) */
    private double      priceLocalDestination; /* i40 - Transport
price (international only) */
    private int         timeAssembly1;    /* i41 - Assembly
time in minutes equipment type 1 */
    private int         timeAssembly2;    /* i42 - Assembly
time in minutes equipment type 2 */
    private int         timeAssembly3;    /* i43 - Assembly
time in minutes equipment type 3 */

    public long getMaterialId() {
        return materialId;
    }

    public ElemType getElemType() {
        return elemType;
    }

    public void setElemType(ElemType elemType) {
        this.elemType = elemType;
    }

    public ElemSubtype getElemSubtype() {
        return elemSubtype;
    }
}
```

Apêndice nº2 - Programa Linguagem Java

2.5 - Bases para identificação dos elementos referentes ao mapa de quantidades

(Material)

```
}

public void setElemSubtype(ElemSubtype elemSubtype) {
    this.elemSubtype = elemSubtype;
}

public void setMaterialId(long materialId) {
    this.materialId = materialId;
}

public long getNumericCode() {
    return numericCode;
}

public void setNumericCode(long numericCode) {
    this.numericCode = numericCode;
}

public String getDescription() {
    return description;
}

public void setDescription(String description) {
    this.description = description;
}

public UnitType getUnitType() {
    return unitType;
}

public void setUnitType(UnitType unitType) {
    this.unitType = unitType;
}

public double getQuantity() {
    return quantity;
}

public void setQuantity(double quantity) {
    this.quantity = quantity;
}

public double getPrice() {
    return price;
}

public void setPrice(double price) {
    this.price = price;
}

public double getPriceAccessorisise() {
    return priceAccessorisise;
}

public void setPriceAccessorisise(double priceAccessorisise) {
```

Apêndice nº2 - Programa Linguagem Java

2.5 - Bases para identificação dos elementos referentes ao mapa de quantidades

(Material)

```
        this.priceAccessorise = priceAccessorise;
    }

    public double getPriceTransport() {
        return priceTransport;
    }

    public void setPriceTransport(double priceTransport) {
        this.priceTransport = priceTransport;
    }

    public double getPriceLocalDestination() {
        return priceLocalDestination;
    }

    public void setPriceLocalDestination(double
priceLocalDestination) {
        this.priceLocalDestination = priceLocalDestination;
    }

    public int getTimeAssembly1() {
        return timeAssembly1;
    }

    public void setTimeAssembly1(int timeAssembly1) {
        this.timeAssembly1 = timeAssembly1;
    }

    public int getTimeAssembly2() {
        return timeAssembly2;
    }

    public void setTimeAssembly2(int timeAssembly2) {
        this.timeAssembly2 = timeAssembly2;
    }

    public int getTimeAssembly3() {
        return timeAssembly3;
    }

    public void setTimeAssembly3(int timeAssembly3) {
        this.timeAssembly3 = timeAssembly3;
    }
}
```

APÊNDICE

2.6

Cálculos (*FormulaUtils*)

Apêndice nº2 - Programa Linguagem Java

2.6 - Cálculos (*FormulaUtils*)

```
package com.isel.mestrado.utils;

import com.isel.mestrado.dto.*;

import java.util.Collection;
import java.util.List;

/**
 * Created by Eurico on 11/01/2015.
 */
@SuppressWarnings("UnusedDeclaration")
public class FormulaUtils {

    /*A1= i27×i28 ;*/
    /**
     * (A1)
     * Return the number of business days
     *
     * @param base Auxiliar base containing the fields
     *
     * @return
     *      Number of business days.
     */
    public static int numberTotalDays( AuxiliarBase base ){
        return base.getNumMonths() * base.getBusinessDays();
    }

    /*A2= A94×i14 ;*/
    public static double numberHoursTotalCostMan1(
    Collection<Material> material, BudgetBase budgetBase ){
        return totalAssemblyTimeTyp1(material) *
        budgetBase.getProductivityFactor();
    }

    /*A3= A2÷8 ;*/
    public static double numberTotalDaysTyp1( Collection<Material>
    material, BudgetBase budgetBase ){
        return numberHoursTotalCostMan1(material, budgetBase) / 8;
    }

    /*A4= A3÷A1 ;*/
    public static int numberOfMenTyp1( Collection<Material>
    material, BudgetBase budgetBase, AuxiliarBase base ){
        return (int)Math.ceil( numberTotalDaysTyp1(material,
        budgetBase) / numberTotalDays( base ));
    }

    /*A5= A4÷2 ; */
    public static int numberOfTeamsTyp1( Collection<Material>
    material, BudgetBase budgetBase, AuxiliarBase base ){
        return numberOfMenTyp1(material, budgetBase, base) / 2;
    }

    /*A6= A96×i14 - Typ2*/
    public static double numberHoursTotalCostMan2(
    Collection<Material> material, BudgetBase budgetBase ){
```

Apêndice nº2 - Programa Linguagem Java

2.6 - Cálculos (*FormulaUtils*)

```
        return totalAssemblyTimeType2(material) *
budgetBase.getProductivityFactor();
    }

    /*A7= A6÷8 ; */
    public static double numberTotalDaysType2( Collection<Material>
material, BudgetBase budgetBase ){
        return numberHoursTotalCostMan2(material, budgetBase) / 8;
    }

    /*A8= A7÷A1 ; */
    public static int numberOfMenType2( Collection<Material>
material, BudgetBase budgetBase, AuxiliarBase base ){
        return (int)Math.ceil( numberTotalDaysType2(material,
budgetBase) / numberTotalDays( base ));
    }

    /*A9= A8÷2 ; */
    public static int numberOfTeamsType2( Collection<Material>
material, BudgetBase budgetBase, AuxiliarBase base ){
        return numberOfMenType2(material, budgetBase, base) / 2;
    }

    /*A10= A98×i14 - Typ3*/
    public static double numberHoursTotalCostMan3(
Collection<Material> material, BudgetBase budgetBase ){
        return totalAssemblyTimeType3(material) *
budgetBase.getProductivityFactor();
    }

    /*A11= A10÷8 ; */
    public static double numberTotalDaysType3( Collection<Material>
material, BudgetBase budgetBase ){
        return numberHoursTotalCostMan3(material, budgetBase) / 8;
    }

    /*A12= A11÷A1 ;*/
    public static int numberOfMenType3( Collection<Material>
material, BudgetBase budgetBase, AuxiliarBase base ){
        return (int)Math.ceil( numberTotalDaysType3(material,
budgetBase) / numberTotalDays( base ));
    }

    /*A13= A12÷2 ; */
    public static int numberOfTeamsType3( Collection<Material>
material, BudgetBase budgetBase, AuxiliarBase base ){
        return numberOfMenType3(material, budgetBase, base) / 2;
    }

    /*A14= A2+A6+A10*/
    public static double totalNumberHoursInEx (Collection<Material>
material, BudgetBase budgetBase){
        return numberHoursTotalCostMan1( material,budgetBase
)+numberHoursTotalCostMan2( material,budgetBase )+
numberHoursTotalCostMan3( material, budgetBase );
    }
}
```

Apêndice nº2 - Programa Linguagem Java

2.6 - Cálculos (*FormulaUtils*)

```
    /*A15= A14÷8 ; */
    public static int totalNumberDaysInEx (Collection<Material>
material, BudgetBase budgetBase){
        return (int) Math.ceil(totalNumberHoursInEx ( material,
budgetBase)/8);
    }

    /*A16= A15÷A1 ;*/
    public static int totalNumberMenInEx (Collection<Material>
material, BudgetBase budgetBase, AuxiliarBase base) {
        return (int)Math.ceil((double)totalNumberDaysInEx (
material,budgetBase)/numberTotalDays( base ));
    }

    /*A17= A16÷2 ; */
    public static int totalNumberTeamsComposed(Collection<Material>
material, BudgetBase budgetBase, AuxiliarBase base) {
        return (int)Math.ceil(totalNumberMenInEx ( material,
budgetBase, base)/2);
    }

    /*A18= A84 ; */
    public static double totalCostFreeOnBoards(Collection<Material>
materialList, BudgetBase budgetBase, Proposal proposal){
        return totalSalesPricesEquipmentDestination( materialList,
budgetBase, proposal );
    }

    /*A19= [A66×i15×(1+(i10 ou i11))]×i9 ; Dependendo da relação
entre o A66 e o i3 */
    public static double costInsuranceContainers(
Collection<Material> materialList, BudgetBase budgetBase, Proposal
proposal ){
        double margin;
        double totalSellingPrices;

        totalSellingPrices = totalSellingPricesSource(materialList,
budgetBase, proposal);
        if ( totalSellingPrices < proposal.getMarginLimit() ){
            margin = budgetBase.getMinimumMarginOrigin();
        }else{
            margin = budgetBase.getMaximumMarginOrigin();
        }
        return
totalSellingPrices*(budgetBase.getInsurancePercent()/100)*(1+(margi
n/100))*budgetBase.getCurrencyExchange();
    }

    /*A20=
[i29×(i17×i9)+i30(i16×i9)]+[i29×(i17×i9)+i30(i16×i9)]×(i10 ou i11)
; */
    /* = [i29×i17+i30×i16] (1+(i10 ou i11))×i9*/
    public static double costDerivedSeaFreight (BudgetBase
budgetBase,Proposal proposal ,Containers containers){

        double margin1;
        double totalPriceContainer;
```

Apêndice nº2 - Programa Linguagem Java

2.6 - Cálculos (*FormulaUtils*)

```
        totalPriceContainer =
        (containers.getNumcontainers20()*budgetBase.getContainerCost20()) +
        (containers.getNumcontainers40()*budgetBase.getContainerCost40());
        if ( totalPriceContainer < proposal.getMarginLimit() ){
            margin1 = budgetBase.getMinimumMarginOrigin();
        }else{
            margin1 = budgetBase.getMaximumMarginOrigin();
        }
        return totalPriceContainer *
        budgetBase.getCurrencyExchange() * (1+(margin1/100));
    }

    /*A21= (i29+i30)×i18 ; */
    public static double
    costInternalTransportDestination(Containers containers,BudgetBase
    budgetBase ){
        return
        ((containers.getNumcontainers20()+containers.getNumcontainers40())*
        budgetBase.getContainerTransportDestiny());
    }

    /*A22= A18+A19+A20+A21 ;
    */
    public static double
    sumElementsReferencedContainers(Collection<Material> materialList,
    BudgetBase budgetBase, Proposal proposal, Containers containers ){
        return totalCostFreeOnBoards(materialList, budgetBase,
        proposal)+
            costInsuranceContainers ( materialList,
        budgetBase, proposal ) +
            costDerivedSeaFreight ( budgetBase, proposal ,
        containers) +
            costInternalTransportDestination( containers,
        budgetBase );
    }

    /*A23= A20÷A84 ;
    */
    public static double costMaritimeFreightPercentage
    (Collection<Material> materialList, BudgetBase budgetBase, Proposal
    proposal, Containers containers){
        return (costDerivedSeaFreight ( budgetBase, proposal ,
        containers)/ totalSalesPricesEquipmentDestination( materialList,
        budgetBase, proposal ))*100;
    }

    /*A24= A21÷A84 ;
    */
    public static double costInternalTransportDestinationPercentage
    (Collection<Material> materialList, BudgetBase budgetBase, Proposal
    proposal, Containers containers){
        return (costInternalTransportDestination( containers,
        budgetBase )/totalSalesPricesEquipmentDestination( materialList,
        budgetBase, proposal ))*100;
    }
}
```

Apêndice nº2 - Programa Linguagem Java

2.6 - Cálculos (*FormulaUtils*)

```
/*A27= A57+A58+A59 ; */
public static double mountPriceEquipment (Material material,
BudgetBase budgetBase ){
    return costPriceLabor1( material, budgetBase)+
           costPriceLabor2( material, budgetBase)+
           costPriceLabor3( material, budgetBase) ;
}

/*A28= i37+i38+i39 ;
*/
public static double costPricesEquipment (Material material){
    return material.getPrice()+
material.getPriceAccessorie()+ material.getPriceTransport() ;
}

/* A29= A28×(i10 ou i11) ; Dependendo da relação entre o i37e
o i3
*/
public static double marginEquipmentSource (Material material,
BudgetBase budgetBase, Proposal proposal ){

    double margin;
    if ( material.getPrice() < proposal.getMarginLimit() ){
        margin = budgetBase.getMinimumMarginOrigin();
    }else{
        margin = budgetBase.getMaximumMarginOrigin();
    }
    return costPricesEquipment ( material ) * margin / 100;
}

/* A30= A28+A29 ;
*/
public static double salePriceEquipmentSource (Material
material, BudgetBase budgetBase, Proposal proposal){
    return costPricesEquipment (material )+
marginEquipmentSource (material, budgetBase, proposal) ;
}

/* A31= A30×i9 ;
*/
public static double costMaterialDestinationCountry (Material
material, BudgetBase budgetBase, Proposal proposal){
    return salePriceEquipmentSource (material, budgetBase,
proposal)* budgetBase.getCurrencyExchange() ;
}

/* A32= A27×i14×i9 ;
*/
public static double costHandLaborDestinationCountry (Material
material, BudgetBase budgetBase){
    return mountPriceEquipment (material, budgetBase )*
           budgetBase.getProductivityFactor()*
budgetBase.getCurrencyExchange() ;
}

/* A33= A31+A32 ;
*/
```

Apêndice nº2 - Programa Linguagem Java

2.6 - Cálculos (*FormulaUtils*)

```
public static double costPriceEquipmentDestinationCountry
(Material material, BudgetBase budgetBase, Proposal proposal){
    return costMaterialDestinationCountry (material,
budgetBase, proposal)+ costHandLaborDestinationCountry (material,
budgetBase) ;
}

/* A34= A31×i20 ;
*/
public static double valueForCustomsClearance (Material
material, BudgetBase budgetBase, Proposal proposal){
    return costMaterialDestinationCountry (material,
budgetBase, proposal)* (budgetBase.getClearancePercent()/100) ;
}

/* A35= A31×i21 ;
*/
public static double valueForRights (Material material,
BudgetBase budgetBase, Proposal proposal){
    return costMaterialDestinationCountry (material,
budgetBase, proposal)* (budgetBase.getRightCostPercent()/100) ;
}

/* A36= A33×i22 ;
*/
public static double valueForIRT_Retention (Material material,
BudgetBase budgetBase, Proposal proposal){
    return costPriceEquipmentDestinationCountry (material,
budgetBase, proposal)* (budgetBase.getRightCostIRTPercent()/100) ;
}

/* A37= A33×i23 ;
*/
public static double valueForStampDuty (Material material,
BudgetBase budgetBase, Proposal proposal){
    return costPriceEquipmentDestinationCountry (material,
budgetBase, proposal)* (budgetBase.getSealPercent()/100) ;
}

/* A38=(A31+i40+A34+A35)+(A31×i22)+(A31×i23) ;
*/
public static double costTotalPriceDestinationCountry (Material
material, BudgetBase budgetBase, Proposal proposal){
    return (costMaterialDestinationCountry (material,
budgetBase, proposal)+ material.getPriceLocalDestination()+
valueForCustomsClearance (material, budgetBase,
proposal)+ valueForRights (material, budgetBase, proposal) )+

(costMaterialDestinationCountry (material,
budgetBase, proposal)* (budgetBase.getRightCostIRTPercent()/100))+
(costMaterialDestinationCountry (material,
budgetBase, proposal)* (budgetBase.getSealPercent()/100)) ;
}

/* A39=A32+(A32×i22)+(A32×i23) ;
*/
```

Apêndice nº2 - Programa Linguagem Java

2.6 - Cálculos (*FormulaUtils*)

```
public static double costTotalPriceHandLaborDestinationCountry
(Material material, BudgetBase budgetBase){
    return costHandLaborDestinationCountry
(material,budgetBase)+
        (costHandLaborDestinationCountry
(material,budgetBase)* (budgetBase.getRightCostIRTPercent()/100))+
        (costHandLaborDestinationCountry
(material,budgetBase)* (budgetBase.getSealPercent()/100)) ;
}

/* A40=A38+A39 ;
*/
public static double totalCostPriceDestinationCountry (Material
material, BudgetBase budgetBase, Proposal proposal){
    return costTotalPriceDestinationCountry (material,
budgetBase, proposal)+ costTotalPriceHandLaborDestinationCountry
(material, budgetBase) ;
}

/* A41= A38×(i12 ou i13) ; Dependendo da relação entre o i37e
o i3 */
public static double marginEquipmentDestinationCountry
(Material material, BudgetBase budgetBase, Proposal proposal ){

    double margin;
    if ( material.getPrice() < proposal.getMarginLimit() ){
        margin = budgetBase.getMinimumMarginDestiny();
    }else{
        margin = budgetBase.getMaximumMarginDestiny();
    }
    return costTotalPriceDestinationCountry (material,
budgetBase, proposal)* (margin/100);
}

/* A42= A39×(i12 ou i13) ; Dependendo da relação entre o i37e
o i3 */
public static double marginHandLaborDestinationCountry
(Material material, BudgetBase budgetBase, Proposal proposal ){

    double margin;
    if ( material.getPrice() < proposal.getMarginLimit() ){
        margin = budgetBase.getMinimumMarginDestiny();
    }else{
        margin = budgetBase.getMaximumMarginDestiny();
    }
    return costTotalPriceHandLaborDestinationCountry (material,
budgetBase)* (margin/100);
}

/* A43= A40×(i12 ou i13) ; Dependendo da relação entre o i37e
o i3 */
public static double marginTotalDestinationCountry (Material
material, BudgetBase budgetBase, Proposal proposal ){

    double margin;
    if ( material.getPrice() < proposal.getMarginLimit() ){
        margin = budgetBase.getMinimumMarginDestiny();
```

Apêndice nº2 - Programa Linguagem Java

2.6 - Cálculos (*FormulaUtils*)

```
        }else{
            margin = budgetBase.getMaximumMarginDestiny();
        }
        return totalCostPriceDestinationCountry ( material,
budgetBase, proposal)* (margin/100);
    }

    /* A44=A38+A41 ;
*/
    public static double totalSalePriceEquipmentDestinationCountry
(Material material, BudgetBase budgetBase, Proposal proposal){
        return costTotalPriceDestinationCountry (material,
budgetBase, proposal)+ marginEquipmentDestinationCountry (material,
budgetBase, proposal ) ;
    }

    /* A45=A39+A42 ;
*/
    public static double totalSalePriceHandLaborDestinationCountry
(Material material, BudgetBase budgetBase, Proposal proposal){
        return costTotalPriceHandLaborDestinationCountry (material,
budgetBase)+ marginHandLaborDestinationCountry (material,
budgetBase, proposal ) ;
    }

    /* A46=A40+A43 ;
*/
    public static double totalSaleDestinationCountry (Material
material, BudgetBase budgetBase, Proposal proposal){
        return totalCostPriceDestinationCountry (material,
budgetBase, proposal)+ marginTotalDestinationCountry (material,
budgetBase, proposal ) ;
    }

    /* A47=A46×i36 ;
*/
    public static double itemSalePrice (Material material,
BudgetBase budgetBase, Proposal proposal){
        return totalSaleDestinationCountry (material, budgetBase,
proposal)* material.getQuantity() ;
    }

    /* A48=A46×i44;
*/
    public static double AmountSpentOtherCosts (Material material,
BudgetBase budgetBase, Proposal proposal){
        return totalSaleDestinationCountry (material, budgetBase,
proposal)* (1+(budgetBase.getOtherCostsPercent()/100)) ;
    }

    /* A49=A48×i36 ;
*/
    public static double itemSpentOtherCosts (Material material,
BudgetBase budgetBase, Proposal proposal){
        return AmountSpentOtherCosts (material, budgetBase,
proposal)* material.getQuantity() ;
    }
}
```

Apêndice nº2 - Programa Linguagem Java

2.6 - Cálculos (*FormulaUtils*)

```
/* A50=i36×A30×i15×[1+(i12 ou i13)]×i9 ; Dependendo da relação
entre o i37e o i3 */
public static double itemSpentTransportInsurance (Material
material, BudgetBase budgetBase, Proposal proposal){

    double margin;
    if ( material.getPrice() < proposal.getMarginLimit() ){
        margin = budgetBase.getMinimumMarginOrigin();
    }else{
        margin = budgetBase.getMaximumMarginOrigin();
    }
    return material.getQuantity()* salePriceEquipmentSource
(material, budgetBase, proposal)*
        (budgetBase.getInsurancePercent()/100)*
(1+(margin/100))* budgetBase.getCurrencyExchange();
    }

/* A51=A47×(A20÷A84) ;
*/
public static double itemSpentTransportSeaFreight
(Collection<Material>materialList, Material material, BudgetBase
budgetBase, Proposal proposal, Containers containers){
    return itemSalePrice (material, budgetBase, proposal)*
        (costDerivedSeaFreight (budgetBase, proposal
,containers)/ totalSalesPricesEquipmentDestination(materialList,
budgetBase, proposal)) ;
    }

/* A52=A47×(A21÷A84) ;
*/
public static double
itemSpentInternalTransportDestinationCountry
(Collection<Material>materialList, Material material, BudgetBase
budgetBase, Proposal proposal, Containers containers){
    return itemSalePrice (material, budgetBase, proposal)*
        costInternalTransportDestination( containers,
budgetBase)/ totalSalesPricesEquipmentDestination( materialList,
budgetBase, proposal );
    }

/* A53=A44+(A50+A51+A52)÷i36 ;
*/
public static double
doublesalePriceMaterialsAndTransportDestinationCountry
(Collection<Material>materialList, Material material, BudgetBase
budgetBase, Proposal proposal, Containers containers){
    return totalSalePriceEquipmentDestinationCountry
(material, budgetBase, proposal)+ (itemSpentTransportInsurance
(material, budgetBase, proposal)+
        itemSpentTransportSeaFreight (materialList,
material, budgetBase, proposal, containers)+
itemSpentInternalTransportDestinationCountry (materialList,
material, budgetBase, proposal, containers) )/
        material.getQuantity() ;
    }
```

Apêndice nº2 - Programa Linguagem Java

2.6 - Cálculos (*FormulaUtils*)

```
    /* A54=A45 ;
*/
    public static double
handLaborAndShippingPriceDestinationCountry (Material material,
BudgetBase budgetBase, Proposal proposal){
        return totalSalePriceHandLaborDestinationCountry (material,
budgetBase, proposal) ;
    }

    /* A55=A53+A54 ;
*/
    public static double
TotalSalePricePlusTransportDestinationCountryUnitary
(Collection<Material>materialList, Material material, BudgetBase
budgetBase, Proposal proposal, Containers containers){
        return
doublesalePriceMaterialsAndTransportDestinationCountry
(materialList, material, budgetBase, proposal, containers) +
        handLaborAndShippingPriceDestinationCountry
(material, budgetBase, proposal) ;
    }

    /* A56=A47+A50+A51+A52 ;
*/
    public static double
TotalSalePricePlusTransportDestinationCountry(Collection<Material>m
aterialList, Material material, BudgetBase budgetBase, Proposal
proposal, Containers containers){
        return  itemSalePrice (material, budgetBase, proposal)+
        itemSpentTransportInsurance (material, budgetBase,
proposal) +
        itemSpentTransportSeaFreight (materialList,
material, budgetBase, proposal, containers)+
        itemSpentInternalTransportDestinationCountry
(materialList, material, budgetBase, proposal, containers) ;
    }

    /* A57=i41×(i24÷60) ;
*/
    public static double costPriceLabor1 (Material material,
BudgetBase budgetBase){
        return
material.getTimeAssembly1()*(budgetBase.getAssemblyPerHour1()/60);
    }

    /* A58=i42×(i25÷60) ;
*/
    public static double costPriceLabor2 (Material material,
BudgetBase budgetBase){
        return
material.getTimeAssembly2()*(budgetBase.getAssemblyPerHour2()/60);
    }

    /* A59=i43×(i26÷60) ;
*/
    public static double costPriceLabor3 (Material material,
BudgetBase budgetBase){
```

Apêndice nº2 - Programa Linguagem Java

2.6 - Cálculos (*FormulaUtils*)

```
        return
material.getTimeAssembly3()*(budgetBase.getAssemblyPerHour3()/60);
    }
    /* A60= $\sum$  (i36xi37) ;
*/
    public static double sumPricesEequipment ( Collection<Material>
materialList ){

        double total;

        total = 0.0;
        for ( Material material : materialList){
            total += material.getQuantity() * material.getPrice();
        }
        return total;
    }

    /* A61= $\sum$  (i36xA27) ;
*/
    public static double sumPricesHandLabor ( Collection<Material>
materialList, BudgetBase budgetBase ){

        double total;

        total = 0.0;
        for ( Material material : materialList){
            total += material.getQuantity() * mountPriceEquipment
(material, budgetBase );
        }
        return total;
    }

    /* A62= $\sum$  (i36xi38) ;
*/
    public static double sumPricesEequipmentAccessorise (
Collection<Material> materialList ){

        double total;

        total = 0.0;
        for ( Material material : materialList){
            total += material.getQuantity() *
material.getPriceAccessorise();
        }
        return total;
    }

    /* A63= $\sum$  (i36xi39) ;
*/
    public static double sumPricesEequipmentTransportation (
Collection<Material> materialList ){

        double total;

        total = 0.0;
        for ( Material material : materialList){
```

Apêndice nº2 - Programa Linguagem Java

2.6 - Cálculos (*FormulaUtils*)

```
        total += material.getQuantity() *
material.getPriceTransport();
    }
    return total;
}

/* A64= $\sum (i36 \times A28)$  ;
*/
public static double sumPricesCostEquipment (
Collection<Material> materialList ){

    double total;

    total = 0.0;
    for ( Material material : materialList){
        total += material.getQuantity() * costPricesEquipment (
material );
    }
    return total;
}

/* A65= $\sum (i36 \times A29)$  ;
*/
public static double sumPricesMarginEquipmentSource (
Collection<Material> materialList, BudgetBase budgetBase, Proposal
proposal ){

    double total;

    total = 0.0;
    for ( Material material : materialList){
        total += material.getQuantity() * marginEquipmentSource
(material, budgetBase, proposal );
    }
    return total;
}

/*A66= $\sum (i36 \times A30)$  ;*/
public static double totalSellingPricesSource(
Collection<Material> materialList, BudgetBase budgetBase, Proposal
proposal ){

    double total;

    total = 0.0;
    for ( Material material : materialList){
        total += material.getQuantity() *
salePriceEquipmentSource (material, budgetBase, proposal);
    }
    return total;
}

/* A67= $\sum (i36 \times A31)$  ;*/
public static double sumCostMaterialDestinationCountry (
Collection<Material> materialList, BudgetBase budgetBase, Proposal
proposal ){
```

Apêndice nº2 - Programa Linguagem Java

2.6 - Cálculos (*FormulaUtils*)

```
double total;

total = 0.0;
for ( Material material : materialList){
    total += material.getQuantity() *
costMaterialDestinationCountry (material, budgetBase, proposal);
}
return total;
}

/* A68= $\sum (i36 \times A32)$  ;*/
public static double sumHandLaborDestinationCountry(
Collection<Material> materialList, BudgetBase budgetBase){

double total;

total = 0.0;
for ( Material material : materialList){
    total += material.getQuantity() *
costHandLaborDestinationCountry (material, budgetBase);
}
return total;
}

/* A69= $\sum (i36 \times A33)$  ;*/
public static double sumCostPriceEquipmentDestinationCountry(
Collection<Material> materialList, BudgetBase budgetBase, Proposal
proposal ){

double total;

total = 0.0;
for ( Material material : materialList){
    total += material.getQuantity() *
costPriceEquipmentDestinationCountry (material, budgetBase,
proposal);
}
return total;
}

/* A70= $\sum (i36 \times i40)$  ;*/
public static double sumPricesEequipmentDestination (
Collection<Material> materialList ){

double total;

total = 0.0;
for ( Material material : materialList){
    total += material.getQuantity() *
material.getPriceLocalDestination();
}
return total;
}

/* A71= $\sum (i36 \times A34)$  ;*/
```

Apêndice nº2 - Programa Linguagem Java

2.6 - Cálculos (*FormulaUtils*)

```
public static double sumValueForCustomsClearance (
Collection<Material> materialList, BudgetBase budgetBase, Proposal
proposal ){

    double total;

    total = 0.0;
    for ( Material material : materialList){
        total += material.getQuantity() *
valueForCustomsClearance (material, budgetBase, proposal);
    }
    return total;
}

/* A72= $\sum (i36 \times A35)$  ;*/
public static double sumValueForRights ( Collection<Material>
materialList, BudgetBase budgetBase, Proposal proposal ){

    double total;

    total = 0.0;
    for ( Material material : materialList){
        total += material.getQuantity() * valueForRights (
material, budgetBase, proposal);
    }
    return total;
}

/* A73= $\sum (i36 \times A36)$  ;*/
public static double sumValueForIRT_Retention (
Collection<Material> materialList, BudgetBase budgetBase, Proposal
proposal ){

    double total;

    total = 0.0;
    for ( Material material : materialList){
        total += material.getQuantity() * valueForIRT_Retention
(material, budgetBase, proposal);
    }
    return total;
}

/* A74= $\sum (i36 \times A37)$  ;*/
public static double sumValueForStampDuty (
Collection<Material> materialList, BudgetBase budgetBase, Proposal
proposal ){

    double total;

    total = 0.0;
    for ( Material material : materialList){
        total += material.getQuantity() * valueForStampDuty
(material, budgetBase, proposal);
    }
    return total;
}
}
```

Apêndice nº2 - Programa Linguagem Java

2.6 - Cálculos (*FormulaUtils*)

```
/* A75=∑ (i36×A38) ;*/
public static double sumCostTotalPriceDestinationCountry (
Collection<Material> materialList, BudgetBase budgetBase, Proposal
proposal ){

    double total;

    total = 0.0;
    for ( Material material : materialList){
        total += material.getQuantity() *
costTotalPriceDestinationCountry (material, budgetBase, proposal);
    }
    return total;
}

/* A76=∑ (i36×A39) ;*/
public static double
sumCostTotalPriceHandLaborDestinationCountry ( Collection<Material>
materialList, BudgetBase budgetBase){

    double total;

    total = 0.0;
    for ( Material material : materialList){
        total += material.getQuantity() *
costTotalPriceHandLaborDestinationCountry (material, budgetBase);
    }
    return total;
}

/* A77=∑ (i36×A40) ;*/
public static double sumTotalCostPriceDestinationCountry (
Collection<Material> materialList, BudgetBase budgetBase, Proposal
proposal ){

    double total;

    total = 0.0;
    for ( Material material : materialList){
        total += material.getQuantity() *
totalCostPriceDestinationCountry (material, budgetBase, proposal);
    }
    return total;
}

/* A78=∑ (i36×A41) ;*/
public static double sumMarginEquipmentDestinationCountry (
Collection<Material> materialList, BudgetBase budgetBase, Proposal
proposal ){

    double total;

    total = 0.0;
    for ( Material material : materialList){
```

Apêndice nº2 - Programa Linguagem Java

2.6 - Cálculos (*FormulaUtils*)

```
        total += material.getQuantity() *
marginEquipmentDestinationCountry (material, budgetBase, proposal
);
    }
    return total;
}

/* A79= $\sum$  (i36xA42) ;*/
public static double sumMarginHandLaborDestinationCountry (
Collection<Material> materialList, BudgetBase budgetBase, Proposal
proposal ){

    double total;

    total = 0.0;
    for ( Material material : materialList){
        total += material.getQuantity() *
marginHandLaborDestinationCountry (material, budgetBase, proposal
);
    }
    return total;
}

/* A80= $\sum$  (i36xA43) ;*/
public static double sumMarginTotalDestinationCountry (
Collection<Material> materialList, BudgetBase budgetBase, Proposal
proposal ){

    double total;

    total = 0.0;
    for ( Material material : materialList){
        total += material.getQuantity() *
marginTotalDestinationCountry (material, budgetBase, proposal );
    }
    return total;
}

/* A81= $\sum$  (i36xA44) ;*/
public static double
sumTotalSalePriceEquipmentDestinationCountry ( Collection<Material>
materialList, BudgetBase budgetBase, Proposal proposal ){

    double total;

    total = 0.0;
    for ( Material material : materialList){
        total += material.getQuantity() *
totalSalePriceEquipmentDestinationCountry (material, budgetBase,
proposal);
    }
    return total;
}

/* A82= $\sum$  (i36xA45) ;*/
```

Apêndice nº2 - Programa Linguagem Java

2.6 - Cálculos (*FormulaUtils*)

```
public static double
sumTotalSalePriceHandLaborDestinationCountry ( Collection<Material>
materialList, BudgetBase budgetBase, Proposal proposal ){

    double total;

    total = 0.0;
    for ( Material material : materialList){
        total += material.getQuantity() *
totalSalePriceHandLaborDestinationCountry (material, budgetBase,
proposal);
    }
    return total;
}

/* A83= $\sum$  (i36xA46) ;*/
public static double sumTotalSaleDestinationCountry (
Collection<Material> materialList, BudgetBase budgetBase, Proposal
proposal ){

    double total;

    total = 0.0;
    for ( Material material : materialList){
        total += material.getQuantity() *
totalSaleDestinationCountry (material, budgetBase, proposal);
    }
    return total;
}

/*A84= $\sum$  (A47) ;*/
/**
 * (A84)
 *
 * sum of total sales prices of the various equipment in the
country of destination (partial).
 *
 * @return
 *     price
 */
public static double
totalSalesPricesEquipmentDestination(Collection<Material>
materialList, BudgetBase budgetBase, Proposal proposal){

    double result;

    result = 0.0;
    for( Material material : materialList ){
        result += itemSalePrice(material, budgetBase, proposal
);
    }
    return result;
}

/* A85= $\sum$  (A48) ;      */
```

Apêndice nº2 - Programa Linguagem Java

2.6 - Cálculos (*FormulaUtils*)

```
public static double sumAmountSpentOtherCosts
(Collection<Material> materialList, BudgetBase budgetBase, Proposal
proposal){

    double result;

    result = 0.0;
    for( Material material : materialList ){
        result += AmountSpentOtherCosts (material, budgetBase,
proposal);
    }
    return result;
}

/* A86=Σ (A49) ;      */
public static double sumItemSpentOtherCosts
(Collection<Material> materialList, BudgetBase budgetBase, Proposal
proposal){

    double result;

    result = 0.0;
    for( Material material : materialList ){
        result += itemSpentOtherCosts (material, budgetBase,
proposal);
    }
    return result;
}

/* A87=Σ (A50) ;      */
public static double sumItemSpentTransportInsurance
(Collection<Material> materialList, BudgetBase budgetBase, Proposal
proposal){

    double result;

    result = 0.0;
    for( Material material : materialList ){
        result += itemSpentTransportInsurance (material,
budgetBase, proposal);
    }
    return result;
}

/* A88=Σ (A51) ;      */
public static double sumItemSpentTransportSeaFreight
(Collection<Material> materialList, BudgetBase budgetBase, Proposal
proposal, Containers containers){

    double result;

    result = 0.0;
    for( Material material : materialList ){
        result += itemSpentTransportSeaFreight (materialList,
material, budgetBase, proposal, containers);
    }
    return result;
}
```

Apêndice nº2 - Programa Linguagem Java

2.6 - Cálculos (*FormulaUtils*)

```
}

/* A89=Σ (A52) ; */
public static double
sumItemSpentInternalTransportDestinationCountry(Collection<Material
> materialList, BudgetBase budgetBase, Proposal proposal,
Containers containers){

    double result;

    result = 0.0;
    for( Material material : materialList ){
        result += itemSpentInternalTransportDestinationCountry
(materialList, material, budgetBase, proposal, containers);
    }
    return result;
}

/* A90=Σ (i36×A53) ; */
public static double
sumDoublesalePriceMaterialsAndTransportDestinationCountry(
Collection<Material> materialList, BudgetBase budgetBase, Proposal
proposal, Containers containers ){

    double total;

    total = 0.0;
    for ( Material material : materialList){
        total += material.getQuantity() *
doublesalePriceMaterialsAndTransportDestinationCountry
(materialList, material, budgetBase, proposal, containers);
    }
    return total;
}

/* A91=Σ (i36×A54) ; */
public static double
sumHandLaborAndShippingPriceDestinationCountry (
Collection<Material> materialList, BudgetBase budgetBase, Proposal
proposal ){

    double total;

    total = 0.0;
    for ( Material material : materialList){
        total += material.getQuantity() *
handLaborAndShippingPriceDestinationCountry (material, budgetBase,
proposal);
    }
    return total;
}

/* A92=Σ (i36×A55) ; */
public static double
sumTotalSalePricePlusTransportDestinationCountryUnitary (
Collection<Material> materialList, BudgetBase budgetBase, Proposal
proposal, Containers containers ){
```

Apêndice nº2 - Programa Linguagem Java

2.6 - Cálculos (*FormulaUtils*)

```
double total;

total = 0.0;
for ( Material material : materialList){
    total += material.getQuantity() *
TotalSalePricePlusTransportDestinationCountryUnitary (materialList,
material, budgetBase, proposal, containers);
}
return total;
}

/* A93= $\sum$  A56 ;      */
public static double sumItemSpentOtherCosts
(Collection<Material> materialList, BudgetBase budgetBase, Proposal
proposal, Containers containers){

    double result;

    result = 0.0;
    for( Material material : materialList ){
        result +=
TotalSalePricePlusTransportDestinationCountry(materialList,
material, budgetBase, proposal, containers);
    }
    return result;
}

/*A94=[ $\sum$  (i36xi41)] $\div$ 60 ;*/
/**
 * (A94)
 * Computes the total assembly time type 1 for a list material.
 *
 * @param materialList
 *         Material to be assembled.
 *
 * @return
 *         Time in minues
 */
public static double
totalAssemblyTimeType1(Collection<Material> materialList){

    double result;

    result = 0.0;
    for( Material material : materialList ){
        result += material.getTimeAssembly1() *
material.getQuantity();
    }
    return result/60;
}

/*A95= $\sum$  (i36xA57) ;      */
public static double sumCostPriceLabor1 ( Collection<Material>
materialList, BudgetBase budgetBase ){

    double total;
```

Apêndice nº2 - Programa Linguagem Java

2.6 - Cálculos (*FormulaUtils*)

```
        total = 0.0;
        for ( Material material : materialList){
            total += material.getQuantity() * costPriceLabor1
(material, budgetBase);
        }
        return total;
    }

    /*A96=[ $\sum$  (i36xi42)]÷60 ;*/
    /**
     * (A96)
     * Computes the total assembly time type 2 for a list material.
     *
     * @param materialList
     *       Material to be assembled.
     *
     * @return
     *       Time in minues
     */
    public static double
totalAssemblyTimeType2(Collection<Material> materialList){

        double result;

        result = 0.0;
        for( Material material : materialList ){
            result += material.getTimeAssembly2() *
material.getQuantity();
        }
        return result/60;
    }

    /*A97= $\sum$  (i36xA58) ;      */
    public static double sumcostPriceLabor2 ( Collection<Material>
materialList, BudgetBase budgetBase){

        double total;

        total = 0.0;
        for ( Material material : materialList){
            total += material.getQuantity() * costPriceLabor2
(material, budgetBase);
        }
        return total;
    }

    /*A98=[ $\sum$  (i36xi43)]÷60 ;*/
    /**
     * (A98)
     * Computes the total assembly time type 3 for a list material.
     *
     * @param materialList
     *       Material to be assembled.
     *
     * @return
     *       Time in minues
     */
```

Apêndice nº2 - Programa Linguagem Java

2.6 - Cálculos (*FormulaUtils*)

```
*/
public static double
totalAssemblyTimeType3(Collection<Material> materialList){

    double result;

    result = 0.0;
    for( Material material : materialList ){
        result += material.getTimeAssembly3() *
material.getQuantity();
    }
    return result/60;
}

/*A99= $\sum$  (i36xA59) ; */
public static double sumcostPriceLabor3 ( Collection<Material>
materialList, BudgetBase budgetBase){

    double total;

    total = 0.0;
    for ( Material material : materialList){
        total += material.getQuantity() * costPriceLabor3
(material, budgetBase);
    }
    return total;
}

/*A100= A66+[A66xi15x(1+(i10 ou i11))]+(A20÷i9) ; Dependendo
da relação entre o i37e o i3 */
public static double totalSalePriceCountryOrigin
(Collection<Material> materialList, Material material, BudgetBase
budgetBase, Proposal proposal ,Containers containers ){

    double margin;
    if ( material.getPrice() < proposal.getMarginLimit() ){
        margin = budgetBase.getMinimumMarginOrigin()/100;
    }else{
        margin = budgetBase.getMaximumMarginOrigin()/100;
    }
    return totalSellingPricesSource(materialList, budgetBase,
proposal)+
        ( totalSellingPricesSource(materialList,
budgetBase, proposal) * (budgetBase.getInsurancePercent()/100)*
(1+margin)) +
        ( costDerivedSeaFreight (budgetBase, proposal ,
containers)/ budgetBase.getCurrencyExchange() ) ;

}

/*A101=A22 ; */
public static double
totalSalePrice_CIF_DestinationCountry(List<Material> materialList,
BudgetBase budgetBase, Proposal proposal, Containers containers ){
    return sumElementsReferencedContainers( materialList,
budgetBase, proposal, containers );
}
}
```

Apêndice nº2 - Programa Linguagem Java

2.6 - Cálculos (*FormulaUtils*)

```
    /*A102=A18 ;      */
    public static double
totalSalePrice_FOB_DestinationCountry(Collection<Material>
materialList, BudgetBase budgetBase, Proposal proposal){
    return totalCostFreeOnBoards(materialList, budgetBase,
proposal);
}

    /*A103=A64 ;      */
    public static double totalCostEquipmentCountryOrigin(
Collection<Material> materialList ){
    return sumPricesCostEquipment (materialList );
}

    /*A104=A61×i9×i14 ;*/
    public static double totalHandLaborDestinationCountry(
Collection<Material> materialList, BudgetBase budgetBase ){
    return sumPricesHandLabor (materialList, budgetBase ) *
budgetBase.getCurrencyExchange() *
budgetBase.getProductivityFactor();
}

    /*A105=A65 ;      */
    public static double totalMarginCountryOrigin
(Collection<Material> materialList, BudgetBase budgetBase, Proposal
proposal ){
    return sumPricesMarginEquipmentSource ( materialList,
budgetBase, proposal ) ;
}

    /*A106=A80 ;      */
    public static double totalMarginDestinationCountry
(Collection<Material> materialList, BudgetBase budgetBase, Proposal
proposal ){
    return sumMarginTotalDestinationCountry (materialList,
budgetBase, proposal ) ;
}

    /*A107=A73 ;      */
    public static double totalValue_IRT_Retention
(Collection<Material> materialList, BudgetBase budgetBase, Proposal
proposal ){
    return sumValueForIRT_Retention (materialList, budgetBase,
proposal );
}

    /*A108=A74 ;      */
    public static double totalValueStampDutyRetention
(Collection<Material> materialList, BudgetBase budgetBase, Proposal
proposal ){
    return sumValueForStampDuty ( materialList, budgetBase,
proposal );
}

    /*A109=A72 ;      */
```

Apêndice nº2 - Programa Linguagem Java

2.6 - Cálculos (*FormulaUtils*)

```
public static double totalValueOfTheRights
(Collection<Material> materialList, BudgetBase budgetBase, Proposal
proposal ){
    return sumValueForRights ( materialList, budgetBase,
proposal );
}

/*A110=(i29×i17+i30×i16)×i9 ;*/
public static double totalPriceContainerDestinationCountry (
BudgetBase budgetBase, Containers containers ){
    return ( containers.getNumcontainers20()*
budgetBase.getContainerCost20()+ containers.getNumcontainers40()*
budgetBase.getContainerCost40() ) *
budgetBase.getCurrencyExchange();
}

/*A111=(i29+i30)×i18 ;      */
public static double
totalValueInternalTransportContainerDestinationCountry ( BudgetBase
budgetBase, Containers containers ){
    return ( containers.getNumcontainers20() +
containers.getNumcontainers40() ) *
budgetBase.getContainerTransportDestiny();
}

/*A112=A71 ;                */
public static double totalValueForCustomsClearance
(Collection<Material> materialList, BudgetBase budgetBase, Proposal
proposal ){
    return sumValueForCustomsClearance (materialList,
budgetBase, proposal );
}

/*A113=A19 ;                */
public static double
totalPriceInsuranceContainersDestinationCountry
(Collection<Material> materialList, BudgetBase budgetBase, Proposal
proposal ){
    return costInsuranceContainers( materialList, budgetBase,
proposal );
}

/*A114=A103÷[A103+A104÷i9] ; */
public static double totalCostEquipmentCountryOriginPercentage
(Collection<Material> materialList, BudgetBase budgetBase){
    return (totalCostEquipmentCountryOrigin(materialList )/
(totalCostEquipmentCountryOrigin(materialList
)+(totalHandLaborDestinationCountry(materialList, budgetBase
)/budgetBase.getCurrencyExchange())) ) *100;
}

/*A115=A104÷[A104+A103×i9] ; */
public static double totalCostHandLaborCountryOriginPercentage
(Collection<Material> materialList, BudgetBase budgetBase){
    return totalHandLaborDestinationCountry(materialList,
budgetBase ) / (totalHandLaborDestinationCountry(materialList,
budgetBase
```

Apêndice nº2 - Programa Linguagem Java

2.6 - Cálculos (*FormulaUtils*)

```
)+(totalCostEquipmentCountryOrigin(materialList)*budgetBase.getCurrencyExchange())) * 100;
    }

    /*A116=A105÷A103 ;          */
    public static double totalMarginCountryOriginPercentage
(Collection<Material> materialList, BudgetBase budgetBase, Proposal
proposal ){
    return (totalMarginCountryOrigin (materialList, budgetBase,
proposal )/ totalCostEquipmentCountryOrigin( materialList ) ) *100;
    }

    /*A117=A106÷A77 ;          */
    public static double totalMarginDestinationCountryPercentage
(Collection<Material> materialList, BudgetBase budgetBase, Proposal
proposal ){
    return (totalMarginDestinationCountry (materialList,
budgetBase, proposal)/ sumTotalCostPriceDestinationCountry (
materialList, budgetBase, proposal )) *100;
    }

    /*A118=A20+A21 ;          */
    public static double fullAmountOfOceanFreight ( BudgetBase
budgetBase, Proposal proposal ,Containers containers ){
    return costDerivedSeaFreight (budgetBase,proposal
,containers)+ costInternalTransportDestination(containers,
budgetBase ) ;
    }

    /*A119=A118+A19+A86 ;          */
    public static double finalValueOnTheDestinationCountry
(Collection<Material> materialList, BudgetBase budgetBase, Proposal
proposal ,Containers containers ){
    return fullAmountOfOceanFreight (budgetBase, proposal,
containers)+ costInsuranceContainers (materialList, budgetBase,
proposal )+ sumItemSpentOtherCosts (materialList, budgetBase,
proposal) ;
    }

    /*A120=i36×A30 ;          */
    public static double partialPriceEquipmentOriginCountry (
Material material, BudgetBase budgetBase, Proposal proposal){
    return material.getQuantity()* salePriceEquipmentSource
(material, budgetBase, proposal);
    }

    /*A121=A19÷i9 ;          */
    public static double insuranceValueOriginCountry
(Collection<Material> materialList, BudgetBase budgetBase, Proposal
proposal){
    return costInsuranceContainers(materialList, budgetBase,
proposal )/ budgetBase.getCurrencyExchange();
    }

    /*A122=A20÷i9 ;          */
    public static double valueofoceanfreightOriginCountry (
BudgetBase budgetBase, Proposal proposal,Containers containers){
```

Apêndice nº2 - Programa Linguagem Java

2.6 - Cálculos (*FormulaUtils*)

```
        return costDerivedSeaFreight (budgetBase, proposal ,
containers)/ budgetBase.getCurrencyExchange();
    }

    /*A123=A122+A121+A66 ;          */
    public static double totalAmountProFormaNumber5
(Collection<Material> materialList, BudgetBase budgetBase, Proposal
proposal, Containers containers){
        return valueofoceanfreightOriginCountry ( budgetBase,
proposal, containers)+ insuranceValueOriginCountry ( materialList,
budgetBase, proposal)+
            totalSellingPricesSource ( materialList,
budgetBase, proposal) ;
    }

    /*A124=A27×i14 ;                */
    public static double UnitPriceAssembly ( Material material,
BudgetBase budgetBase){
        return mountPriceEquipment (material, budgetBase )*
budgetBase.getProductivityFactor() ;
    }

    /*A125=A124×i36 ;              */
    public static double partialPriceAssembly ( Material material,
BudgetBase budgetBase){
        return UnitPriceAssembly ( material, budgetBase)*
material.getQuantity() ;
    }

    /*A126=∑ A125 ;                */
    public static double sumPartialPriceAssembly
(Collection<Material> materialList, BudgetBase budgetBase){

        double result;

        result = 0.0;
        for( Material material : materialList ){
            result += partialPriceAssembly ( material, budgetBase);
        }
        return result;
    }

    /*A127=A66+A126 ;              */
    public static double totalAmountProFormaNumber7
(Collection<Material> materialList, BudgetBase budgetBase, Proposal
proposal){
        return totalSellingPricesSource( materialList, budgetBase,
proposal)+ sumPartialPriceAssembly (materialList, budgetBase) ;
    }

}
```

APÊNDICE

3

**Caso de estudo -
Programa de teste aos valores**

Apêndice nº3 - Programa de teste aos valores

```
package com.isel.mestrado.utils;

import com.isel.mestrado.dto.*;
import com.isel.mestrado.enumerate.ElemSubtype;
import com.isel.mestrado.enumerate.ElemType;
import com.isel.mestrado.enumerate.ProposalType;
import com.isel.mestrado.enumerate.UnitType;
import org.junit.Assert;
import org.junit.Before;
import org.junit.Test;
import java.util.*;

public class FormulaUtilsTest {

    AuxiliarBase base;
    List<Material> materialList;
    BudgetBase budgetBase;
    Containers containers;
    Proposal proposal;
    UnitType unitType;

    @Before
    public void setUpTest(){

        Material materialAux;

        /*Define the Auxiliar base*/
        base = new AuxiliarBase();
        base.setBusinessDays(20);          /*i28 - Define o número
de dias úteis/trabalháveis por mês;*/
        base.setNumMonths(3);             /*i27 - Identifica o
número de meses atribuído (pelo cliente) para execução da obra;*/

        /*Define Budgeted Base */
        budgetBase = new BudgetBase();
        budgetBase.setCurrencyExchange(1.4);    /* i9 -
exchange/Relationship between coins */
        budgetBase.setMinimumMarginOrigin(15);    /* i10 -
Company margin in source A (Org.) */
        budgetBase.setMaximumMarginOrigin(15);    /* i11 -
Company margin in source B (Org.)*/
        budgetBase.setMinimumMarginDestiny(25);    /* i12 -
Company margin in the target A (Dist.) */
        budgetBase.setMaximumMarginDestiny(25);    /* i13 -
Company margin in the target B (Dist.) */
        budgetBase.setProductivityFactor(1);    /* i14 - Local
productivity (percentage) */
        budgetBase.setInsurancePercent(1.5);    /* i15 -
Insurance value (percentage) */
        budgetBase.setContainerCost40(4000);    /* i16 - Cost
value of a container 40" */
        budgetBase.setContainerCost20(2650);    /* i17 - Cost
value of a container 20" */
        budgetBase.setContainerTransportDestiny(1725); /* i18 -
Transport cost of 1 container in the destination country */
    }
}
```

Apêndice nº3 - Programa de teste aos valores

```
        budgetBase.setContainerTransportOrigin(0);          /* i19 -
Transport cost of containers in the source country */
        budgetBase.setClearancePercent(10.5);              /* i20 -
Cost related to customs clearance (percentage) */
        budgetBase.setRightCostPercent(0);                 /* i21 -
Right Cost (percentage) */
        budgetBase.setRightCostIRTPercent(3.5);           /* i22 -
Costs relating to importation taxes */
        budgetBase.setSealPercent(1);                      /* i23 -
Cost related to the stamp tax (percentage) */
        budgetBase.setAssemblyPerHour1(30);                /* i24 -
Cost of hand labor time type 1 */
        budgetBase.setAssemblyPerHour2(30);                /* i25 -
Cost of hand labor time type 2 */
        budgetBase.setAssemblyPerHour3(30);                /* i26 -
Cost of hand labor time type 3 */
        budgetBase.setOtherCostsPercent(0);                /* i44 -
Amount spent related to other cost (percentage) (only on proposals
with the country of destination and origin is different */

        /*Define the Containers*/
        containers = new Containers();
        containers.setNumcontainers20(2); /*i29 - Representa o
número de um contentor de 20" utilizados;*/
        containers.setNumcontainers40(1); /*i30 - Representa o
número de um contentor de 40" utilizados;*/

        unitType = new UnitType();
        unitType.setDescription("Meters");
        unitType.setId('m');
        unitType.setUnity("m");

        /*Material List*/
        materialList = new ArrayList<>(); /*Build the list where
the materials are*/

        materialAux = new Material(); /*Material d25*/

        materialAux.setElemType(ElemType.Air); /* i31 -
Tipo de elemento*/
        materialAux.setElemSubtype(ElemSubtype.Cables); /*
i32 - Subtipo de elemento*/
        materialAux.setNumericCode(1); /* i33 -
Contains the material numeric code */
        materialAux.setDescription("d 25"); /* i34 -
Contains the material description */
        materialAux.setUnitType(unitType); /* i35 -
Contains the material unit type */
        materialAux.setQuantity(50); /* i36 -
Contains the material quantity */
        materialAux.setPrice(0.35); /* i37 - Unit
value (materiel) */
        materialAux.setPriceAccessories(0.22); /* i38 - Unit
value of accessories */
        materialAux.setPriceTransport(0); /* i39 -
Value for the transportation (equipment) */
```

Apêndice nº3 - Programa de teste aos valores

```
        materialAux.setPriceLocalDestination(0);          /* i40 -
Transport price (international only) */
        materialAux.setTimeAssembly1(1000);             /* i41 -
Assembly time in minutes equipment type 1 */
        materialAux.setTimeAssembly2(0);                /* i42 -
Assembly time in minutes equipment type 2 */
        materialAux.setTimeAssembly3(0);                /* i43 -
Assembly time in minutes equipment type 3 */

//TODO: Add Other properties/material nº1
materialList.add(materialAux);

        materialAux = new Material();          /*Material d32*/

        materialAux.setElemType(ElemType.Air);        /* i31 -
Tipo de elemento*/
        materialAux.setElemSubtype(ElemSubtype.Cables); /*
i32 - Subtipo de elemento*/
        materialAux.setUnitType(unitType);            /* i35 -
Contains the material unit type */
        materialAux.setNumericCode(2);                /* i33 -
Contains the material numeric code */
        materialAux.setDescription("d 32");           /* i34 -
Contains the material description */
        materialAux.setQuantity(100);                /* i36 -
Contains the material quantity */
        materialAux.setPrice(0.53);                   /* i37 - Unit
value (materiel) */
        materialAux.setPriceAccessories(0.33);        /* i38 - Unit
value of accessories */
        materialAux.setPriceTransport(0);             /* i39 -
Value for the transportation (equipment) */
        materialAux.setPriceLocalDestination(0);      /* i40 -
Transport price (international only) */
        materialAux.setTimeAssembly1(0);              /* i41 -
Assembly time in minutes equipment type 1 */
        materialAux.setTimeAssembly2(15);             /* i42 -
Assembly time in minutes equipment type 2 */
        materialAux.setTimeAssembly3(0);              /* i43 -
Assembly time in minutes equipment type 3 */

//TODO: Add Other properties/material nº2
materialList.add(materialAux);

        materialAux = new Material();          /*Material d40*/

        materialAux.setElemType(ElemType.Air);        /* i31 -
Tipo de elemento*/
        materialAux.setElemSubtype(ElemSubtype.Cables); /*
i32 - Subtipo de elemento*/
        materialAux.setUnitType(unitType);            /* i35 -
Contains the material unit type */
        materialAux.setNumericCode(3);                /* i33 -
Contains the material numeric code */
        materialAux.setDescription("d 40");           /* i34 -
Contains the material description */
        materialAux.setQuantity(10);                  /* i36 -
Contains the material quantity */
```

Apêndice nº3 - Programa de teste aos valores

```
        materialAux.setPrice(0.67);                /* i37 - Unit
value (materiel) */
        materialAux.setPriceAccessorise(0.44);    /* i38 - Unit
value of accessories */
        materialAux.setPriceTransport(0);         /* i39 -
Value for the transportation (equipment) */
        materialAux.setPriceLocalDestination(0);  /* i40 -
Transport price (international only) */
        materialAux.setTimeAssembly1(0);         /* i41 -
Assembly time in minutes equipment type 1 */
        materialAux.setTimeAssembly2(0);         /* i42 -
Assembly time in minutes equipment type 2 */
        materialAux.setTimeAssembly3(15);        /* i43 -
Assembly time in minutes equipment type 3 */
//TODO: Add Other properties/material nº3
materialList.add(materialAux);

Client client;
client = new Client();

Company company;
company = new Company();
company.setCompanyId(123);
company.setCompanyName("example");

/*Define Proposal */
proposal = new Proposal();
proposal.setProposalType(ProposalType.INTERNATIONAL);
/* i0 - national or international budget */
proposal.setName("testel");                      /* i1 -
Work name */
proposal.setLocalSite("Angola");                 /* i2 -
Work location */
proposal.setMarginLimit(50000);                 /* i3 -
Limit in which the margin A changes to the margin B */
proposal.setCompany(company);                   /* i4 -
Company Reference */
proposal.setClient(client);                     /* i5 -
Customer name */
proposal.setDeliveryDate(new Date());           /* i6 -
Current date */
proposal.setClientResponsible("Eurico Clemente"); /* i7 -
It might be deleted */
proposal.setRevisionVersion("0.1 Rev");
/* i8 - Number of proposal revision */

}

@Test /*A1 ;*/
public void testNumberTotalDays() throws Exception {
    int expected = 60;
    int num;

    num = FormulaUtils.numberTotalDays( base );
    Assert.assertEquals(expected, num );
}

@Test /*A2 ;*/
```

Apêndice nº3 - Programa de teste aos valores

```
public void testNumberHoursTotalCostMan1() throws Exception {
    double expected = 833.33;
    double num;

    num = FormulaUtils.numberHoursTotalCostMan1( materialList,
budgetBase);
    Assert.assertEquals(expected, num, 0.01 );
}

@Test /*A3 ;*/
public void testNumberTotalDaysType1() throws Exception {
    double expected = 104.1666667;
    double num;

    num = FormulaUtils.numberTotalDaysType1( materialList,
budgetBase);
    Assert.assertEquals(expected, num, 0.01 );
}

@Test /*A4 Arredondar para cima ;*/
public void testNumberOfMenType1() throws Exception {
    int expected = 2;
    int num;

    num = FormulaUtils.numberOfMenType1( materialList,
budgetBase, base);
    Assert.assertEquals(expected, num );
}

@Test /*A5 ;*/
public void testNumberOfTeamsType1() throws Exception {
    int expected = 1;
    int num;

    num = FormulaUtils.numberOfTeamsType1( materialList,
budgetBase, base);
    Assert.assertEquals(expected, num );
}

@Test /*A6 ;*/
public void testNumberHoursTotalCostMan2() throws Exception {
    double expected = 25;
    double num;

    num = FormulaUtils.numberHoursTotalCostMan2( materialList,
budgetBase);
    Assert.assertEquals(expected, num, 0.01);
}

@Test /*A7 ;*/
public void testNumberTotalDaysType2() throws Exception {
    double expected = 3.125;
    double num;

    num = FormulaUtils.numberTotalDaysType2( materialList,
budgetBase);
    Assert.assertEquals(expected, num, 0.01);
}
```

Apêndice nº3 - Programa de teste aos valores

```
@Test /*A8 Arredondar para cima ;*/
public void testNumberOfMenType2() throws Exception {
    int expected = 1;
    int num;

    num = FormulaUtils.numberOfMenType2( materialList,
budgetBase, base);
    Assert.assertEquals(expected, num );
}

@Test /*A9 ;*/
public void testNumberOfTeamsType2() throws Exception {
    int expected = 0;
    int num;

    num = FormulaUtils.numberOfTeamsType2( materialList,
budgetBase, base);
    Assert.assertEquals(expected, num );
}

@Test /*A10 ;*/
public void testNumberHoursTotalCostMan3() throws Exception {
    double expected = 2.5;
    double num;

    num = FormulaUtils.numberHoursTotalCostMan3( materialList,
budgetBase);
    Assert.assertEquals(expected, num, 0.01);
}

@Test /*A11 ;*/
public void testNumberTotalDaysType3() throws Exception {
    double expected = 0.3125;
    double num;

    num = FormulaUtils.numberTotalDaysType3( materialList,
budgetBase);
    Assert.assertEquals(expected, num, 0.01);
}

@Test /*A12 Arredondar para cima ;*/
public void testNumberOfMenType3() throws Exception {
    int expected = 1;
    int num;

    num = FormulaUtils.numberOfMenType3( materialList,
budgetBase, base);
    Assert.assertEquals(expected, num );
}

@Test /*A13 ;*/
public void testNumberOfTeamsType3() throws Exception {
    int expected = 0;
    int num;

    num = FormulaUtils.numberOfTeamsType3( materialList,
budgetBase, base);
```

Apêndice nº3 - Programa de teste aos valores

```
        Assert.assertEquals(expected, num );
    }

    @Test /*A14 ;*/
    public void testTotalNumberHoursInEx() throws Exception {
        double expected = 860.83;
        double num;

        num = FormulaUtils.totalNumberHoursInEx( materialList,
budgetBase);
        Assert.assertEquals(expected, num, 0.01 );
    }

    @Test /*A15 ;*/
    public void testTotalNumberDaysInEx() throws Exception {
        int expected = 108;
        int num;

        num = FormulaUtils.totalNumberDaysInEx( materialList,
budgetBase);
        Assert.assertEquals(expected, num );
    }

    @Test /*A16 ;*/
    public void testTotalNumberMenInEx() throws Exception {
        int expected = 2;
        int num;

        num = FormulaUtils.totalNumberMenInEx( materialList,
budgetBase, base);
        Assert.assertEquals(expected, num );
    }

    @Test /*A17 Arredondar para cima ;*/
    public void testTotalNumberTeamsComposed() throws Exception {
        int expected = 1;
        int num;

        num = FormulaUtils.totalNumberTeamsComposed( materialList,
budgetBase, base);
        Assert.assertEquals(expected, num );
    }

    @Test /*A18 ;*/
    public void testTotalCostsEquipmentDestination() throws
Exception {
        double expected = 47518.15;
        double num;

        num = FormulaUtils.totalCostFreeOnBoards(materialList,
budgetBase, proposal);
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A19 ;*/
    public void testCostInsuranceContainers() throws Exception {
        double expected = 3.49 ;
        double num;
```

Apêndice nº3 - Programa de teste aos valores

```
        num = FormulaUtils.costInsuranceContainers( materialList,
budgetBase, proposal);
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A20 ;*/
    public void testCostDerivedSeaFreight() throws Exception {
        double expected = 14973;
        double num;

        num = FormulaUtils.costDerivedSeaFreight( budgetBase,
proposal, containers);
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A21 ;*/
    public void testCostInternalTransportDestination() throws
Exception {
        double expected = 5175;
        double num;

        num = FormulaUtils.costInternalTransportDestination(
containers, budgetBase);
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A22 ;*/
    public void testSumElementsReferencedContainers() throws
Exception {
        double expected = 67669.64;
        double num;

        num = FormulaUtils.sumElementsReferencedContainers(
materialList, budgetBase, proposal, containers);
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A23 ;*/
    public void testCostMaritimeFreightPercentage() throws
Exception {
        double expected = 31.51;
        double num;

        num = FormulaUtils.costMaritimeFreightPercentage(
materialList, budgetBase, proposal, containers);
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A24 ;*/
    public void testCostInternalTransportDestinationPercentage()
throws Exception {
        double expected = 10.89;
        double num;

        num =
FormulaUtils.costInternalTransportDestinationPercentage(
materialList, budgetBase, proposal, containers);
```

Apêndice nº3 - Programa de teste aos valores

```
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A27 ;*/
    public void testMountPriceEquipment() throws Exception {
        double expected = 7.50;
        double num;
        Material material;

        material = materialList.get(2);
        num = FormulaUtils.mountPriceEquipment( material,
budgetBase );
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A28 ;*/
    public void testCostPricesEquipment() throws Exception {
        double expected = 1.11;
        double num;
        Material material;

        material = materialList.get(2);
        num = FormulaUtils.costPricesEquipment( material );
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A29 ;*/
    public void testMarginEquipmentSource() throws Exception {
        double expected = 0.17;
        double num;
        Material material;

        material = materialList.get(2);
        num = FormulaUtils.marginEquipmentSource( material,
budgetBase, proposal );
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A30 ;*/
    public void testSalePriceEquipmentSource() throws Exception {
        double expected = 1.28;
        double num;
        Material material;

        material = materialList.get(2);
        num = FormulaUtils.salePriceEquipmentSource( material,
budgetBase, proposal );
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A31 ;*/
    public void testCostMaterialDestinationCountry() throws
Exception {
        double expected = 1.79;
        double num;
        Material material;

        material = materialList.get(2);
```

Apêndice nº3 - Programa de teste aos valores

```
        num = FormulaUtils.costMaterialDestinationCountry(
material, budgetBase, proposal );
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A32 ;*/
    public void testCostHandLaborDestinationCountry() throws
Exception {
        double expected = 10.50;
        double num;
        Material material;

        material = materialList.get(2);
        num = FormulaUtils.costHandLaborDestinationCountry(
material, budgetBase );
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A33 ;*/
    public void testCostPriceEquipmentDestinationCountry() throws
Exception {
        double expected = 12.2871;
        double num;
        Material material;

        material = materialList.get(2);
        num = FormulaUtils.costPriceEquipmentDestinationCountry(
material, budgetBase, proposal );
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A34 ;*/
    public void testValueForCustomsClearance() throws Exception {
        double expected = 0.19;
        double num;
        Material material;

        material = materialList.get(2);
        num = FormulaUtils.valueForCustomsClearance( material,
budgetBase, proposal );
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A35 ;*/
    public void testValueForRights() throws Exception {
        double expected = 0;
        double num;
        Material material;

        material = materialList.get(2);
        num = FormulaUtils.valueForRights( material, budgetBase,
proposal );
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A36 ;*/
    public void testValueForIRT_Retention() throws Exception {
        double expected = 0.43;
```

Apêndice nº3 - Programa de teste aos valores

```
double num;
Material material;

material = materialList.get(2);
num = FormulaUtils.valueForIRT_Retention( material,
budgetBase, proposal );
Assert.assertEquals(expected, num, 0.01);
}

@Test /*A37 ;*/
public void testValueForStampDuty() throws Exception {
double expected = 0.12;
double num;
Material material;

material = materialList.get(2);
num = FormulaUtils.valueForStampDuty( material, budgetBase,
proposal );
Assert.assertEquals(expected, num, 0.01);
}

@Test /*A38 ;*/
public void testCostTotalPriceDestinationCountry() throws
Exception {
double expected = 2.06;
double num;
Material material;

material = materialList.get(2);
num = FormulaUtils.costTotalPriceDestinationCountry(
material, budgetBase, proposal );
Assert.assertEquals(expected, num, 0.01);
}

@Test /*A39 ;*/
public void testCostTotalPriceHandLaborDestinationCountry()
throws Exception {
double expected = 10.97;
double num;
Material material;

material = materialList.get(2);
num =
FormulaUtils.costTotalPriceHandLaborDestinationCountry( material,
budgetBase );
Assert.assertEquals(expected, num, 0.01);
}

@Test /*A40 ;*/
public void testTotalCostPriceDestinationCountry() throws
Exception {
double expected = 13.03;
double num;
Material material;

material = materialList.get(2);
num = FormulaUtils.totalCostPriceDestinationCountry(
material, budgetBase, proposal );
```

Apêndice nº3 - Programa de teste aos valores

```
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A41 ;*/
    public void testMarginEquipmentDestinationCountry() throws
Exception {
        double expected = 0.51;
        double num;
        Material material;

        material = materialList.get(2);
        num = FormulaUtils.marginEquipmentDestinationCountry(
material, budgetBase, proposal );
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A42 ;*/
    public void testMarginHandLaborDestinationCountry() throws
Exception {
        double expected = 2.74;
        double num;
        Material material;

        material = materialList.get(2);
        num = FormulaUtils.marginHandLaborDestinationCountry(
material, budgetBase, proposal );
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A43 ;*/
    public void testMarginTotalDestinationCountry() throws
Exception {
        double expected = 3.26;
        double num;
        Material material;

        material = materialList.get(2);
        num = FormulaUtils.marginTotalDestinationCountry( material,
budgetBase, proposal );
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A44 ;*/
    public void testTotalSalePriceEquipmentDestinationCountry()
throws Exception {
        double expected = 2.57;
        double num;
        Material material;

        material = materialList.get(2);
        num =
FormulaUtils.totalSalePriceEquipmentDestinationCountry( material,
budgetBase, proposal );
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A45 ;*/
```

Apêndice nº3 - Programa de teste aos valores

```
public void testTotalSalePriceHandLaborDestinationCountry()
throws Exception {
    double expected = 13.72;
    double num;
    Material material;

    material = materialList.get(2);
    num =
FormulaUtils.totalSalePriceHandLaborDestinationCountry( material,
budgetBase, proposal );
    Assert.assertEquals(expected, num, 0.01);
}

@Test /*A46 ;*/
public void testTotalSaleDestinationCountry() throws Exception
{
    double expected = 16.28;
    double num;
    Material material;

    material = materialList.get(2);
    num = FormulaUtils.totalSaleDestinationCountry( material,
budgetBase, proposal );
    Assert.assertEquals(expected, num, 0.01);
}

@Test /*A47 ;*/
public void testItemSalePrice() throws Exception {
    double expected = 162.85;
    double num;
    Material material;

    material = materialList.get(2);
    num = FormulaUtils.itemSalePrice( material, budgetBase,
proposal);
    Assert.assertEquals(expected, num, 0.01);
}

@Test /*A48 ;*/
public void testAmountSpentOtherCosts() throws Exception {
    double expected = 16.28;
    double num;
    Material material;

    material = materialList.get(2);
    num = FormulaUtils.AmountSpentOtherCosts( material,
budgetBase, proposal );
    Assert.assertEquals(expected, num, 0.01);
}

@Test /*A49 ;*/
public void testItemSpentOtherCosts() throws Exception {
    double expected = 162.85;
    double num;
    Material material;

    material = materialList.get(2);
```

Apêndice nº3 - Programa de teste aos valores

```
        num = FormulaUtils.itemSpentOtherCosts( material,
budgetBase, proposal);
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A50 ;*/
    public void testItemSpentTransportInsurance() throws Exception
    {
        double expected = 0.31;
        double num;
        Material material;

        material = materialList.get(2);
        num = FormulaUtils.itemSpentTransportInsurance( material,
budgetBase, proposal);
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A51 ;*/
    public void testItemSpentTransportSeaFreight() throws Exception
    {
        double expected = 51.31;
        double num;
        Material material;

        material = materialList.get(2);
        num =
FormulaUtils.itemSpentTransportSeaFreight(materialList, material,
budgetBase, proposal, containers);
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A52 ;*/
    public void testItemSpentInternalTransportDestinationCountry()
throws Exception {
        double expected = 17.73;
        double num;
        Material material;

        material = materialList.get(2);
        num =
FormulaUtils.itemSpentInternalTransportDestinationCountry(materialL
ist, material, budgetBase, proposal, containers);
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A53 ;*/
    public void
testDoublesalePriceMaterialsAndTransportDestinationCountry() throws
Exception {
        double expected = 9.50;
        double num;
        Material material;

        material = materialList.get(2);
        num =
FormulaUtils.doublesalePriceMaterialsAndTransportDestinationCountry
(materialList, material, budgetBase, proposal, containers);
```

Apêndice nº3 - Programa de teste aos valores

```
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A54 ;*/
    public void testHandLaborAndShippingPriceDestinationCountry()
throws Exception {
        double expected = 13.72;
        double num;
        Material material;

        material = materialList.get(2);
        num =
FormulaUtils.handLaborAndShippingPriceDestinationCountry( material,
budgetBase, proposal );
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A55 ;*/
    public void
testTotalSalePricePlusTransportDestinationCountryUnitary() throws
Exception {
        double expected = 23.22;
        double num;
        Material material;

        material = materialList.get(2);
        num =
FormulaUtils.TotalSalePricePlusTransportDestinationCountryUnitary(
materialList, material, budgetBase, proposal, containers);
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A56 ;*/
    public void testTotalSalePricePlusTransportDestinationCountry()
throws Exception {
        double expected = 232.2;
        double num;
        Material material;

        material = materialList.get(2);
        num =
FormulaUtils.TotalSalePricePlusTransportDestinationCountry(
materialList, material, budgetBase, proposal, containers);
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A57 ;*/
    public void testCostPriceLabor1() throws Exception {
        double expected = 0;
        double num;
        Material material;

        material = materialList.get(2);
        num = FormulaUtils.costPriceLabor1( material, budgetBase
);
        Assert.assertEquals(expected, num, 0.01);
    }
}
```

Apêndice nº3 - Programa de teste aos valores

```
@Test /*A58 ;*/
public void testCostPriceLabor2() throws Exception {
    double expected = 0;
    double num;
    Material material;

    material = materialList.get(2);
    num = FormulaUtils.costPriceLabor2( material, budgetBase
);
    Assert.assertEquals(expected, num, 0.01);
}

@Test /*A59 ;*/
public void testCostPriceLabor3() throws Exception {
    double expected = 7.50;
    double num;
    Material material;

    material = materialList.get(2);
    num = FormulaUtils.costPriceLabor3( material, budgetBase
);
    Assert.assertEquals(expected, num, 0.01);
}

@Test /*A60 ;*/
public void testSumPricesEequipment() throws Exception {
    double expected = 77.20;
    double num;

    num = FormulaUtils.sumPricesEequipment( materialList);
    Assert.assertEquals(expected, num, 0.01);
}

@Test /*A61 ;*/
public void testSumPricesHandLabor() throws Exception {
    double expected = 25825;
    double num;

    num = FormulaUtils.sumPricesHandLabor( materialList,
budgetBase);
    Assert.assertEquals(expected, num, 0.01);
}

@Test /*A62 ;*/
public void testSumPricesEequipmentAccessoris() throws
Exception {
    double expected = 48.40;
    double num;

    num = FormulaUtils.sumPricesEequipmentAccessoris(
materialList);
    Assert.assertEquals(expected, num, 0.01);
}

@Test /*A63 ;*/
public void testSumPricesEequipmentTransportation() throws
Exception {
    double expected = 0;
```

Apêndice nº3 - Programa de teste aos valores

```
        double num;

        num = FormulaUtils.sumPricesEequipmentTransportation(
materialList);
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A64 ;*/
    public void testSumPricesCostEquipment() throws Exception {
        double expected = 125.60;
        double num;

        num = FormulaUtils.sumPricesCostEquipment( materialList);
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A65 ;*/
    public void testSumPricesMarginEquipmentSource() throws
Exception {
        double expected = 18.84;
        double num;

        num = FormulaUtils.sumPricesMarginEquipmentSource(
materialList, budgetBase, proposal );
        Assert.assertEquals(expected, num, 0.01 );
    }

    @Test /*A66 ;*/
    public void testTotalSellingPricesSource() throws Exception {
        double expected = 144.44 ;
        double num;

        num = FormulaUtils.totalSellingPricesSource( materialList,
budgetBase, proposal );
        Assert.assertEquals(expected, num , 0.001);
    }

    @Test /*A67 ;*/
    public void testSumCostMaterialDestinationCountry() throws
Exception {
        double expected = 202.22;
        double num;

        num = FormulaUtils.sumCostMaterialDestinationCountry(
materialList, budgetBase, proposal );
        Assert.assertEquals(expected, num, 0.01 );
    }

    @Test /*A68 ;*/
    public void testSumHandLaborDestinationCountry() throws
Exception {
        double expected = 36155 ;
        double num;

        num = FormulaUtils.sumHandLaborDestinationCountry(
materialList, budgetBase);
        Assert.assertEquals(expected, num, 0.01);
    }
}
```

Apêndice nº3 - Programa de teste aos valores

```
@Test /*A69 ;*/
public void testSumCostPriceEquipmentDestinationCountry()
throws Exception {
    double expected = 36357.22;
    double num;

    num = FormulaUtils.sumCostPriceEquipmentDestinationCountry(
materialList, budgetBase, proposal );
    Assert.assertEquals(expected, num, 0.01 );
}

@Test /*A70 ;*/
public void testSumPricesEequipmentDestination() throws
Exception {
    double expected = 0;
    double num;

    num = FormulaUtils.sumPricesEequipmentDestination(
materialList );
    Assert.assertEquals(expected, num, 0.01);
}

@Test /*A71 ;*/
public void testSumValueForCustomsClearance() throws Exception
{
    double expected = 21.23;
    double num;

    num = FormulaUtils.sumValueForCustomsClearance (
materialList, budgetBase, proposal );
    Assert.assertEquals(expected, num, 0.01);
}

@Test /*A72 ;*/
public void testSumValueForRights() throws Exception {
    double expected = 0;
    double num;

    num = FormulaUtils.sumValueForRights( materialList,
budgetBase, proposal );
    Assert.assertEquals(expected, num, 0.01);
}

@Test /*A73 ;*/
public void testSumValueForIRT_Retention() throws Exception {
    double expected = 1272.50;
    double num;

    num = FormulaUtils.sumValueForIRT_Retention( materialList,
budgetBase, proposal );
    Assert.assertEquals(expected, num, 0.01);
}

@Test /*A74 ;*/
public void testSumValueForStampDuty() throws Exception {
    double expected = 363.57;
    double num;
```

Apêndice nº3 - Programa de teste aos valores

```
        num = FormulaUtils.sumValueForStampDuty( materialList,
budgetBase, proposal );
        Assert.assertEquals(expected, num, 0.01 );
    }

    @Test /*A75 ;*/
    public void testSumCostTotalPriceDestinationCountry() throws
Exception {
        double expected = 232.55;
        double num;

        num = FormulaUtils.sumCostTotalPriceDestinationCountry(
materialList, budgetBase, proposal );
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A76 ;*/
    public void testSumCostTotalPriceHandLaborDestinationCountry()
throws Exception {
        double expected = 37781.98;
        double num;

        num =
FormulaUtils.sumCostTotalPriceHandLaborDestinationCountry(
materialList, budgetBase );
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A77 ;*/
    public void testSumTotalCostPriceDestinationCountry() throws
Exception {
        double expected = 38014.52;
        double num;

        num = FormulaUtils.sumTotalCostPriceDestinationCountry(
materialList, budgetBase, proposal );
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A78 ;*/
    public void testSumMarginEquipmentDestinationCountry() throws
Exception {
        double expected = 58.14;
        double num;

        num = FormulaUtils.sumMarginEquipmentDestinationCountry(
materialList, budgetBase, proposal );
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A79 ;*/
    public void testSumMarginHandLaborDestinationCountry() throws
Exception {
        double expected = 9445.49;
        double num;
```

Apêndice nº3 - Programa de teste aos valores

```
        num = FormulaUtils.sumMarginHandLaborDestinationCountry(
materialList, budgetBase, proposal );
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A80 ;*/
    public void testSumMarginTotalDestinationCountry() throws
Exception {
        double expected = 9503.63;
        double num;

        num = FormulaUtils.sumMarginTotalDestinationCountry(
materialList, budgetBase, proposal );
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A81 ;*/
    public void testSumTotalSalePriceEquipmentDestinationCountry()
throws Exception {
        double expected = 290.69;
        double num;

        num =
FormulaUtils.sumTotalSalePriceEquipmentDestinationCountry(
materialList, budgetBase, proposal );
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A82 ;*/
    public void testSumTotalSalePriceHandLaborDestinationCountry()
throws Exception {
        double expected = 47227.47;
        double num;

        num =
FormulaUtils.sumTotalSalePriceHandLaborDestinationCountry(
materialList, budgetBase, proposal );
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A83 ;*/
    public void testSumTotalSaleDestinationCountry() throws
Exception {
        double expected = 47518.15 ;
        double num;

        num = FormulaUtils.sumTotalSaleDestinationCountry(
materialList, budgetBase, proposal );
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A84 ;*/
    public void testTotalSalesPricesEquipmentDestination() throws
Exception {
        double expected = 47518.15;
        double num;
```

Apêndice nº3 - Programa de teste aos valores

```
        num = FormulaUtils.totalSalesPricesEquipmentDestination(
materialList, budgetBase, proposal );
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A85 ;*/
    public void testSumAmountSpentOtherCosts() throws Exception {
        double expected = 947.68;
        double num;

        num = FormulaUtils.sumAmountSpentOtherCosts( materialList,
budgetBase, proposal );
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A86 ;*/
    public void testSumItemSpentOtherCosts() throws Exception {
        double expected = 47518.15;
        double num;

        num = FormulaUtils.sumItemSpentOtherCosts( materialList,
budgetBase, proposal );
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A87 ;*/
    public void testSumItemSpentTransportInsurance() throws
Exception {
        double expected = 3.49;
        double num;

        num = FormulaUtils.sumItemSpentTransportInsurance(
materialList, budgetBase, proposal );
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A88 ;*/
    public void testSumItemSpentTransportSeaFreight() throws
Exception {
        double expected = 14973;
        double num;

        num = FormulaUtils.sumItemSpentTransportSeaFreight(
materialList, budgetBase, proposal, containers );
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A89 ;*/
    public void
testSumItemSpentInternalTransportDestinationCountry() throws
Exception {
        double expected = 5175;
        double num;

        num =
FormulaUtils.sumItemSpentInternalTransportDestinationCountry(materi
alList, budgetBase, proposal, containers);
        Assert.assertEquals(expected, num, 0.01 );
    }
}
```

Apêndice nº3 - Programa de teste aos valores

```
    }

    @Test /*A90 ;*/
    public void
testSumDoublesalePriceMaterialsAndTransportDestinationCountry()
throws Exception {
    double expected = 20442.17;
    double num;

    num =
FormulaUtils.sumDoublesalePriceMaterialsAndTransportDestinationCoun
try( materialList, budgetBase, proposal, containers );
    Assert.assertEquals(expected, num, 0.01);
}

    @Test /*A91 ;*/
    public void
testSumHandLaborAndShippingPriceDestinationCountry() throws
Exception {
    double expected = 47227.47;
    double num;

    num =
FormulaUtils.sumHandLaborAndShippingPriceDestinationCountry(
materialList, budgetBase, proposal );
    Assert.assertEquals(expected, num, 0.01);
}

    @Test /*A92 ;*/
    public void
testSumTotalSalePricePlusTransportDestinationCountryUnitary()
throws Exception {
    double expected = 67669.64;
    double num;

    num =
FormulaUtils.sumTotalSalePricePlusTransportDestinationCountryUnitar
y( materialList, budgetBase, proposal, containers );
    Assert.assertEquals(expected, num, 0.01);
}

    @Test /*A94 ;*/
    public void testTotalAssemblyTimeType1() throws Exception {
    double expected = 833.33;
    double num;

    num = FormulaUtils.totalAssemblyTimeType1(materialList );
    Assert.assertEquals(expected , num, 0.004);
}

    @Test /*A95 ;*/
    public void testSumCostPriceLabor1() throws Exception {
    double expected = 25000;
    double num;

    num = FormulaUtils.sumCostPriceLabor1( materialList,
budgetBase );
    Assert.assertEquals(expected, num , 0);
}
```

Apêndice nº3 - Programa de teste aos valores

```
}

@Test /*A96 ;*/
public void testTotalAssemblyTimeType2() throws Exception {
    double expected = 25;
    double num;

    num = FormulaUtils.totalAssemblyTimeType2( materialList );
    Assert.assertEquals(expected, num, 0.01);
}

@Test /*A97 ;*/
public void testSumcostPriceLabor2() throws Exception {
    double expected = 750;
    double num;

    num = FormulaUtils.sumcostPriceLabor2( materialList,
budgetBase );
    Assert.assertEquals(expected, num, 0.01);
}

@Test /*A98 ;*/
public void testTotalAssemblyTimeType3() throws Exception {
    double expected = 2.50;
    double num;

    num = FormulaUtils.totalAssemblyTimeType3( materialList);
    Assert.assertEquals(expected, num, 0.01);
}

@Test /*A99 ;*/
public void testSumcostPriceLabor3() throws Exception {
    double expected = 75;
    double num;

    num = FormulaUtils.sumcostPriceLabor3( materialList,
budgetBase );
    Assert.assertEquals(expected, num, 0.01);
}

@Test /*A100 ;*/
public void testTotalSalePriceCountryOrigin() throws Exception
{
    double expected = 10841.93;
    double num;
    Material material;

    material = materialList.get(0);
    num = FormulaUtils.totalSalePriceCountryOrigin(
materialList, material, budgetBase, proposal, containers );
    Assert.assertEquals(expected, num, 0.01);
}

@Test /*A101 ;*/
public void testTotalSalePrice_CIF_DestinationCountry() throws
Exception {
    double expected = 67669.64;
    double num;
```

Apêndice nº3 - Programa de teste aos valores

```
        num = FormulaUtils.totalSalePrice_CIF_DestinationCountry(
materialList, budgetBase, proposal, containers );
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A102 ;*/
    public void testTotalSalePrice_FOB_DestinationCountry() throws
Exception {
        double expected = 47518.15;
        double num;

        num = FormulaUtils.totalSalePrice_FOB_DestinationCountry(
materialList, budgetBase, proposal );
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A103 ;*/
    public void testTotalCostEquipmentCountryOrigin() throws
Exception {
        double expected = 125.60;
        double num;

        num = FormulaUtils.totalCostEquipmentCountryOrigin(
materialList );
        Assert.assertEquals(expected, num, 0.01 );
    }

    @Test /*A104 ;*/
    public void testTotalHandLaborDestinationCountry() throws
Exception {
        double expected = 36155;
        double num;

        num = FormulaUtils.totalHandLaborDestinationCountry(
materialList, budgetBase );
        Assert.assertEquals(expected, num, 0.01);
    }

    @Test /*A105 ;*/
    public void testTotalMarginCountryOrigin() throws Exception {
        double expected = 18.84;
        double num;

        num = FormulaUtils.totalMarginCountryOrigin( materialList,
budgetBase, proposal );
        Assert.assertEquals(expected, num, 0.01 );
    }

    @Test /*A106 ;*/
    public void testTotalMarginDestinationCountry() throws
Exception {
        double expected = 9503.63;
        double num;

        num = FormulaUtils.totalMarginDestinationCountry(
materialList, budgetBase, proposal );
        Assert.assertEquals(expected, num, 0.01);
    }
}
```

Apêndice nº3 - Programa de teste aos valores

```
}

@Test /*A107 ;*/
public void testTotalValue_IRT_Retention() throws Exception {
    double expected = 1272.50;
    double num;

    num = FormulaUtils.totalValue_IRT_Retention( materialList,
budgetBase, proposal );
    Assert.assertEquals(expected, num, 0.01 );
}

@Test /*A108 ;*/
public void testTotalValueStampDutyRetention() throws Exception
{
    double expected = 363.57;
    double num;

    num = FormulaUtils.totalValueStampDutyRetention(
materialList, budgetBase, proposal );
    Assert.assertEquals(expected, num, 0.01);
}

@Test /*A109 ;*/
public void testTotalValueOfTheRights() throws Exception {
    double expected = 0;
    double num;

    num = FormulaUtils.totalValueOfTheRights( materialList,
budgetBase, proposal );
    Assert.assertEquals(expected, num, 0.01);
}

@Test /*A110 ;*/
public void testTotalPriceContainerDestinationCountry() throws
Exception {
    double expected = 13020;
    double num;

    num = FormulaUtils.totalPriceContainerDestinationCountry(
budgetBase, containers );
    Assert.assertEquals(expected, num, 0.01);
}

@Test /*A111 ;*/
public void
testTotalValueInternalTransportContainerDestinationCountry() throws
Exception {
    double expected = 5175.0;
    double num;

    num =
FormulaUtils.totalValueInternalTransportContainerDestinationCountry
( budgetBase, containers );
    Assert.assertEquals(expected, num, 0.01);
}

@Test /*A112 ;*/
```

Apêndice nº3 - Programa de teste aos valores

```
public void testTotalValueForCustomsClearance() throws
Exception {
    double expected = 21.23;
    double num;

    num = FormulaUtils.totalValueForCustomsClearance(
materialList, budgetBase, proposal );
    Assert.assertEquals(expected, num, 0.01);
}

@Test /*A113 ;*/
public void
testTotalPriceInsuranceContainersDestinationCountry() throws
Exception {
    double expected = 3.49;
    double num;

    num =
FormulaUtils.totalPriceInsuranceContainersDestinationCountry(
materialList, budgetBase, proposal );
    Assert.assertEquals(expected, num, 0.01);
}

@Test /*A114 ;*/
public void testTotalCostEquipmentCountryOriginPercentage()
throws Exception {
    double expected = 0.48;
    double num;

    num =
FormulaUtils.totalCostEquipmentCountryOriginPercentage (
materialList, budgetBase );
    Assert.assertEquals(expected, num, 0.01);
}

@Test /*A115 ;*/
public void testTotalCostHandLaborCountryOriginPercentage()
throws Exception {
    double expected = 99.52;
    double num;

    num =
FormulaUtils.totalCostHandLaborCountryOriginPercentage (
materialList, budgetBase );
    Assert.assertEquals(expected, num, 0.01);
}

@Test /*A116 ;*/
public void testTotalMarginCountryOriginPercentage() throws
Exception {
    double expected = 15;
    double num;

    num = FormulaUtils.totalMarginCountryOriginPercentage(
materialList, budgetBase, proposal );
    Assert.assertEquals(expected, num, 0.01);
}
```

Apêndice nº3 - Programa de teste aos valores

```
@Test /*A117 ;*/
public void testTotalMarginDestinationCountryPercentage()
throws Exception {
    double expected = 25;
    double num;

    num = FormulaUtils.totalMarginDestinationCountryPercentage(
materialList, budgetBase, proposal );
    Assert.assertEquals(expected, num, 0.01 );
}

@Test /*A118 ;*/
public void testFullAmountOfOceanFreight() throws Exception {
    double expected = 20148;
    double num;

    num = FormulaUtils.fullAmountOfOceanFreight( budgetBase,
proposal, containers );
    Assert.assertEquals(expected, num, 0.01);
}

@Test /*A119 ;*/
public void testFinalValueOnTheDestinationCountry() throws
Exception {
    double expected = 67669.64;
    double num;

    num = FormulaUtils.finalValueOnTheDestinationCountry(
materialList, budgetBase, proposal, containers );
    Assert.assertEquals(expected, num, 0.01);
}

@Test /*A120 !Atenção valor variável! ;*/
public void testPartialPriceEquipmentOriginCountry() throws
Exception {
    double expected = 12.77;
    double num;
    Material material;

    material = materialList.get(2);
    num = FormulaUtils.partialPriceEquipmentOriginCountry(
material, budgetBase, proposal);
    Assert.assertEquals(expected, num, 0.005 );
}

@Test /*A121 ;*/
public void testInsuranceValueOriginCountry() throws Exception
{
    double expected = 2.49;
    double num;

    num = FormulaUtils.insuranceValueOriginCountry(
materialList, budgetBase, proposal );
    Assert.assertEquals(expected, num, 0.01);
}

@Test /*A122 ;*/
```

Apêndice nº3 - Programa de teste aos valores

```
public void testValueofocanfreightOriginCountry() throws
Exception {
    double expected = 10695;
    double num;

    num = FormulaUtils.valueofocanfreightOriginCountry(
budgetBase, proposal, containers );
    Assert.assertEquals(expected, num, 0.01);
}

@Test /*A123 ;*/
public void testTotalAmountProFormaNumber5() throws Exception {
    double expected = 10841.93;
    double num;

    num = FormulaUtils.totalAmountProFormaNumber5(
materialList, budgetBase, proposal, containers );
    Assert.assertEquals(expected, num, 0.01);
}

@Test /*A124 !Atenção valor variável! ;*/
public void testUnitPriceAssembly() throws Exception {
    double expected = 7.50;
    double num;
    Material material;

    material = materialList.get(2);
    num = FormulaUtils.UnitPriceAssembly( material, budgetBase
);
    Assert.assertEquals(expected, num, 0.01);
}

@Test /*A125 !Atenção valor variável! ;*/
public void testPartialPriceAssembly() throws Exception {
    double expected = 75;
    double num;
    Material material;

    material = materialList.get(2);
    num = FormulaUtils.partialPriceAssembly( material,
budgetBase);
    Assert.assertEquals(expected, num, 0.01);
}

@Test /*A126 ;*/
public void testSumPartialPriceAssembly() throws Exception {
    double expected = 25825;
    double num;

    num = FormulaUtils.sumPartialPriceAssembly( materialList,
budgetBase );
    Assert.assertEquals(expected, num, 0.01);
}

@Test /*A127 ;*/
public void testTotalAmountProFormaNumber7() throws Exception {
    double expected = 25969.44;
    double num;
```

Apêndice nº3 - Programa de teste aos valores

```
        num = FormulaUtils.totalAmountProFormaNumber7(  
materialList, budgetBase, proposal);  
        Assert.assertEquals(expected, num, 0.01);  
    }  
}
```